

24/12



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
FACULTAD DE PSICOLOGIA  
División de Estudios PROFESIONALES

UNIVERSIDAD NACIONAL  
AUTONOMA DE  
MEXICO

DESARROLLO Y EVALUACION DE UNA INTERFASE PARA EL  
USUARIO DEL LENGUAJE DE INTERROGACION ESTRUCTURADO A  
BASES DE DATOS ( S Q L ).

TESIS PROFESIONAL

Que para obtener el Título de :

LICENCIADO EN PSICOLOGIA

Presenta :

JAVIER KATUMA ARAD TOYOHARA

Director de tesis :

MTRD. FRANCISCO JAVIER URBINA SORIA

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# INDICE

RESUMEN .....	1
I. ANTECEDENTES .....	2
1.0 INTRODUCCION .....	2
2.0 BASES DE DATOS .....	4
2.1 Sistema administrador de base de datos .....	5
2.2 Ventajas del sistema administrador de bases de datos..	6
3.0 LENGUAJES DE INTERROGACION A BASES DE DATOS .....	7
3.1 Interrogación Por menú .....	7
3.2 Lenguaje natural de interrogación .....	7
3.3 Lenguaje formal de interrogación .....	7
4.0 COMPARACION DE INTERROGADORES A BASES DE DATOS .....	9
5.0 SQL, LENGUAJE DE INTERROGACION ESTRUCTURADO .....	10
5.1 Evaluación psicológica del SQL .....	10
5.2 Mecanismos psicológicos en la producción de interro- gantes .....	11
5.2.1 Análisis de las etapas de producción de interro- gantes .....	13
6.0 FUENTES DE ERROR EN LA ETAPA DE CONDIFICACION DE INTERROGANTES EN SQL .....	14
6.1 Errores de tipo sintáctico .....	14
6.2 Errores de tipo lexicográfico al transcribir las clau- sulas del SQL.....	14
6.3 Errores de tipo lexicográfico al asignar nombres a las tablas y atributos.....	15
7.0 CARACTERISTICAS DE LA INTERFASE PARA SQL. ....	15
7.1 Objetivos de la interfase.....	16
7.2 Plantilla SQL.....	17
7.3 Plantilla de nombres de tablas y atributos .....	18
II. METODO .....	19
1.0 SUJETOS. ....	19
2.0 EQUIPO Y MATERIALES .....	19
3.0 DISEÑO EXPERIMENTAL .....	19
4.0 PROCEDIMIENTO .....	20
4.1 Fase de ambientación .....	20
4.2 Fase experimental.....	20
4.3 Tareas .....	21
4.3.1 Verificación .....	21
4.3.2 Construcción .....	21
4.3.3 Análisis de las tareas .....	22
4.4 Fase de evaluación .....	22
4.5 Periodos de evaluación .....	22
4.6 Variable independiente .....	22
4.7 Variables dependientes .....	22
4.7.1 Tiempos .....	22
4.7.2 Respuestas correctas .....	22
4.7.3 Errores .....	23

III. RESULTADOS .....	24
1.0 TAREA DE CONSTRUCCION .....	24
1.1 Aciertos .....	24
1.2 Tiempos .....	27
1.3 Errores .....	29
1.3.1 Errores menores .....	29
1.3.2 Errores de tipo lógico.....	31
1.3.3 Errores de tipo sintactico .....	32
2.0 TAREA DE VERIFICACION. ....	35
2.1 Aciertos .....	35
2.2 Tiempos .....	37
3.0 TAREA DE VERIFICACIÓN (RECHAZOS CORRECTOS).....	39
3.1 Rechazos correctos .....	39
3.2 Tiempos .....	42
IV. DISCUSION. ....	43
1.0 ¿ CUAL ES EL EFECTO DE LA INTERFASE EN EL APRENDIZAJE DEL SQL? .....	43
1.1 Programar los interrogantes eficientemente .....	43
1.2 Identificar las reglas sintácticas válidas para construir interrogantes .....	43
1.3 Detectar errores de sintaxis en interrogantes ya formulados .....	44
1.4 Limitaciones de la interfase .....	44
2.0 DESEMPEÑO DE LOS SUJETOS UNA SEMANA POSTERIOR AL EXPERIMENTO .....	44
3.0 LIMITACIONES METODOLOGICAS DEL PRESENTE ESTUDIO.....	45
3.1 Diseño .....	45
3.2 Evaluación .....	45
4.0 PERSPECTIVAS PARA EL DESARROLLO FUTURO DE IMNTERFASES PARA EL USUARIO DEL SQL.....	45
4.1 Descripción global de la producción de interrogantes...48	
4.1.1 Formulación del problema.....	48
4.1.2 Preparación del plan.....	48
4.1.3 Codificación del problema.....	50
4.2 Comparación de 3 interfases para el usuario del SQL...51	
4.2.1.1 Formulación del problema (requerimiento 1)...51	
4.2.1.2 Formulación del problema (requerimiento 2)...53	
4.2.2 Preparación del plan.....	53
4.2.3 Codificación del problema.....	55
5.0 CONCLUSIONES .....	59
V. BIBLIOGRAFIA.....	60

## VI. APENDICES.

Apendice A.	Modelos de sistemas de bases de datos.....	62
Apendice B.	Claúsulas del SQL.....	67
Apendice C.	Tipos de interrogante del SQL.....	70
Apendice D.	Instructivo de Colores.....	74
Apendice E.	Notas del SQL.....	83
Apendice F.	Programa de entrenamiento.....	113
Apendice G.	Estrategia para el diseño de interfaces para el usuario de lenguajes formales de Interrogación a bases de datos.....	115
Apéndice H.	Interrogantes en SQL utilizados en la tarea de verificación .....	117
Apéndice I.	Preguntas utilizadas en la tarea de construcción.....	123
VII. GLOSARIO.....		125

## INDICE DE FIGURAS Y TABLAS.

FIGURA 1.	Componentes de un sistema administrador de bases de datos .....	6
FIGURA 2.	Etapas y factores en la producción de interrogantes..	12
FIGURA 3.	Interfase para el SQL .....	16
FIGURA 4.	Tarea de construcción: promedio de aciertos de los grupos control e interfase por tipo de Interrogante ..	25
FIGURA 5.	Tarea de construcción: Promedio de aciertos de los grupos control e interfase en el interrogante de selección de grupos en la primera y segunda evaluación .....	26
FIGURA 6.	Tiempos promedio de construcción de interrogantes de los grupos control e interfase en la primera y segunda evaluación .....	28
FIGURA 7.	Tarea de verificación: Promedio de aciertos de los grupos control e interfase por tipo de interrogante..	36
FIGURA 8.	Tiempos promedio de verificación de interrogantes de los grupos control e interfase en la primera y segunda evaluación .....	38
FIGURA 9.	Tarea de verificación: Promedio de rechazos correctos de los grupos control e interfase por tipo de interrogante .....	40
FIGURA 10.	Tiempos promedio de verificación de interrogantes con errores de sintaxis de los grupos control e interfase en la primera y segunda evaluación .....	43
FIGURA 11.	Interfase para el usuario de SQL en base a menús ..	48
FIGURA 12.	Ayuda para formular interrogantes .....	52
FIGURA 13.	Ayuda para la preparación del plan .....	54
FIGURA 14.	Ayuda para la codificación del interrogante .....	56
FIGURA 15.	Construcción de interrogantes .....	57
TABLA 1.	Esquema de clasificación para las consideraciones de diálogo .....	3
TABLA 2.	Comparación de interrogadores a bases de datos.....	8
TABLA 3.	Análisis de varianza para los aciertos en la tarea de construcción .....	24

TABLA 4.	Análisis de varianza para los tiempos en la tarea de construcción .....	27
TABLA 5.	Errores menores en la tarea de construcción.....	30
TABLA 6.	Errores de tipo lógico en la tarea de construcción...	31
TABLA 7.	Errores de tipo sintáctico en la tarea de construcción.....	34
TABLA 8.	Análisis de varianza para los aciertos en la tarea de verificación .....	35
TABLA 9.	Análisis de varianza para los tiempos en los aciertos de la tarea de verificación .....	37
TABLA 10.	Análisis de varianza para los rechazos correctos en la tarea de verificación .....	39
TABLA 11.	Errores cometidos en la tarea de verificación.....	41
TABLA 12.	Análisis de varianza para los tiempos en los rechazos correctos de la tarea de verificación .....	42

## RESUMEN

### Desarrollo y Evaluación de una Interfase para el Usuario del Lenguaje de Interrogación Estructurado a Bases de Datos (SQL).

En el presente trabajo se describe el desarrollo y evaluación de una interfase para el usuario del SQL (Lenguaje de Interrogación Estructurado a bases de datos). La interfase está diseñada con el propósito de aumentar la facilidad de uso de este lenguaje y consiste en una plantilla que identifica las cláusulas más comunes del SQL mediante colores. Y se ajusta al teclado de funciones PF (16) de las terminales OT-80. La función de los colores en la plantilla consiste en proporcionar a los usuarios indicios perceptuales de la validez e invalidez sintáctica de los tipos de interrogantes del SQL. (selección, proyección, ordenamiento, Único, operaciones agregadas, agrupamiento y selección de grupos). Con el objeto de evaluar los alcances y limitaciones de esta interfase, se entrenó en las mismas condiciones a dos grupos de sujetos en el manejo básico de la sintaxis del SQL, excepto que un grupo de sujetos (experimental) se entrenó con la ayuda de la interfase, mientras que el otro grupo (control) se entrenó sin ella. Todos los sujetos fueron entrenados mediante un modelo algorítmico que describía secuencialmente las manipulaciones de tablas (atributos, registros y datos) requeridas en cada tipo de interrogante sintáctica del SQL. Posteriormente al entrenamiento, los sujetos fueron evaluados en dos tareas; construcción y verificación de interrogantes (queries). En la tarea de construcción los sujetos programaban un interrogante en SQL dada una pregunta en español; en la tarea de verificación, los sujetos juzgaban el valor de verdad de un interrogante (detectaban errores de sintaxis). Los resultados encontrados son los siguientes: 1) en la tarea de construcción los sujetos entrenados con la interfase tuvieron mayor número de aciertos y utilizaron menor tiempo para programar interrogantes en SQL en comparación con los sujetos del grupo control; la diferencia en aciertos de ambos grupos se manifestó en la construcción sintáctica de selección de grupos. 2) En la tarea de verificación los sujetos se enfrentaron a dos situaciones: a) juzgar un interrogante como verdadero cuando es verdadero (acierto) y b) juzgar un interrogante como falso cuando es falso (rechazo correcto). Los sujetos entrenados con la interfase tuvieron significativamente mayor número de rechazos correctos que los sujetos del grupo control, sin embargo ambos grupos no se diferenciaron en el tiempo de verificación; en cuanto a los aciertos no se encontraron diferencias significativas en la cantidad de aciertos ni en el tiempo de verificación de los mismos. En este estudio se concluye que la interfase para el SQL tuvo un efecto positivo en el aprendizaje del SQL pero que el desempeño de los sujetos en esta interfase no fue del todo óptima; habría que tomar en cuenta aspectos no considerados en el diseño de la interfase para mejorar aún más la facilidad de uso del SQL. A este respecto se presenta una opción que podría cumplir con tal propósito.

## I. ANTECEDENTES

### 1.0 INTRODUCCION.

En la actualidad, las computadoras son cada vez más solicitadas por usuarios ajenos a la programación; hoy en día no es extraña la solicitud de computadoras en ambientes tan diversos como el hogar, la empresa, la escuela y la industria.

Esta gran demanda implica un nuevo reto para los diseñadores de sistemas: adaptar los sistemas a conveniencia de los nuevos usuarios, y no esperar a que sean estos quienes se adapten a los sistemas (Ehrenreich, 1981).

Este reto no solo implica instrumentar sistemas que se ajusten a las necesidades específicas de informática de los usuarios; mas bien, el verdadero problema consiste en diseñar los sistemas de acuerdo a las capacidades y limitaciones del ser humano.

Esto es considerar el diseño de hardware y ambientes de trabajo adecuados a la antropometría de los usuarios, así como el problema de comunicación entre el humano y la computadora, es decir el diseño de la interfase de información entre sistema y usuario (Williges y Williges, 1983). Este problema concierne a todos los medios por los cuales interactúan el usuario y el sistema, por ejemplo, documentación del sistema ya sea en papel o en línea (Cohill y Williges, 1985; Torres y Bibriesca, 1986); calidad de los mensajes del sistema cuando el usuario comete errores (Brown, 1983); formatos de presentación de la información al usuario (Just, Carpenter y Woolley, 1982; Gould y Grischkowsky, 1984); diseños de modos de diálogo sistema-usuario similares al lenguaje nativo de los usuarios (Reisner, 1977; Ehrenreich, 1981).

La calidad de estos medios es crucial, ya que de ello depende en gran parte la aceptación o rechazo del sistema por parte de los usuarios, máxime cuando estos son novatos.

Williges y Williges (1983) proponen una clasificación de aspectos a considerar en el desarrollo de lineamientos para el diseño de la interfase sistema-usuario. (Véase Tabla 1)

## ANTECEDENTES

TABLA 1. Esquema de clasificación para las consideraciones de diálogo (tomada y resumida de Williges y Williges, 1983)

---

### 1. ORGANIZACION DE DATOS

- 1.1 Codificación de la información
- 1.2 Densidad de la información
- 1.3 Rótulos
- 1.4 Formatos
- 1.5 Arreglo de la pantalla

### 2. MODOS DE DIALOGO

- 2.0 Elección del diálogo
- 2.1 Llenado de formas
- 2.3 Inquirir a la computadora
- 2.4 Lenguaje de comandos
- 2.5 Lenguajes Formales de Interrogación
- 2.6 Lenguajes Naturales Restringidos

### 3. DISPOSITIVOS DE ENTRADA DE LOS USUARIOS

- 3.0 Procedimientos de captura de datos
- 3.1 Selección del Dispositivo de entrada
- 3.2 Teclados
- 3.3 Control directo de los apuntadores
- 3.4 Controles continuos
- 3.5 Tabletas de gráficas
- 3.6 Analizadores de voz

### 4. RETROALIMENTACION Y MANEJO DE ERRORES

- 4.1 Retroalimentación
- 4.2 Recuperación de errores
- 4.3 Control del usuario
- 4.4 Ayuda y documentación
- 4.5 Ayudas de la computadora

### 5. SEGURIDAD Y PREVENCIÓN DE DESASTRES

- 5.1 Cancelación de comandos
  - 5.2 Verificación de acciones ambiguas o destructivas
  - 5.3 Control de secuencias
  - 5.4 Fallas del sistema
-

## ANTECEDENTES

En esta tabla se presentan algunos de los aspectos involucrados en el estudio de la interacción sistema-usuario. En el tópicó de organización de datos se investiga los aspectos relacionados a la forma en la que el sistema debe de presentar la información a sus usuarios, por ejemplo, ¿Cuál es el efecto per se de una pantalla de tubos de rayos catódicos (CRT) y por otro lado cuál el efecto del trabajo por sí solo en el desempeño de un operador? (Gould y Grischkowsky, 1984). En el área de modos de diálogo se investigan las formas de comunicación directa entre el sistema y el usuario, por ejemplo, ¿Cuáles son las ventajas y desventajas del lenguaje natural como modo de diálogo entre sistema y usuario? (Ehrenreich, 1981). En los dispositivos de entrada de los usuarios se estudian los medios de operación que ofrece el sistema, por ejemplo, ¿cual es la configuración idónea del teclado alfanumérico que permita al usuario escribir más rápidamente con un mínimo de errores? (Hirsch, 1981). En la retroalimentación y manejo de errores se investigan las medidas de apoyo del sistema ante acciones equivocadas del usuario, por ejemplo, ¿Que opciones hay para mejorar la calidad de los mensajes de error? (Brown, 1983).

El interés por establecer las condiciones óptimas para el diálogo hombre-computadora compete a lo que se denomina estudio de Factores Humanos. "Es una disciplina técnica que estudia básicamente lo que constituye las mejores interfases entre el humano y la máquina. Los intereses pueden entrar en el desarrollo de un producto ó un sistema en cualquiera ó en todas las etapas, desde el diseño inicial hasta la evaluación y uso en la práctica. Los estudios resultan casi siempre de recomendaciones específicas de diseño ó cambio de productos ó de procedimientos para su operación de tal manera que sus usuarios no los encuentren difíciles de usar" (Hirsch, 1981).

El presente trabajo pretende incorporarse a dicha área, en este escrito se presenta el diseño y evaluación de un dispositivo de entrada para los usuarios de un modo de diálogo sistema-usuario.

Una de las necesidades de informática más apremiantes en la sociedad actual concierne al manejo de volúmenes de datos en gran escala, que requieren ser registrados y recuperados con un máximo de eficiencia y, preferentemente, con un mínimo de conocimientos de programación (dadas las características de los usuarios potenciales). Los sistemas administradores de bases de datos (SABAD) son una instancia de sistemas diseñados para cumplir con esas demandas; administrar grandes volúmenes de datos asumiendo que el usuario no tiene conocimientos avanzados en programación.

### 2.0 BASE DE DATOS.

Por base de datos se puede entender a un conjunto de archivos electrónicos que almacenan datos.

La unidad básica de una base de datos de tipo relacional (véase Apéndice A) se denomina atributo, ejemplos de atributo son: R. F. C. del empleado, el nombre del empleado; los atributos se agrupan en tablas (archivos), por ejemplo la tabla de empleado contiene los atributos de R. F. C. del empleado,

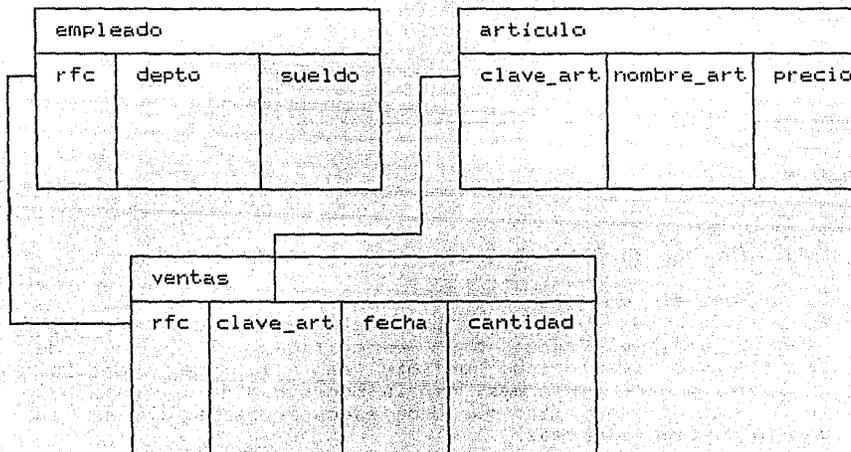
## ANTECEDENTES

departamento y sueldo del empleado.

tabla empleado

atributo	atributo	atributo
rfc	depto	sueldo
UISF501203	Almacén	500,000.00
TOCO550703	Véntas	600,000.00
AATJ621204	Almacén	300,000.00
CAHG601215	Compras	600,000.00
HUMM580812	Véntas	300,000.00
LUZB621201	Véntas	400,000.00

Ahora bien, por base de datos se entiende a una colección de tablas relacionadas entre sí.



(Véase Apéndice A, para mayor información respecto al tópico de bases de datos)

### 2.1 SISTEMA ADMINISTRADOR DE BASES DE DATOS.

El SABAD se compone por una serie de lenguajes y utilerías que permiten en condiciones óptimas: la definición, el registro, procesamiento ó recuperación de grandes volúmenes de datos (lo que es propiamente la base de datos). Comúnmente se identifica al Sistema administrador de base de datos como "base de datos"; en este trabajo entenderemos por base de datos al dispositivo que almacena los datos; y por sistema administrador de base de datos al conjunto de lenguajes y utilerías que permiten utilizar los datos (la base de datos).

## ANTECEDENTES

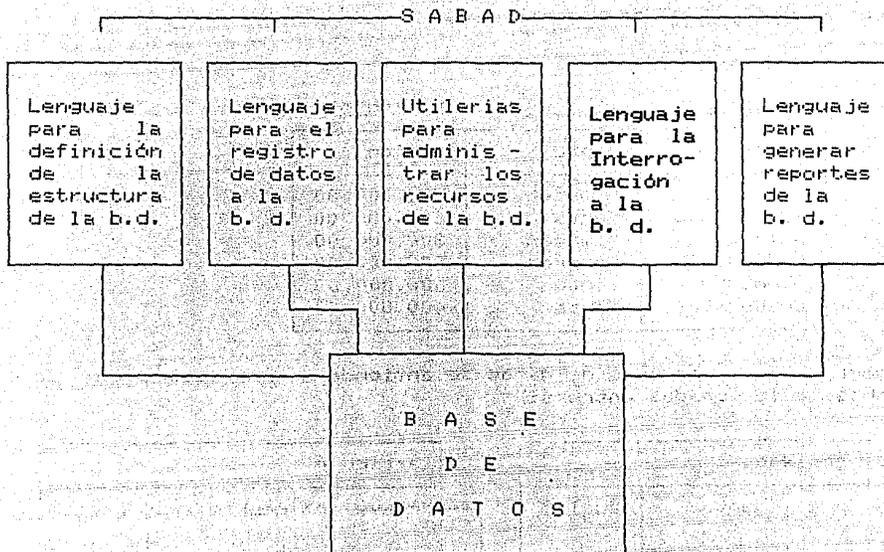


FIGURA 1. Componentes de un sistema administrador de bases de datos

### 2.2 VENTAJAS DEL SISTEMA ADMINISTRADOR DE BASES DE DATOS.

El SABAD es un gran avance tecnológico, dado que los lenguajes y utilerías que lo conforman están orientados para que el usuario efectúe tareas bien específicas (lenguajes de propósitos específicos) como definir pantallas de captura, extraer datos, generar informes, sin la necesidad de programar en detalle la secuencia de pasos que debe ejecutar la máquina en la realización de dichas tareas.

Lo opuesto ocurre en los lenguajes convencionales de programación tales como Basic ó Pascal (lenguajes de propósitos generales) en los cuales el programador debe especificar detalladamente la serie de pasos que debe ejecutar la máquina en una tarea dada. En los lenguajes que conforman los SABAD, el programador no necesita saber la secuencia de pasos que ocurren durante la ejecución de una tarea, es decir el usuario de estos sistemas establece lo que quiere que realice la máquina sin la necesidad de establecer el cómo. Es por esta razón por la cual los SABAD son tan populares y solicitados en el mercado.

El modo de diálogo entre sistema y usuario, concretamente el lenguaje de interrogación a la base de datos tiene una gran relevancia para los diseñadores de SABAD, en virtud de que los usuarios finales (usuarios para los que se genera la base de datos) se concentran primordialmente en esta tarea.

En este sentido, los esfuerzos de los diseñadores se han dirigido a instrumentar lenguajes de interrogación a bases de datos que tengan un máximo de potencia y gran facilidad de uso.

## ANTECEDENTES

### 3.0 LENGUAJES DE INTERROGACION A BASES DE DATOS.

Existen por lo menos tres formas posibles de interrogar a una bases de datos (Torres, 1984a):

- Interrogación por Menú.
- Lenguaje Natural de Interrogación.
- Lenguaje Formal de Interrogación

Veamos como un usuario interrogaría a una base de datos con cada una de estas formas de interrogación.

Suponga que desea saber cuáles empleados (R. F. C.) ganan de 500,000.00 en adelante.

#### 3.1 INTERROGACION POR MENU.

En la interrogación por Menú, el usuario elige la opción que cumpla con su requerimiento de información, esta forma de interrogación no asume sintaxis o vocabulario alguno para interrogar a la base de datos.

por ejemplo,

M E N U
1. Relación de Empleados
2. Relación de Empleados y Deptos.
3. Relación de Empleados y Sueldos

¿Opción ? 3                      <el usuario elige la opción 3 >  
¿Sueldo >= ? 500,000.00 <el sistema pregunta por el Sueldo>  
                                         <y el usuario tecldea 500,000.00 >

#### 3.2 LENGUAJE NATURAL DE INTERROGACION.

En el lenguaje natural de interrogación (LENI), el usuario interroga mediante su lenguaje materno a la base de datos, esta forma de interrogación tampoco asume el aprendizaje de una sintaxis o vocabulario en particular.

por ejemplo,

Dame los empleados que ganen de 500,000.00 en adelante.

#### 3.3 LENGUAJE FORMAL DE INTERROGACION.

En un lenguaje formal de interrogación (LEFI), el usuario interroga a la base de datos mediante una sintaxis y vocabulario estructurado, que requieren aprendizaje.

ANTECEDENTES

por ejemplo,

(Interrogante en SQL, Structured Query Language)

```
select rfc
from nómina
where sueldo >= 500,000.00/
```

Una de las preguntas más frecuentes que se hacen los usuarios al enfrentarse por primera vez a un sistema es, ¿ Qué tanto entrenamiento requiero para interactuar con el sistema ?, ¿ Qué lenguaje tengo que aprender ?

En este sentido, el interrogador ideal para una base de datos debe ser aquél que ofrezca, por un lado FACILIDAD DE USO, y por otro, POTENCIA óptima.

TABLA 2. Comparación de Interrogadores a Bases de Datos ( tomada de Torres, 1984a )

INTERROGACION	VERSATILIDAD	SITUACIONES A LAS QUE SE ADECUA	POTENCIA	FACILIDAD DE USO	FORMA DE INTERROGACION	COSTOS TECNICOS DE OPERACION
Por Menú	Restringida	Interrogantes bien definidos	Baja	Muy Alta	Automática	Bajo
L E N I	Ilimitada	Interrogantes muy diversos e indefinidos	Alta	Alta, pero el usuario debe de adaptarse a las capacidades del sistema	Diversa	Alto
L E F I	Ilimitada	Interrogantes muy diversos e indefinidos	Alta	Alta, pero el usuario requiere de un entrenamiento formal	Estructurada	Mediano

## ANTECEDENTES

### 4.0 COMPARACION DE INTERROGADORES A BASES DE DATOS.

Como podemos constatar, la interrogación por menú es la más sencilla y económica de las tres y resulta la más adecuada cuando los interrogantes están bien definidos. (véase Tabla 2)

Ahora bien, cuando las necesidades de interrogación son muy versátiles o desconocidas, el menú de opciones es limitado, ya que el programador del menú no tiene idea de la diversidad de opciones de consulta de los usuarios (en condiciones reales el menú se conforma en la medida que se van detectando las opciones de interrogación). Esta restricción del menú de interrogación no se aplica a los lenguajes natural y formal de interrogación, ya que éstos ofrecen una gama más amplia de posibilidades de interrogación, es decir tienen mayor potencia.

Al parecer, la interrogación a las bases de datos mediante el lenguaje de uno mismo sería el idóneo para los usuarios finales, ya que la potencialidad de este lenguaje es ilimitada, además, la adaptación de los usuarios al sistema sería casi inmediata. No es inmediata, porque el usuario tendría que adaptarse a las limitaciones del lenguaje, por ejemplo.

usuario > "dame los nombres de los clientes que solicitaron más de diez manuscritos durante la semana pasada".

sistema > "no entiendo la palabra manuscritos, me puedes dar un equivalente".

usuario > "libros".

sistema > "no entiendo la semana pasada, creo que te refieres a un periodo de tiempo, si es así, te pediría que me dieras la fecha de inicio y la fecha de término en el formato dd-mm-aa".

Sin embargo, hay críticas severas que cuestionan al lenguaje natural como el interrogador ideal para una base de datos.

En primer término, el volumen del léxico de un lenguaje natural es considerablemente extenso.

EL lenguaje natural es muy ambiguo para ser interpretado correctamente por un sistema de cómputo Hill (1972).

p. ej. Juan ha crecido un pie.  
El metro tiene 10500 metros.

El estilo de redacción varía prácticamente de usuario a usuario.

p. ej. ¿Cuál es el motivo de la huelga?  
¿ La huelga, que motivo tiene?  
¿ Qué persigue la huelga?  
¿ Qué fin tiene la huelga?

Imagine los requerimientos técnicos necesarios para el mantenimiento de un sistema que enfrentara estos aspectos. Este lenguaje sería incosteable comercialmente.

## ANTECEDENTES

La alternativa a los Lenguajes Naturales de Interrogación, la constituye los Lenguajes Formales de Interrogación. Estos lenguajes se componen de un léxico muy reducido y, por un conjunto muy estructurado de reglas sintácticas. Estos lenguajes están diseñados para proporcionar al usuario: facilidad de uso, potencialidad y un estilo estructurado de pedir las cosas.

El SQL (Structured Query Language) es una instancia de lenguaje formal de interrogación, este lenguaje fue desarrollado por la IBM y es considerado el interrogador estándar a bases de datos (Entrevista a Michael Braude). (Véase Apéndice A)

### 5.0 SQL, LENGUAJE DE INTERROGACION ESTRUCTURADO.

El SQL se compone por un vocabulario ( cláusulas en el idioma inglés ) y un conjunto estructurado de reglas sintácticas. (véase el Apéndice B para detalles de sus cláusulas y sintaxis, y el Apéndice C para el tipo de interrogantes -estructuras sintácticas - con las cuales trabaja)

En este tipo de lenguaje, el usuario declara el tipo de datos que requiere sin la necesidad de establecer el cómo, para esto el usuario requiere de un entrenamiento para el aprendizaje del vocabulario y la sintaxis del lenguaje. Esta restricción haría pensar que un LENI sería el candidato ideal de interrogador de base de datos.

### 5.1 EVALUACION PSICOLOGICA DEL SQL.

Small y Weldon (1983) emprendieron un experimento en el que compararon la efectividad y facilidad de uso del inglés versus un LEFI (SQL) en una tarea de interrogación a una base de datos.

Los sujetos del experimento realizaron tareas de recuperación de datos en inglés y en SQL sin que se dieran cuenta. Los autores no encontraron diferencias significativas en la exactitud de las tareas en ambos lenguajes; y, lo que es más interesante, encontraron que el SQL es más rápido de utilizar que el inglés en este tipo de tareas. Este resultado se explica debido a que la estructura del SQL esta diseñada conforme a la estructura de las tareas específicas de interrogación a una B. D. (Sandberg, 1981; Torres, 1984a).

Es decir, el SQL obliga al usuario a definir sus interrogantes en términos de la estructura per se de la base de datos, lo cual representa una gran ventaja en virtud de que el usuario expresa sus necesidades de forma metódica, evitando con esto la ambigüedad intrínseca del lenguaje natural.

En otras palabras, el tiempo que se invierte en el aprendizaje del lenguaje se compensa en precisión y rapidez a la hora de la interrogación.

Sin embargo, a pesar de que el SQL es fácil y rápido de usar, los usuarios deben de aprender su sintaxis y vocabulario. Ahora bien, ¿ qué tan fácil es de aprender el SQL ? ¿ A qué tipo de problemas se enfrentan los usuarios de SQL ? A este respecto Reiser (1977), hizo una evaluación psicológica del SQL con el

## ANTECEDENTES

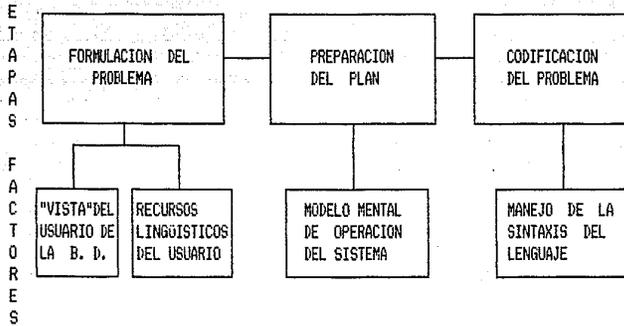
objeto de observar, 1) Qué tan fácil es de aprender y 2) Qué mecanismos están involucrados en la construcción de interrogantes en SQL. La autora entrenó en el uso del SQL a sujetos con experiencia en programación y sin ella, y reporta que el lenguaje sí es fácil de aprender por ambos tipos de usuarios, aunque para esto recomienda que los rasgos del lenguaje deben dividirse en grupos, de tal manera que éstos se utilicen de acuerdo a las necesidades o sofisticación de los usuarios (véase apéndice B).

Sin embargo, el aprendizaje del SQL no está exento de errores, posteriormente se presentará un análisis de los errores que encontró Reiser (1977). Previo a este análisis es conveniente conceptualizar la forma en que trabaja un usuario al utilizar el SQL.

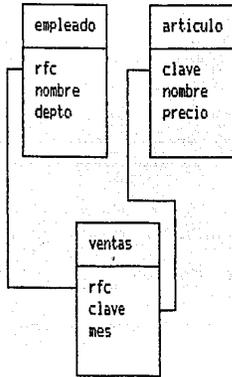
### 5.2 MECANISMOS PSICOLÓGICOS EN LA PRODUCCIÓN DE INTERROGANTES.

Gould y Ascher (1975) identifican tres etapas que ocurren en el proceso de producción de un interrogante, que comprende desde el planteamiento de la consulta por el usuario hasta la elaboración del interrogante en SQL, 1) formulación del problema, 2) preparación del plan y 3) codificación del problema (véase fig. 2).

ANTECEDENTES



vista del usuario de la base de datos.



La vista corresponde a la porción de la Base de Datos a la que tiene acceso el usuario. Es común que para cada usuario exista una vista en particular.

modelo conceptual de operación del sistema.

1

clave	nombre_art	precio
1	LAPIZ	5
2	BORRADOR	10
3	HOJA	15
4	PLUMA	30
5	FOLDER	40
6	PLUMIN	50

sintaxis del lenguaje (SQL)

```
select precio
from articulo
where nombre_art= 'PLUMA'/'
```

2

clave	nombre_art	precio
1	LAPIZ	5
3	HOJA	15
4	PLUMA	30
5	FOLDER	40
6	PLUMIN	50

3

clave	nombre_art	precio
4	PLUMA	30

4

precio
30

El modelo mental de operación del sistema corresponde a las operaciones que infiere el usuario como necesarias para obtener los datos.

1. El usuario quiere saber el precio de la pluma.
2. Se selecciona el registro que tenga en el nombre\_art "PLUMA"
3. Del registro seleccionado se proyecta el precio.

FIGURA 2. Etapas y factores en la producción de interrogantes.

## ANTECEDENTES

### 5.2.1. ANALISIS DE LAS ETAPAS DE PRODUCCION DE INTERROGANTES.

La relación de estas etapas es secuencial ya que cometer algún error en cualquiera de las etapas conduce a una producción inadecuada del interrogante. Los errores son muy diversos, tales como omitir un signo de puntuación al escribir el interrogante, utilizar erróneamente las cláusulas del lenguaje, elegir un plan equivocado para recuperar los datos, definir inadecuadamente lo que se desea consultar.

Las etapas también son independientes, Gould y Ascher (1975) reportan que al manipular alguna etapa el tiempo invertido en ella se incrementaba mientras que el tiempo dedicado a las otras etapas no se alteraba.

Asumiendo que las etapas son secuenciales e independientes se podrían inferir ciertos factores que afectan dichas etapas (Sternberg, 1969).

En la formulación del problema el usuario define el interrogante (qué datos y en qué formato los desea) en su lenguaje materno (no necesariamente de una manera explícita); en esta etapa se ven involucrados dos factores básicos: 1) "vista" del usuario de la base de datos, es decir tablas a las que tiene acceso (ver glosario) y 2) lingüístico (véase fig 2).

En la segunda etapa el usuario traza un plan mediante el cual predice el tipo de operaciones requeridas por el sistema para obtener los datos definidos en la etapa anterior. La consistencia entre el modelo conceptual del usuario del funcionamiento del sistema y del funcionamiento real del sistema es crucial para el éxito de esta etapa. (Williges y Williges, 1983; Schlager y Daden, 1986) (véase fig 2).

Finalmente, en la etapa de codificación del problema, el usuario traduce las operaciones del sistema inferidas en la etapa de preparación en términos de la sintaxis del lenguaje de interrogación a utilizarse (véase fig. 2).

Con el objeto de observar los mecanismos psicológicos que se ven involucrados en la construcción de interrogantes, Reisner (1977) hace un análisis de los errores que cometen los usuarios en el uso del SQL, análisis que se ubica en la etapa de codificación del problema de acuerdo al esquema de Gould y Ascher (1975).

El objetivo práctico de este análisis consiste en determinar las medidas que pueden adoptarse para mejorar la calidad la interfase de SQL.

¿ A que problemas se enfrenta el usuario del SQL ?

¿ Qué errores cometen los usuarios del SQL ?

¿ De que forma se podría incrementar la facilidad de uso del SQL ?

¿ De que manera se pueden reducir los errores que cometen los usuarios del SQL ?

## ANTECEDENTES

### 6.0 FUENTES DE ERROR EN LA ETAPA DE CODIFICACION DE INTERROGANTES EN SQL.

A continuación se presenta un análisis de los errores que cometieron los usuarios de SQL del estudio de Reisner (1977).

#### 6.1 ERRORES DE TIPO SINTACTICO.

Como es de esperarse, los errores de tipo sintáctico surgen en virtud de que los usuarios:

- 1) combinan erróneamente las cláusulas;
- 2) omiten una cláusula que funge como requisito para la introducción de otra cláusula.

Ejemplos del primer caso serían:

```
select ... between *  
from  
where
```

\* la cláusula between no puede utilizarse conjuntamente con la palabra select.

del segundo caso:

```
select  
from  
where  
group by *
```

\* para usar la cláusula group by se requiere declarar previamente una operación agregada. (en conjunción con la cláusula select)

#### 6.2 ERRORES DE TIPO LEXICOGRAFICO AL TRANSCRIBIR LAS CLAUSULAS DEL SQL.

Los errores de tipo lexicográfico pueden generarse durante el recuerdo de cláusulas de SQL en la etapa de codificación de interrogantes.

El tipo de errores encontrados son:

1. Errores de terminación, por ejemplo se escribe "grouped by" en vez de "group by".
2. Sinónimos u otros sustitutos, por ejemplo se utiliza "ave", "average" o "avg." en vez de "avg".
3. De deletreo, por ejemplo se escribe "dec" por "desc".

Estos errores se cometen por usuarios cuya lengua materna es el inglés, y podríamos suponer que se generaran los mismos tipos de errores en los sujetos de habla hispana. A estos errores habría que añadir los que surjan a raíz del desconocimiento del idioma inglés, por lo que se incrementarían considerablemente el tipo y la tasa de errores que cometerían sujetos hispanoparlantes.

## ANTECEDENTES

### 6.3 ERRORES DE TIPO LEXICOGRAFICO AL ASIGNAR NOMBRES A LAS TABLAS Y ATRIBUTOS.

Al transcribir los nombres de las tablas y de sus atributos pueden ocasionarse errores muy costosos.

Algunos errores susceptibles de cometerse al recordar los nombres de tablas y atributos son:

1. de terminación (ej: nombres por nombre)
2. sinónimos (ej: docentes por académicos)
3. deletreo (ej: investigaciones por investigaciones)
4. errores en el tipo de datos:

1. Los usuarios tratan a dos atributos idénticos con distintos nombres como diferentes; (ej. docente, investigador, cuando en realidad forman parte de la misma unidad: académicos).
2. Los usuarios tratan a dos atributos con los mismos dominios pero pertenecientes a contextos diferentes como iguales, por ejemplo, número (de la tabla académico) y número (de la tabla departamentos) cuando en realidad son diferentes.

Estos errores producen pérdida de tiempo y reflejan cierta inconsistencia en la conceptualización de la base de datos.

Resumiendo, los errores que pueden ser cometidos en la etapa de codificación de interrogantes en SQL son:

- (1) errores de tipo léxicográfico al escribir las cláusulas del SQL y nombres de tablas y atributos.
- (2) errores sintácticos al combinar erróneamente las cláusulas.

Ambos tipos de errores repercuten en la adaptación individual del usuario al sistema, por lo que es de suma importancia mejorar la interfase estándar del SQL.

A partir de los análisis de estos errores, Torres y Arao (1986, 1987) diseñaron una interfase que tiene por objetivo el acrecentar la facilidad de uso del SQL. Aunque la interfase se diseñó originalmente para sujetos hispano-parlantes, la codificación de colores y las reglas de combinación son potencialmente útiles para cualquier usuario del SQL.

### 7.0 CARACTERISTICAS DE LA INTERFASE PARA SQL.

Como parte del proyecto del sistema de información académica de la Facultad de Psicología, UNAM se instrumentó una interfase para SQL (Torres y Arao, 1986).

La interfase se compone por dos plantillas, en la primera (plantilla SQL) se identifica a las cláusulas del SQL mediante colores y en la segunda los nombres de tablas y atributos.

## ANTECEDENTES

### 7.1 LOS OBJETIVOS DE LA INTERFASE SON LOS SIGUIENTES:

La interfase es un intento para mejorar la calidad de un modo de diálogo entre sistema y usuario. En el contexto de producción de interrogantes, la interfase esta orientada a optimizar la etapa de codificación del problema (véase Fig. 2)

Los objetivos de la interfase para el SQL son los siguientes:

1. Reducir el tiempo de codificación de interrogantes en SQL.
2. Minimizar los errores de tipo lexicográfico en la generación de las cláusulas y de nombres de tablas y atributos.
3. Minimizar los errores de tipo sintáctico

rojo	amari- llo	azul	azul medio	azul medio	azul medio	cafe	cafe medio	rojo medio	rojo claro	rojo claro	rojo claro	rojo claro	rojo claro	rojo claro	rojo claro/ verde	verde/ rojo claro
select	from	where	between	and	or	order by	desc	unique	count (*)	max()	min()	avg()	sum()	group by	having	
PF1	PF2	PF3	PF4	PF5	PF6	PF7	PF8	PF9	PF10	PF11	PF12	PF13	PF14	PF15	PF16	

### PLANTILLA DE SQL

amarillo	amarillo	amarillo
aca	dep	
amarillo	amarillo	amarillo
amarillo	amarillo	amarillo
amarillo		
	num	nom

Tablero de movimientos del cursor

### PLANTILLA DE NOMBRES DE TABLAS Y ATRIBUTOS

FIGURA 3. Interfase para el SQL.

## ANTECEDENTES

### 7.2 PLANTILLA SQL.

Esta plantilla se ajusta al teclado de funciones de la terminal OT-80 que consta de 16 teclas de funciones (teclas PF) en la parte superior del teclado. Al oprimir alguna tecla de función PF se genera el nombre de la cláusula sin la necesidad de escribir carácter por carácter. (véase fig. 3). Cabe mencionar que solo se requieren modificaciones menores para utilizar esta interfase en teclado del tipo IBM-PC-AT los cuales constan de 14 teclas PF y que son de uso común en el mercado.

La Plantilla incluye las cláusulas más usuales del SQL, que están organizadas de la forma siguiente:

- la posición de las cláusulas en la plantilla obedece al orden en que éstas se declaran en los interrogantes de SQL.
- las cláusulas se identifican por colores que proporcionan indicios de la sintaxis del SQL. Para este efecto, el usuario se entrena en el manejo de una sintaxis de colores diseñada a partir de la sintaxis del SQL y de esta forma el usuario se introduce intuitivamente a la sintaxis del lenguaje. (véase Apéndice D para la sintaxis de los colores y Apéndice G para el procedimiento que se utilizó en el diseño de la interfase).

Por ejemplo, en el entrenamiento de la sintaxis de colores, el usuario aprende que la plantilla tiene tres colores básicos, (rojo, amarillo y azul) y que estos tres colores se utilizan en ese orden en renglones separados.

equivalentes

rojo	select
amarillo	from
azul	where

El usuario también aprende que el color azul, tiene como variante el color azul medio, el cuál debe de ser utilizado exclusivamente con el azul.

equivalencia válida

rojo	select
amarillo	from
azul azul medio	where or

equivalencia inválida

rojo azul medio	select or
amarillo	from
azul	where

## ANTECEDENTES

### 7.3 PLANTILLA DE NOMBRES DE TABLAS Y ATRIBUTOS.

El complemento de la plantilla de las cláusulas de SQL consiste de otra plantilla que tiene definidos nombres de tablas y atributos. Sin embargo, la plantilla con colores puede usarse de forma independiente a la segunda plantilla.

La plantilla de nombres de atributos se puede diseñar de la forma siguiente:

Se utiliza el tablero de controles del cursor de la terminal QT-SO compuesto de doce teclas, en las cuales se definen las abreviaturas de nombres de tablas y dominios de atributos. (véase fig. 3)

Los nombres de tablas y atributos se deben generar de acuerdo con una regla de abreviación que permite identificar con exactitud la naturaleza y procedencia de los atributos.

Ahora bien, ¿cual es el efecto de la interfase en el aprendizaje del SQL ?

¿ Qué tipo de ayuda proporcionan los teclados de la interfase como dispositivo de entrada para el usuario ?

¿ Qué tan efectivo es el uso del color como un indicio redundante en el aprendizaje y uso de la sintaxis del SQL ?

El presente trabajo tiene por objetivo evaluar esta interfase que ha sido diseñada para mejorar la interfase estándar del SQL y contestar las preguntas planteadas anteriormente.

## METODO

### II. METODO

#### 1.0 SUJETOS.

Se entrenó a 20 sujetos voluntarios cuarto semestre de la Licenciatura de la Facultad de Psicología, UNAM, a los cuales se le ofreció un punto en la calificación final de una materia por su participación. De los 20 sujetos 14 eran del sexo femenino y 6 del masculino, el promedio de edad era de 22 años, cabe agregar que los sujetos no tenían experiencia en programación. Los sujetos se asignaron a dos grupos de 10.

#### 2.0 EQUIPO Y MATERIALES.

2.1 Lenguaje de Interrogación Estructurado (SQL) Versión 3.1, instalado en el sistema QNYX 8002-A.

2.2 Interfase para el SQL.

2.3 Cuadernillo de notas de SQL, se utilizaron dos versiones que se diferenciaban únicamente en los nombres de tablas y atributos utilizados en los ejemplos.

2.4 Programa de computadora para la tarea de Verificación.

2.5 Cronómetro.

#### 3.0 DISEÑO EXPERIMENTAL.

Se utilizó un diseño de un factor entre sujetos y dos factores intra sujetos.

Factor Entre: Tipo de entrenamiento, con interfase y sin ella. A cada condición le correspondió un grupo de sujetos.

Factor Intra 1: Tipos de interrogante (selección, proyección, ordenamiento, Único, operaciones agregadas, agrupamiento y selección de grupos).

Factor Intra 2: Periodos de evaluación, un día después y una semana posterior al entrenamiento.

Grupo Experimental										Grupo Control										Factor Entre
s	p	o	ú	o	a	s	s	p	o	ú	o	a	s		Factor Intra 1					
e	r	r	n	p	r	e	e	r	r	n	p	r	e							
l	e	d	i	c	a	u	l	e	d	i	c	a	u							
c	e	n	o	g	p	g	c	e	n	o	g	p	g							
c	c	a		r	a	r	c	c	a		r	a	r		Factor Intra 2					
i	c	m		e	m	u	i	c	m		e	m	u							
u	u	u	u	u	u	u	u	u	u	u	u	u	u							
n	n	n	n	n	n	n	n	n	n	n	n	n	n							
a	a	a	a	a	a	a	a	a	a	a	a	a	a							
d	d	d	d	d	d	d	d	d	d	d	d	d	d							
í	í	í	í	í	í	í	í	í	í	í	í	í	í							
a	a	a	a	a	a	a	a	a	a	a	a	a	a							

## METODO

### 4.0 PROCEDIMIENTO

#### 4.1 FASE DE AMBIENTACION.

Durante un periodo de 30 minutos los sujetos del grupo experimental recibieron entrenamiento en el manejo de la interfase. El instructor indicaba a los usuarios que leyeran el instructivo de colores (ver apéndice D) inciso por inciso, al término de cada inciso, el instructor se aseguraba de que las reglas sintácticas de colores estudiadas por los usuarios quedaran entendidas. Asimismo, se entrenó a estos usuarios en el uso de la plantilla de nombre de tablas y atributos (aprendieron las reglas para generar los nombres de tablas y atributos).

A los sujetos de la condición control se les compensó ese tiempo describiéndoles las tendencias actuales de los interrogadores a bases de datos.

#### 4.2 FASE EXPERIMENTAL.

Se entrenó a dos grupos de 10 usuarios en el manejo de la sintaxis básica del SQL. Un grupo de usuarios se entrenó con la plantilla de SQL y la plantilla de nombres de tablas y atributos (Grupo Experimental), y el otro entrenó sin plantillas.

La duración del entrenamiento fué de ocho horas, dos horas por cuatro días consecutivos. Ambos grupos de usuarios fueron entrenados por el mismo instructor, y vieron el mismo temario. Cada grupo de usuarios se subdividió en dos, de tal manera que se entrenó a los usuarios en grupos de cinco.

La forma en la que se impartió el curso fué la siguiente; la sintaxis del SQL fue enseñada por rasgos (tipos de interrogante). Los usuarios leían en el manual (véase Apéndice E) la parte correspondiente al tipo de interrogante en estudio, el instructor explicaba el tipo de situaciones en las que se utilizaba el tipo de interrogante y explicaba conceptualmente (en papel) la secuencia de operaciones que se realizaban en cada interrogante, (modelo algorítmico según Schlagel y Ogden (1986)), para este efecto, el instructor se valió de unas cartulinas para simular las secuencias; en estos momentos se esta en la fase final de un programa en PASCAL con el mismo objetivo) y explicaba la sintaxis del interrogante.

A los usuarios del grupo experimental se les asociaba las reglas sintácticas del interrogante con la plantilla de colores que tenían en el tablero de las terminales QT-80. Después de la explicación en papel, los usuarios realizaban ejercicios en línea.

## 4.3 TAREAS

## 4.3.1 VERIFICACION.

En esta tarea el sujeto verificó el valor de verdad de una serie de interrogantes.

Ejemplo: [interrogante de agrupamiento]

```
select unique, depniv
from dep
group by depniv/
```

Para esta tarea se contó con un programa hecho en PASCAL que presentaba los interrogantes al azar (para cada sujeto), registraba las respuestas y los tiempos de verificación (miliseg.).

La tarea consistió de 42 interrogantes experimentales (21 verdaderos y 21 con errores de sintaxis; 3 por cada tipo de interrogante) los sujetos tuvieron una sola oportunidad para verificar cada interrogante.

Esta tarea estuvo orientada completamente a la sintaxis del SQL. Los sujetos verificaron la sintaxis de una serie de interrogantes. En esta tarea se manipularon selectivamente las posibles fuentes de error en la sintaxis del SQL (véase Apéndice H).

## 4.3.2 CONSTRUCCION.

Dado un interrogante en español los sujetos elaboraron el interrogante correspondiente en SQL.

[interrogante de proyección]

¿ Cuáles son los nombres de los académicos que tienen el grado de doctor ? (interrogante en español)

interrogante en SQL

sql> ???

[interrogante de agrupamiento]

¿ Cuántos departamentos están adscritos a cada una de las divisiones de la Facultad ? (interrogante en español)

Interrogante en SQL

sql> ???

La tarea consistió en 14 preguntas experimentales y tres de práctica (preguntas en español a ser traducidas a SQL), los sujetos contaron con dos oportunidades para elaborar el interrogante en SQL. En el caso de que el SQL reportara un error, el instructor traducía el mensaje emitido por la máquina sin insinuar la solución. Los errores de tipo lógico de los usuarios fueron ignorados por el instructor.

## METODO

Los usuarios leían el interrogante en español para posteriormente traducirlo a SQL, el tiempo de construcción contempló desde el momento que el usuario empezaba a escribir el interrogante hasta que lo finalizaba (véase apéndice I).

### 4.4.3 ANALISIS DE LAS TAREAS.

Las tareas de verificación y construcción fueron elegidas para la evaluación, por la razón de que el usuario realiza estas tareas en el contexto real de programación; identifica errores de sintaxis y, elaboran interrogantes dado un interrogante en español.

### 4.4 FASE DE EVALUACION.

La evaluación fué automática para la tarea de Verificación, para la tarea de construcción el instructor llevó un registro manual de la evaluación.

Los sujetos entrenados con la interfase, disponieron de ella durante la evaluación.

A todos los sujetos se les aplicó las tareas de Verificación y Construcción.

En las tareas no se permitió recurrir a las notas. Dentro de ambas tareas se diferencian los tipos de interrogante del lenguaje: unión, proyección, selección, agrupamiento, etc. (véase Apéndice C).

### 4.5 PERIODOS DE EVALUACION

Se evaluó a los sujetos fueron evaluados individualmente un día después del entrenamiento y una semana posterior al entrenamiento. Se citó a los sujetos una semana después, sin insinuarles que se les aplicaría una segunda evaluación, las preguntas de la tarea de construcción y los interrogantes de la verificación fueron los mismos.

### 4.6 VARIABLE INDEPENDIENTE.

La variable independiente manipulada en este estudio consistió en la interfase para el SQL.

### 4.7 VARIABLES DEPENDIENTES.

#### 4.7.1 Tiempo para verificar y construir interrogantes.

Para la tarea de verificación, el programa registró el tiempo en milisegundos para cada interrogante.

En la tarea de construcción, el instructor registró el tiempo en segundos para cada interrogante elaborado por los sujetos.

#### 4.7.2 Número de repuestas correctas/incorrectas por tipo de interrogante.

## METODO

En la tarea de verificación el programa registró las respuestas de los sujetos.

En la tarea de construcción, el instructor registró los aciertos o las fallas de los sujetos.

### 4.7.3 Tipo y Cantidad de errores cometidos .

En la tarea de construcción los interrogantes en SQL fueron almacenados en archivos para un análisis de los errores.

Los tipos de errores encontrados fueron los siguientes:

1. Errores de tipo sintáctico. El sujeto utilizaba inadecuadamente las cláusulas.
2. Errores de tipo lógico. El sujeto no declaraba una cláusula clave en el interrogante.
3. Errores menores. Un error menor se definió como aquél que no repercutía gravemente en la construcción del interrogante, por ejemplo el sujeto se equivocaba al indicar signos de puntuación.

## RESULTADOS

### III. RESULTADOS.

Se registraron dos variables dependientes: cantidad de aciertos y tiempos en las tareas de construcción y verificación de interrogantes. Cada variable dependiente fue analizada por tarea, con análisis de varianza de un factor entre y dos factores intra (Grupos X Interrogante X Intervalos de Evaluación). También se registró el tipo de errores que se cometieron en la tarea de construcción, pero este análisis se presenta por separado.

#### 1.0 TAREA DE CONSTRUCCION.

##### 1.1 ACIERTOS.

TABLA 3. Análisis de varianza para los aciertos en la tarea de construcción.

DISEÑO DE TRES FACTORES UN FACTOR ENTRE /DOS FACTORES INTRA					
FUENTE	SUMA DE CUADRADOS	G. L.	MEDIA DE CUADRADOS	RAZON F.	p
ENTRE SUJETOS	22.21	19			
Grupos	4.38	1	4.38	4.42	< .05
Error Entre	17.84	18	0.99		
INTRA SUJETOS	124.21	260			
Interrogantes	56.00	6	9.33	27.31	< .01
Grupos X interrogantes	4.80	6	0.80	2.34	< .05
Error interrogantes	36.91	108	0.34		
Intervalos evaluación	1.89	1	1.89	13.96	< .01
Grupos X Intervalos	0.17	1	0.17	1.29	
Error Intervalos	2.44	18	0.14		
Interrog X Intervalos	2.69	6	0.45	2.58	< .05
Grup X Interr X Interv	0.60	6	0.10	0.58	
Error Interr_Interval	18.71	108	0.17		

Los aciertos se analizaron con un análisis de varianza de 2 X 7 X 2. (Grupos X Interrogantes X Intervalos de Evaluación). Los grupos, interrogantes e interacción (grupos X interrogante) tuvieron un efecto significativo  $F(1,18) = 4.42$ ,  $p < .05$ ,  $F(6,108) = 27.31$ ,  $p < .01$ , y  $F(6,108) = 2.34$ ,  $p < .05$ . El grupo interfase obtuvo mayor número de aciertos que el grupo control, particularmente, en el interrogante de selección de grupos  $t(98) = 2.36$ ,  $p = .01$  (véase fig. 4); en esta figura, se observa que en cada grupo, ciertos interrogantes fueron más difíciles de construir; en el grupo control, los interrogantes de agrupamiento y selección de grupos; y, en el grupo interfase, el interrogante de agrupamiento. Por otro lado, el efecto de los intervalos, así como el efecto de la interacción (interrogantes X intervalos) fueron significativos,  $F(1,18) = 13.96$ ,  $p < .01$ , y  $F(6,108) = 2.58$ ,  $p < .05$ , mediante el análisis a posteriori se observó que los sujetos de ambos grupos obtuvieron más aciertos en el interrogante de selección de grupos de la primera a la segunda evaluación  $t(121) = 2.35$ ,  $p = .01$  (ver fig. 5).



Tipos de Interrogante

■ Control □ Interfase

Figura 4. Tarea de construcción: Promedio de aciertos por grupo y por tipo de interrogante

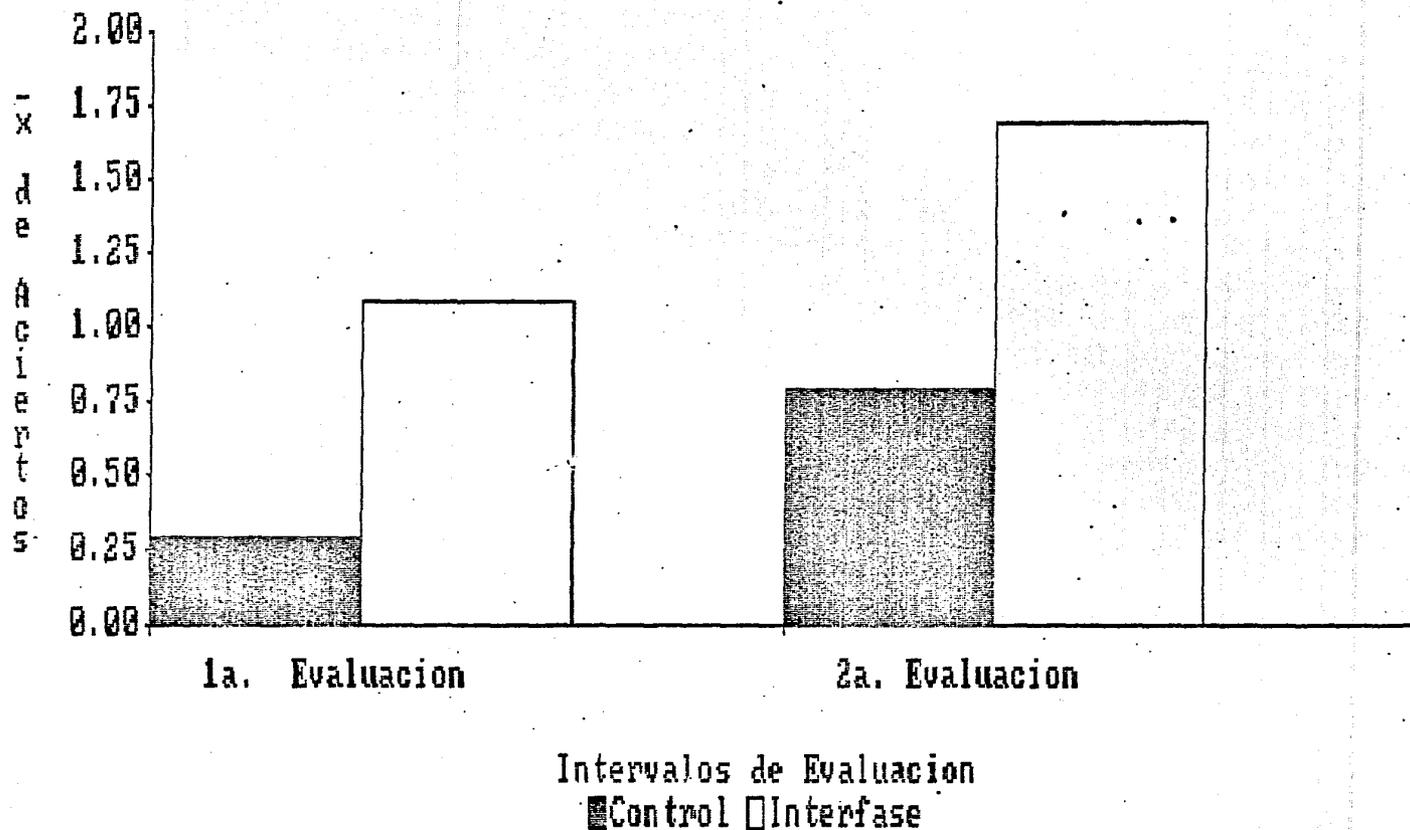


Figura 5. Tarea de construcción: Promedio de aciertos por grupo en el interrogante de selección de grupos

## 1.2 TIEMPOS.

TABLA 4. Análisis de varianza para los tiempos en la tarea de construcción.

DISEÑO DE TRES FACTORES UN FACTOR ENTRE / DOS FACTORES INTRA					
FUENTE	SUMA DE CUADRADOS	G. L.	MEDIA DE CUADRADOS	RAZON F	p
ENTRE SUJETOS	711139.17	19			
Grupos	525954.01	1	525954.01	51.12	< .01
Error Entre	185185.16	18	10288.06		
INTRA SUJETOS	3720084.57	540			
Interrogantes	615213.84	13	47324.14	7.14	< .01
Grupos X interrogantes	459763.39	13	35366.41	5.34	< .01
Error interrogantes	1550945.34	234	6627.97		
Intervalos evaluación	92983.31	1	92983.31	30.01	< .01
Grupos X Intervalos	39211.78	1	39211.78	12.66	< .01
Error Intervalos	55773.19	18	3098.51		
Interrog X Intervalos	97786.69	13	7521.59	2.30	< .
Grup X Interr X Interv	43324.72	13	3332.67	1.02	
Error Interr_Interval	765088.31	234	3269.61		

Los tiempos se analizaron con un análisis de varianza de  $2 \times 14 \times 2$ . (Grupos X Interrogantes X Intervalos de Evaluación). El grupo control invirtió mas tiempo en la construcción de interrogantes que el grupo interfase  $F(1,18) = 51.12$ ,  $p < .01$  (véase fig. 6). Asimismo, el efecto de los intervalos de evaluación y el efecto de la interacción (grupos X intervalos) fueron significativos  $F(1,18) = 30.01$ ,  $p < .01$  y  $F(1,18) = 12.66$ ,  $p < .01$ . En la figura 6 se observan como el grupo control utilizó menos tiempo de la primera a la segunda evaluación  $t(27) = 2.47$ ,  $p = .01$ , mientras que el grupo interfase utilizó aproximadamente el mismo tiempo en ambas evaluaciones, es importante notar que, aunque el grupo control disminuyó su tiempo de la primera a la segunda evaluación no igualó al grupo interfase.

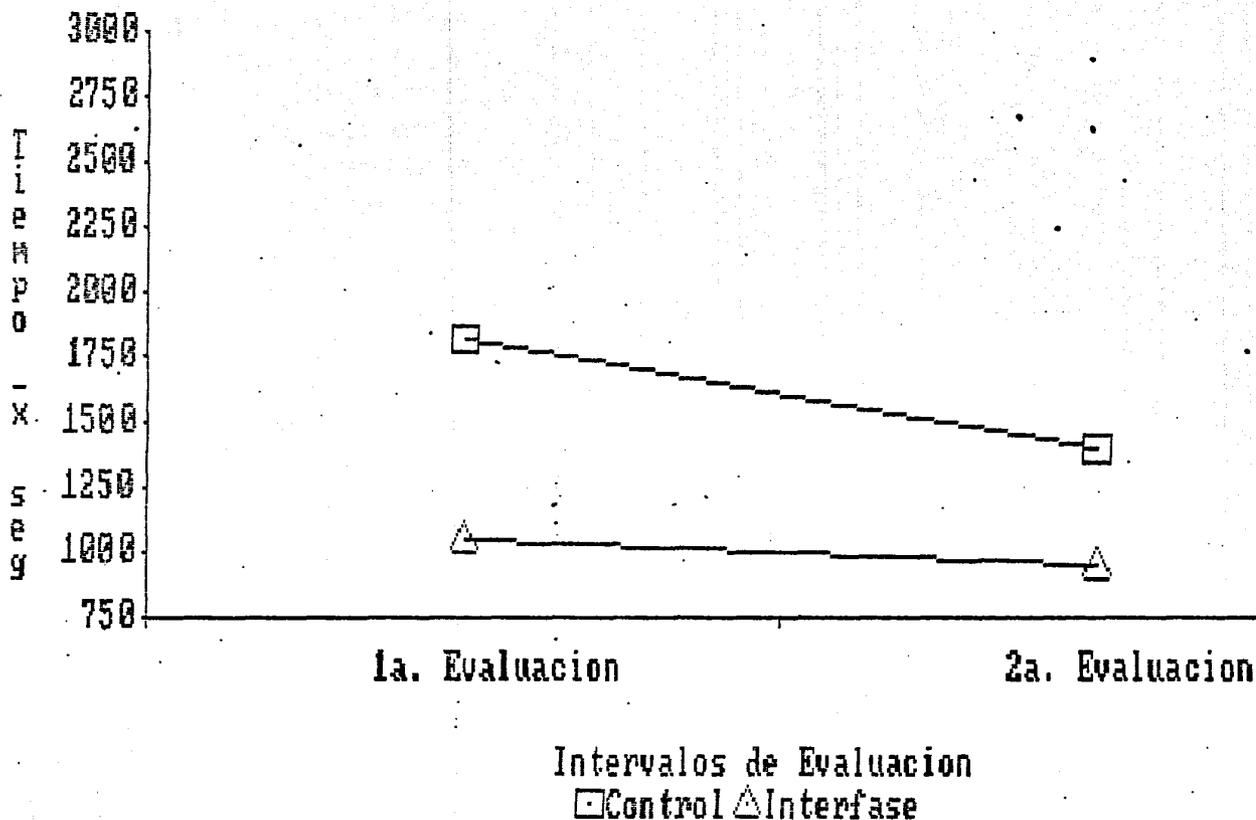


Figura 6. Tiempos promedio de construcción de interrogantes, por grupo en la primera y segunda evaluación

## RESULTADOS

### 1.3 ERRORES.

#### 1.3.1 ERRORES MENORES.

Un error se considero como menor cuando no repercutía gravemente en la estructura sintáctica o lógica del interrogante.

Los errores encontrados fueron:

- 1) Errores de puntuación, los usuarios reemplazaban signos, algunas veces por similitud física por ej. apostrofo invertido ['] por ['']; o por cercanía espacial [?] por [/], (estos signos están en la misma tecla pero se diferencian oprimiendo la tecla shift).
- 2) Errores lexicográficos en la enunciación de tablas o atributos, ej. grados por grado, salarios por salario.
- 3) Errores en la ubicación de atributos en tablas.
- 4) Errores lexicográficos en la enunciación de cláusulas de SQL, ej. "wehere" por "where", "frem" por "from".
- 5) Traducción de las cláusulas del inglés al español, ej. "descendentemente" por "desc", "y" por "and".

En la tabla 5 se presenta el porcentaje de sujetos que cometieron estos errores por lo menos una vez. En esta tabla podemos observar dos puntos importantes: 1) los errores de puntuación fueron cometidos por los sujetos de ambos grupos. 2) los sujetos del grupo interfase no cometieron los demás tipos de errores.

## RESULTADOS

TABLA 5. Errores menores cometidos por los usuarios en la tarea de construcción de interrogantes.

Porcentaje de sujetos que cometieron por lo menos un error de cierta clase durante la primera y segunda evaluación

Tipo de Error	Grupo Control		Grupo Interfase	
	1a Evalua	2a Evalua	1a Evalua	2a Evalua
1:	50	30	50	10
2	80	20	0	0
3:	30	30	0	0
4:	40	0	0	0
5:	10	10	0	0

- 1: errores de puntuación.
- 2: errores lexicográficos en la transcripción de nombre de tablas y atributos.
- 3: errores por la ubicación de atributos en tablas diferentes.
- 4: errores lexicográficos en la transcripción de cláusulas del SQL.
- 5: errores lexicográficos por la traducción de los nombres de las cláusulas del inglés al español.

## RESULTADOS

### 1.3.2 ERRORES DE TIPO LOGICO.

Un error se consideró como lógico cuando el usuario no declaró una cláusula clave para el interrogante (véase tabla 5).

Los errores de tipo lógico encontrados fueron:

- 1) Omisión de la cláusula de "group by" en el interrogante de agrupamiento, este error fué muy común en ambos grupos de sujetos, y fue determinante para que los sujetos fallaran en la construcción de interrogantes de agrupamiento.
- 2) Omisión de las cláusulas de "group by" y "having" en el interrogante de selección de grupos.

Es importante hacer énfasis en que estos errores afectaron igualmente a los sujetos entrenados con la interfase y sin ella, particularmente el error de omitir la cláusula de "group by" en el interrogante de agrupamiento fue el que determinó que en ambos grupos el interrogante de agrupamiento fuera el más difícil. El hecho de que estos errores se hayan dado indistintamente en ambos grupos implica que la interfase no ayudo a evitar este tipo errores; esto es razonable, dado que la interfase esta diseñada para ofrecer unicamente ayuda de tipo sintáctico; este punto será revisado con más detalle en la discusión.

TABLA 6. Errores de tipo lógico cometidos por los usuarios en la tarea de construcción.

Porcentaje de sujetos que cometieron por lo menos un error de cierta clase por tipo de interrogante, durante la primera y segunda evaluación

Tipo de Interrogante	Grupo Control		Grupo Interfase	
	1a Evalua	2a Evalua	1a Evalua	2a Evalua
Agrupamiento 1:	70	40	70	70
Selección 2: de Grupos	20	20	10	10

1: omiten la cláusula "group by" en el interrogante.

2: omiten las cláusulas "group by" y "having".

## RESULTADOS

### 1.3.3 ERRORES DE TIPO SINTACTICO.

Los errores sintácticos encontrados fueron:

- 1) Omisión de la cláusula "from" del interrogante, Ejemplo:

incorrecto	correcto
.....	.....
select grado	select grado
academ/	from academ/

- 2) Omisión de la cláusula "and" al utilizar between, Ejemplo:

incorrecto	correcto
.....	.....
select grado	select grado
from academ	from academ
where hijos between 1 4/	where hijos between 1 and 4/

- 3) Omisión de la cláusula "where" del interrogante, Ejemplo:

incorrecto	correcto
.....	.....
select grado	select grado
from academ	from academ
hijos = 1/	where hijos = 1/

- 4) Omisión del atributo por el cual se ordena, Ejemplo:

incorrecto	correcto
.....	.....
select grado, nombre	select grado, nombre
from academ	from academ
order by desc	order by grado desc/

- 5) Invertir el orden de "unique"\_\_atributo, Ejemplo:

incorrecto	correcto
.....	.....
select grado unique	select unique grado
from academ/	from academ/

- 6) Reemplazar la cláusula "order by" por "group by", Ejemplo:

incorrecto	correcto
.....	.....
select grado, avg(hijos)	select grado, avg(hijos)
from academ	from academ
order by grado/	group by grado/

## RESULTADOS

- 7) Utilizar la cláusula de "having" sin declarar previamente "group by", Ejemplo:

incorrecto	correcto
.....	.....
select grado,avg(hijos)	select grado, avg(hijos)
from academ	from academ
having avg(hijos)> 2/	group by grado
	having avg(hijos)> 2/

- 8) Utilizar operaciones agregadas junto a la cláusula de "where", Ejemplo:

```
select grado,avg(hijos)
from academ
where avg(hijos)> 2/
```

En la tabla 7 se presenta la cantidad de sujetos que obtuvieron por lo menos alguno de estos errores. Es importante mencionar que: 1) los errores correspondientes a omitir el atributo por el cual se ordena e invertir el orden unique\_atributo son los únicos que no son atacados por la plantilla de colores del SQL, y 2) El error al utilizar la cláusula de "having" sin declarar previamente la cláusula de "group by" y el error de utilizar las operaciones agregadas junto a la cláusula de where fueron determinantes para que se estableciera la diferencia en aciertos del grupo control e interfase en el interrogante de selección de grupos.

## RESULTADOS

**TABLA 7. Errores de tipo sintáctico cometidos por los usuarios en la tarea de construcción.**

Porcentaje de sujetos que cometieron por lo menos un error de cierta clase durante la primera y segunda evaluación

Tipo de Error	Grupo Control		Grupo Interfase	
	1a Evalua	2a Evalua	1a Evalua	2a Evalua
1:	40	0	20	0
2:	20	0	0	0
3:	0	0	20	0
4:	20	10	20	10
5:	30	0	10	0
6:	20	30	10	0
7:	20	50	0	0
8:	40	20	20	0

- 1: omiten la cláusula from del interrogante
- 2: omiten la cláusula and al utilizar la cláusula between
- 3: omiten la cláusula where para establecer la condición de selección
- 4: omiten el atributo por el cual se ordena
- 5: invierten el orden unique atributo (atributo unique)
- 6: utilizan order by por group by
- 7: utilizan la cláusula having sin declarar previamente group by
- 8: utilizan operaciones agregadas junto a la cláusula where

## RESULTADOS

### 2.0 TAREA DE VERIFICACION.

#### 2.1 ACIERTOS.

TABLA 8. Análisis de varianza para los aciertos en la tarea de verificación.

DISEÑO DE TRES FACTORES UN FACTOR ENTRE /DOS FACTORES INTRA					
FUENTE	SUMA DE CUADRADOS	G. L.	MEDIA DE CUADRADOS	RAZON F	p
ENTRE SUJETOS	15.92	19			
Grupos	0.80	1	0.80	0.96	
Error Entre	15.12	18	0.84		
INTRA SUJETOS	119.79	260			
Interrogantes	11.04	6	1.84	4.24	< .01
Grupos X interrogantes	8.42	6	1.40	3.24	< .01
Error interrogantes	46.83	108	0.43		
Intervalos evaluación	4.89	1	4.89	12.26	< .01
Grupos X Intervalos	0.29	1	0.29	0.73	
Error Intervalos	7.18	18	0.40		
Interrog X Intervalos	2.24	6	0.37	1.08	
Grup X Interr X Interv	1.54	6	0.26	0.74	
Error Interr_interval	37.37	108	0.35		

Los aciertos fueron analizados con un análisis de varianza de  $2 \times 7 \times 2$ . (Grupos X Interrogantes X Intervalos de Evaluación). Los grupos control e interfase obtuvieron la misma cantidad de aciertos,  $F(1,18) = 0.96$ . Este resultado era de esperarse, dado que no había ninguna razón para esperar que ambos grupos difirieran en este resultado. Ahora bien, el efecto de los interrogantes, y de la interacción (grupos X interrogante) fueron significativos,  $F(6,108) = 4.24$ ,  $p < .01$  y  $F(6,108) = 3.24$ ,  $p < .01$ . En la figura 7 se ilustran estos efectos. Concretamente, el interrogante de selección de grupos control fue el más difícil de verificar por los sujetos del grupo control  $t(109) = 2.36$ ,  $p = .01$ . Por otro lado, el efecto de los intervalos fue significativo, ambos grupos de sujetos obtuvieron más aciertos en la segunda evaluación  $F(1,18) = 12.26$ ,  $p < .01$ .

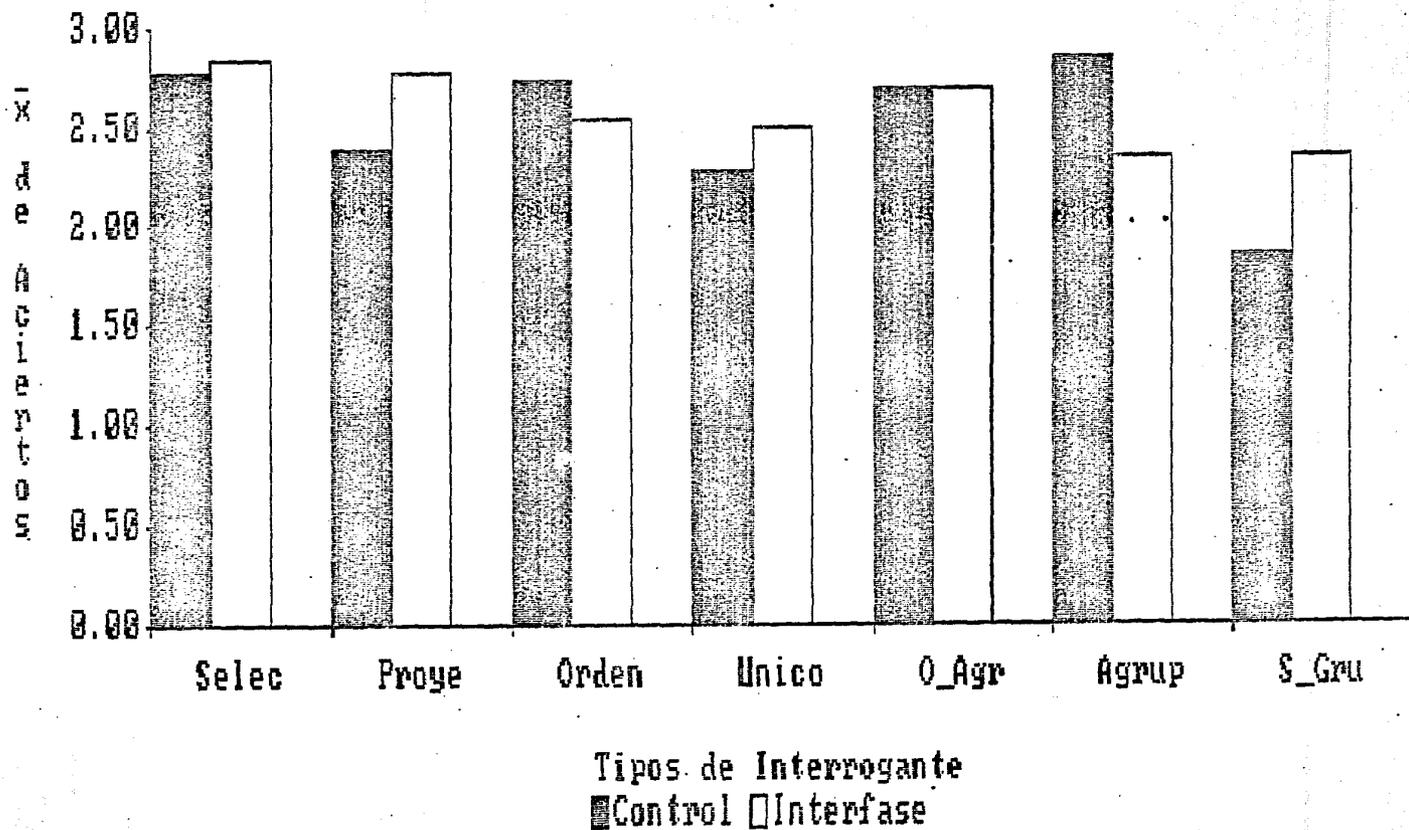


Figura 7. Tarea de verificación: Promedio de aciertos por grupo y por tipo de interrogante

## RESULTADOS

### 2.2 TIEMPOS.

TABLA 9. Análisis de varianza para los tiempos en los aciertos de la tarea de verificación.

DISEÑO DE TRES FACTORES UN FACTOR ENTRE /DOS FACTORES INTRA					
FUENTE	SUMA DE CUADRADOS	G. L.	MEDIA DE CUADRADOS	RAZON F	p
ENTRE SUJETOS	342274878.85	19			
Grupos	3614953.60	1	3614953.60	0.19	
Error Entre	338659925.25	18	18814440.29		
INTRA SUJETOS	1323672957.60	820			
Interrogantes	167797076.78	20	8389853.84	5.50	< .01
Grupos X interrogantes	21515971.28	20	1075798.56	0.71	
Error interrogantes	548660449.00	360	1524056.80		
Intervalos evaluación	111963756.70	1	111963756.70	48.22	< .01
Grupos X Intervalos	9948455.03	1	9948455.03	4.28	
Error Intervalos	41792370.78	18	2321798.38		
Interrog X Intervalos	25593592.98	20	1279679.65	1.21	
Grup X Interr X Interv	16532430.95	20	826621.55	0.78	
Error Interr_Interval	379868854.07	360	1055191.26		

Los tiempos fueron analizados con un análisis de varianza de  $2 \times 21 \times 2$ . (Grupos X Interrogantes X Intervalos de Evaluación). Como se ilustra en la figura 8, ambos grupos de sujetos utilizaron el mismo tiempo en la verificación de aciertos  $F(1,18) = 0.19$ , e invirtieron menor tiempo en la segunda evaluación  $F(1,18) = 48.22$ ,  $p < .01$ .

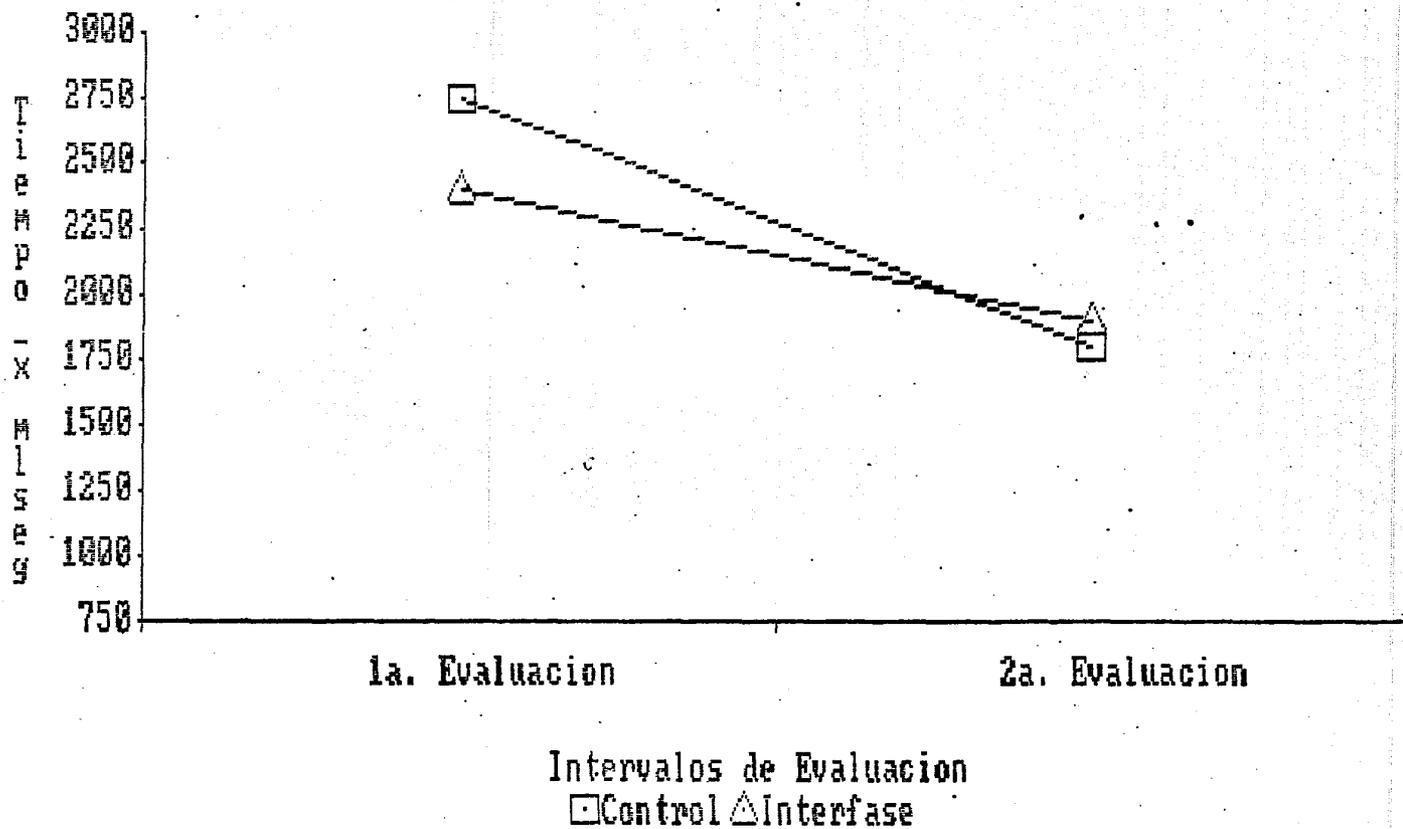


Figura 8. Tiempos promedio de verificación de interrogantes correctos por grupo en la primera y segunda evaluación.

## RESULTADOS

### 3.0 TAREA DE VERIFICACION [RECHAZOS CORRECTOS].

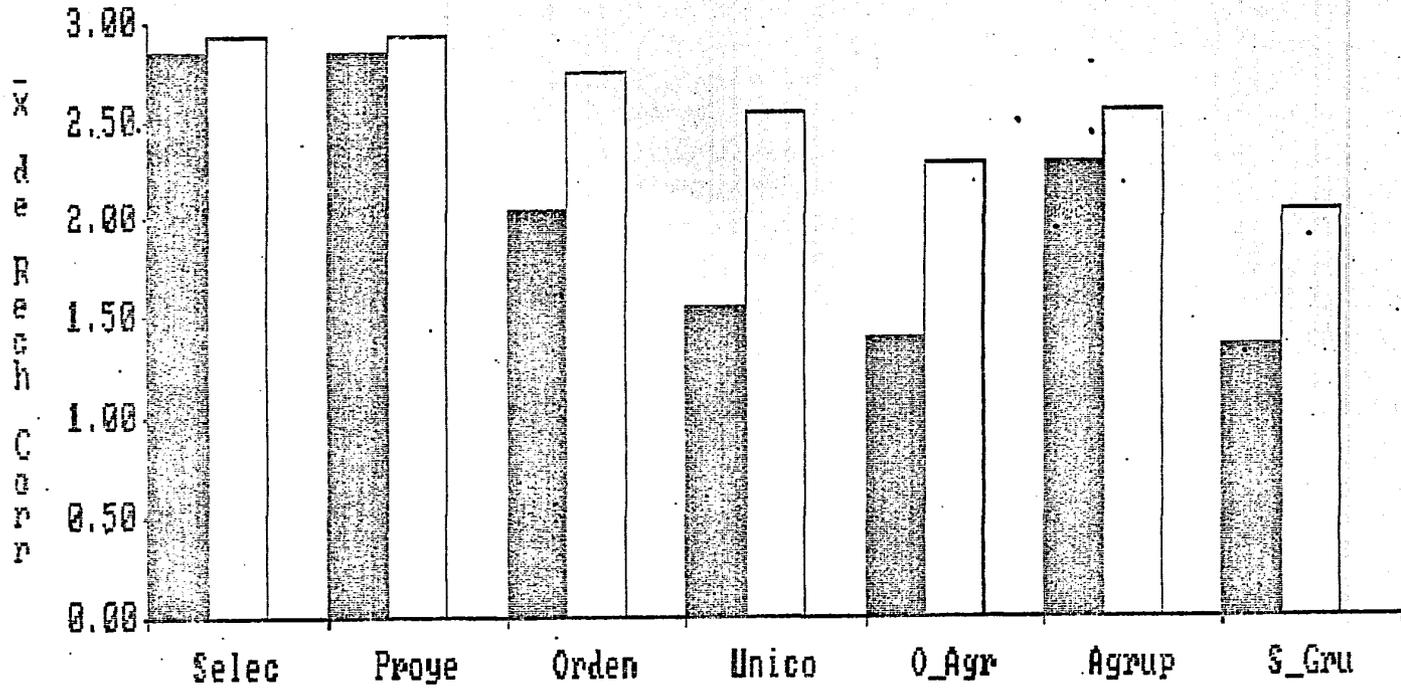
#### 3.1 RECHAZOS CORRECTOS.

TABLA 10. Análisis de varianza para los rechazos correctos en la tarea de verificación.

DISEÑO DE TRES FACTORES UN FACTOR ENTRE / DGS FACTORES INTRA					
FUENTE	SUMA DE CUADRADOS	G. L.	MEDIA DE CUADRADOS	RAZON F	p
ENTRE SUJETOS	21.35	19			
Grupos	11.60	1	11.60	21.42	< .01
Error Entre	9.75	18	0.54		
INTRA SUJETOS	155.64	260			
Interrogantes	69.12	6	11.52	32.15	< .01
Grupos X interrogantes	12.32	6	2.05	5.73	< .01
Error interrogantes	38.70	108	0.36		
Intervalos evaluación	0.29	1	0.29	0.98	
Grupos X Intervalos	0.60	1	0.60	2.04	
Error Intervalos	5.32	18	0.30		
Interrog X Intervalos	2.84	6	0.47	2.02	
Grup X Interr X Interv	1.12	6	0.19	0.80	
Error Interr_Interval	25.33	108	0.23		

Los rechazos correctos fueron analizados con un análisis de varianza de  $2 \times 7 \times 2$ , [Grupos X Interrogantes X Intervalos de Evaluación]. Como se observa en la figura 9, el grupo interfase detecto mas interrogantes con errores de sintaxis que el grupo control  $F(1,18) = 21.42, p < .01$ . El efecto de los interrogantes y el efecto de la interacción [grupos X interrogantes] fueron significativas,  $F(6,108) = 32.15, p < .01$  y  $F(6,108) = 5.73, p < .01$  respectivamente; en la figura 9 también se observa que el grupo interfase obtuvo un mayor número de rechazos correctos que el control en los interrogantes de: ordenamiento, Único, operaciones agregadas y selección de grupos. Ahora bien, en cada grupo determinados interrogantes fueron mas difíciles de verificar que otros; en el grupo control, los interrogantes de selección de grupos, operaciones agregadas y Único; en el grupo interfase, el interrogante de selección de grupos  $t(122) = 2.35, gl 122, p = .01$ .

En la tabla 11 se presenta el porcentaje de sujetos de ambos grupos que rechazaron correctamente los interrogantes con errores de sintaxis, en general se observa que los sujetos del grupo interfase obtuvieron más rechazos correctos que el grupo control.



Tipos de Interrogante  
 ■ Control □ Interfase

Figura 9. Tarea de verificación. Promedio de rechazos correctos por grupo y por tipo de interrogante

RESULTADOS

TABLA 11. Errores cometidos por los usuarios en la tarea de verificación.

Porcentaje de usuarios que rechazaron correctamente interrogantes con errores de sintaxis durante la primera y segunda evaluación

Tipo de Interrogante	Grupo Control		Grupo Interfase	
	1a Evalua	2a Evalua	1a Evalua	2a Evalua
Selección	s1:	90	100	90
	s2:	100	100	100
	s3:	100	80	100
Proyección	p1:	100	100	100
	p2:	90	90	100
	p3:	100	90	100
Ordenamiento	o1:	80	90	100
	o2:	30	10	50
	o3:	100	90	100
Unico	u1:	30	40	80
	u2:	30	20	90
	u3:	100	90	100
Agregadas	a1:	10	0	30
	a2:	100	100	100
	a3:	50	30	90
Agrupamiento	g1:	80	70	100
	g2:	70	50	70
		100	80	90
Selección de Grupos	h1:	70	90	100
	h2:	10	20	50
	h3:	20	30	80

RELACION DE ERRORES

s1: from antes de select  
s2: where antes de from  
s3: where antes de select y from

p1: from antes de select  
p2: where antes de from  
p3: where antes de select y from

o1: order by con group by y having  
o2: order by con unique  
o3: order by despues de where

u1: unique con group by y having  
u2: unique con group by  
u3: where antes de from

a1: agregadas con order by  
a2: where antes de from  
a3: agregadas con unique

g1: group by con order by  
g2: group by sin agregada  
g3: group by antes de where

h1: having despues de where  
h2: having sin operacion agregada  
h3: having sin group by

## RESULTADOS

### 3.2 TIEMPOS.

TABLA 12. Análisis de varianza para los tiempos en los rechazos correctos de la tarea de verificación.

DISEÑO DE TRES FACTORES UN FACTOR ENTRE /DOS FACTORES INTRA					
FUENTE	SUMA DE CUADRADOS	G. L.	MEDIA DE CUADRADOS	RAZON F	p
ENTRE SUJETOS	104744351.21	19			
Grupos	11571179.67	1	11571179.67	2.24	
Error Entre	93173171.54	18	5176287.31		
INTRA SUJETOS	913647016.64	820			
Interrogantes	293633517.98	20	14681675.90	16.83	< .01
Grupos X interrogantes	20416941.56	20	1020847.08	1.17	
Error interrogantes	314003220.61	360	872231.17		
Intervalos evaluación	32698529.00	1	32698529.00	42.90	< .01
Grupos X Intervalos	3019561.63	1	3019561.63	3.96	
Error Intervalos	13720606.11	18	762255.89		
Interrog X Intervalos	12191332.12	20	609566.61	1.03	
Grup X Interr X Interv	11903710.60	20	595185.53	1.01	
Error Interr_Interval	212059597.04	360	589054.44		

Los tiempos en la verificación de rechazos correctos fueron analizados con un análisis de varianza de  $2 \times 21 \times 2$ . (Grupos X Interrogantes X Intervalos de Evaluación). Los grupos control e interfase, utilizaron la misma cantidad de tiempo en la verificación de interrogantes con errores de sintaxis  $F(1,18) = 2.24$ , (véase fig. 10); en la figura 10 también se observa, que ambos grupos utilizaron menos tiempo de la primera a la segunda evaluación  $F(1,18) = 42.90$ ,  $p < .01$ .

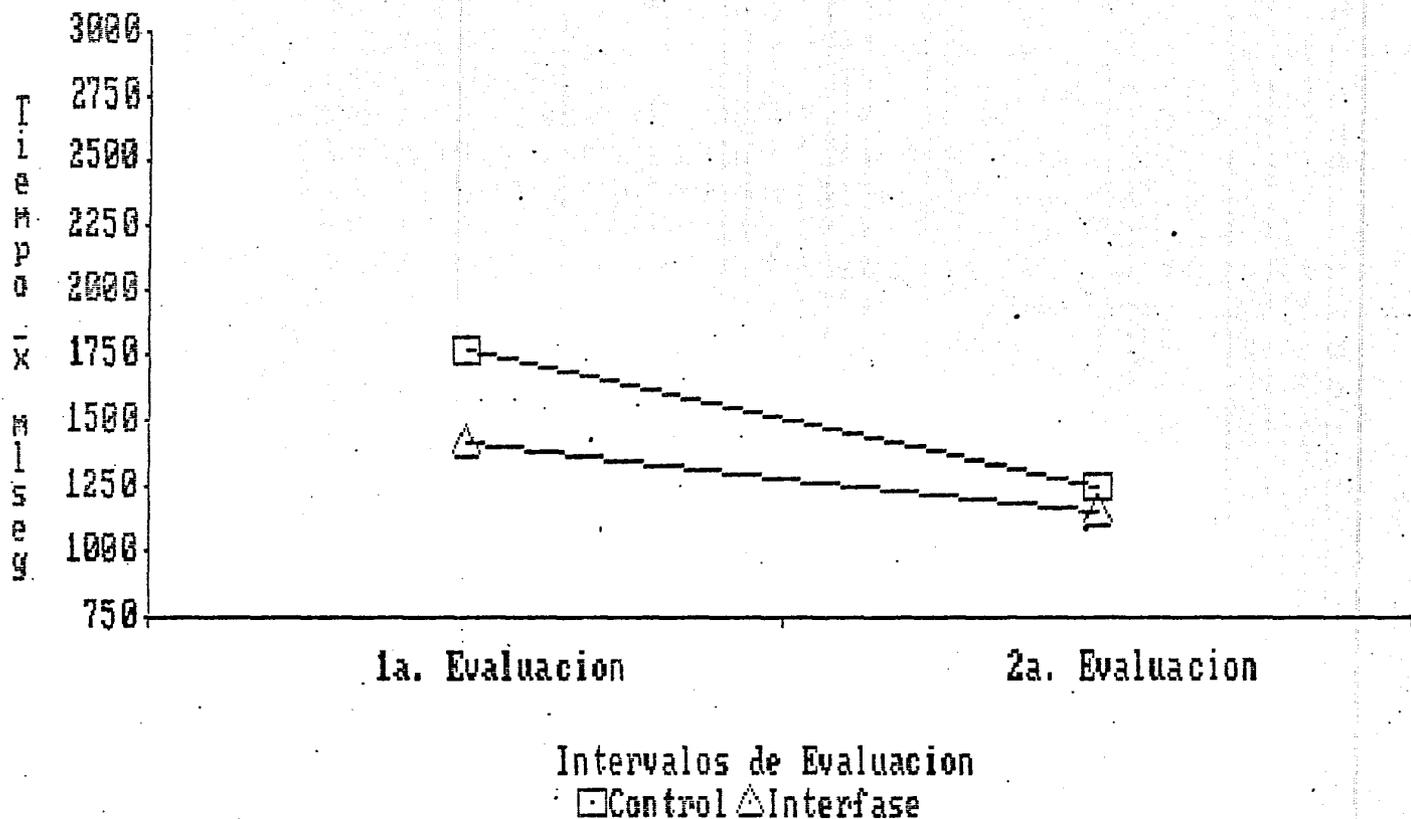


Figura 10. Tiempos promedio de verificación de interrogantes con errores de sintaxis por grupo en la primera y segunda evaluación.

## DISCUSION

### IV. DISCUSION.

En la tarea de construcción los sujetos del grupo experimental obtuvieron mayor número de aciertos que el grupo control y utilizaron menor tiempo para programar interrogantes en SQL, la diferencia en aciertos de ambos grupos se manifestó en la construcción sintáctica de selección de Grupos.

En la tarea de verificación, el grupo experimental obtuvo significativamente mayor número de rechazos correctos que el grupo control, sin embargo ambos grupos no se diferenciaron en el tiempo de verificación. En lo que respecta a los aciertos el grupo experimental y control no se diferenciaron en la cantidad de aciertos ni en el tiempo de verificación de los mismos.

#### 1.0 ¿ CUAL ES EL EFECTO DE LA INTERFASE EN EL APRENDIZAJE DEL SQL?

En virtud de que la interfase bajo estudio está orientada a la sintaxis del SQL, el presente trabajo se dirigió a manipular y evaluar exclusivamente el efecto de la interfase en la etapa de codificación de interrogantes.

En general, puede decirse que el efecto de la interfase fué positivo, dado que la interfase ayudo a los usuarios en los siguientes aspectos:

#### 1.1 PROGRAMAR LOS INTERROGANTES EFICIENTEMENTE.

La interfase proporcionó una doble ventaja operativa a los usuarios, por un lado, al disponer de las teclas predefinidas, se redujo la cantidad de escritura para generar los nombres de las cláusulas, tablas y atributos (véase fig. 5). ahora bien, al generar de esta forma los nombres, también se minimizaron los errores lexicográficos al transcribirlos (ver tabla 3).

#### 1.2 IDENTIFICAR LAS REGLAS SINTACTICAS VALIDAS PARA CONSTRUIR INTERROGANTES.

La interfase ofreció ayuda de tipo perceptual para utilizar adecuadamente las cláusulas del SQL en la construcción de interrogantes.

Para observar el efecto en detalle de la plantilla de colores, es importante analizar el desempeño de los sujetos en el interrogante de selección de grupos. Este interrogante se caracteriza por ser el más complejo, ya que presupone mas reglas sintácticas que los demás interrogantes (ver apéndice E). Dada esta complejidad, los sujetos cometieron errores en este interrogante. En este estudio se identificaron tres tipos de errores, un error de tipo lógico que no sera discutido por estar fuera de contexto y dos errores de tipo sintáctico que se describen a continuación. 1) omitir la cláusula "group by", y 2) declarar operaciones agregadas junto a la cláusula de "where" (véase tabla 5). El primer error puede evitarse si el sujeto recuerda que tiene que declarar la cláusula de "group by" antes de la de "having", si no es así se produce el error. El segundo error es más interesante aún, pues los sujetos que cometen este error utilizan la cláusula de "where" (condiciona registros) por

## DISCUSION

la de "having" (condiciona grupos), este error podría ser evitado si el sujeto recordara que las operaciones agregadas solo se pueden utilizar junto a las cláusulas de "select" ó "having", ambos errores pueden ser identificados con la plantilla de colores del SQL (ver apéndice D).

Otro error interesante es el que ocurre cuando el usuario utiliza "order by" por "group by". Ante este error hay dos explicaciones, 1) similitud fonética, los usuarios las confunden porque terminan en "by", 2) similitud operacional "order by" ordena registros; "group by" ordena registros y los agrupa. cualquier explicación que sea la correcta, el usuario habría distinguido este error si recordara que al utilizar operaciones agregadas no se utiliza la cláusula de "order by". Esta regla también está contemplada en la plantilla de colores (ver apéndice D).

### 1.3 DETECTAR ERRORES DE SINTAXIS EN INTERROGANTES YA FORMULADOS.

La interfase ayudo a los usuarios a rechazar correctamente interrogantes con errores de sintaxis.

### 1.4 LIMITACIONES DE LA INTERFASE.

Algunos detalles sintácticos estuvieron fuera del alcance de la interfase. Los errores tales como la omisión del atributo por el que se ordena e invertir el orden unique\_atributo (véase tabla 5) ilustran un punto muy importante; la plantilla de colores no establece en estos casos la relación posicional cláusulas y atributos, lo cual si ocurre por ejemplo con las cláusulas de operaciones agregadas (véase fig. 3), max(), min(), avg(), sum(). La función del parentesis que abre es indicar al usuario el lugar del atributo, una lógica similar debe aplicarse a las cláusulas que presentaron estos problemas. Por ejemplo, añadir el caracter de subrayado a la izquierda de la cláusulas order by\_ y unique\_.

## 2.0 DESEMPEÑO DE LOS SUJETOS UNA SEMANA POSTERIOR AL ENTRENAMIENTO

En general, hay dos resultados que sugieren que el desempeño de ambos grupos mejoró de la primera a la segunda evaluación: obtuvieron mayor número de aciertos en el interrogante de selección de grupos en la tarea de construcción, y obtuvieron más aciertos en la tarea de verificación. ¿A que se puede atribuir esta ganancia?, Dadas las características del diseño experimental de este estudio, la posibilidad más viable a considerar es el efecto de la práctica. Esta ganancia no se le puede atribuir directamente a la interfase, ya que el único beneficiado hubiera sido el grupo interfase. Este resultado es una evidencia más que se añade a la reportada por Reisner (1977) en cuanto a que el SQL si es facil de aprender aún para usuarios ajenos a la programación.

Ahora bien, ambos grupos de sujetos utilizaron menos tiempo al verificar interrogantes correctamente formulados e interrogantes con errores de sintaxis. Esto posiblemente ocurrió debido a que los sujetos estuvieron mas familiarizados con el

## DISCUSION

sistema en la segunda evaluación y esto influyó en sus latencias de respuesta. Este efecto puede reducirse tanto en ambientes reales como en laboratorio, dando largos periodos de sensibilización en tareas como esta, que pueden parecer aparentemente simples para los usuarios.

### 3.0 LIMITACIONES METODOLOGICAS DEL PRESENTE ESTUDIO.

#### 3.1 DISEÑO.

La interfase para el SQL se compone por una plantilla de colores con las cláusulas del SQL con su teclado correspondiente, plantilla de nombres de tablas y atributos con el teclado correspondiente. El diseño del presente estudio no diferenció el efecto individual de cada uno de los componentes, por ejemplo no se puede saber si el tiempo de los sujetos del grupo interfase en la tarea de construcción se debió al teclado, a la plantilla de cláusulas o a ambas. En un estudio posterior, el diseño debe extraer el efecto individual o interacciones de los componentes de la interfase.

#### 3.2 EVALUACION.

1. En el presente estudio se encontraron algunos efectos de techo en la cantidad de aciertos obtenidos por los sujetos en las tareas de construcción (véase fig. 4).

2. El registro del tiempo fué global, no se distinguió el tiempo que se invertía exclusivamente en la escritura de otras actividades, como retroceder el cursor para corregir errores, interrupciones de los sujetos etc.

3. El registro de los errores se limitó a aquellos que los usuarios no corregían, cabe agregar que hubo errores que cometían los sujetos que se corregían inmediatamente.

Para solucionar el primer punto es recomendable expandir el número de interrogantes para las tarea de construcción. Para mejorar la captura de tiempos y registros sería ideal contar con una cámara de video para llevar un registro detallado de estas dos variables.

### 4.0 PERSPECTIVAS PARA EL DESARROLLO FUTURO DE INTERFASES PARA EL USUARIO DE SQL.

Si bien el efecto de la interfase para el SQL tuvo un efecto positivo en la etapa de codificación de interrogantes, los resultados no son del todo óptimos ya que algunos errores que cometieron los sujetos del grupo experimental estuvieron lejos del alcance potencial de la interfase. Asimismo, los errores potenciales de las etapas de formulación del problema y preparación del plan tales como definir inadecuadamente el problema ó elegir una estrategia inadecuada son también ajenos a la interfase. Luego entonces, es conveniente considerar una alternativa integral para el diseño de interfases para el usuario de lenguajes de interrogación a bases de datos en general y para el SQL en particular.

## DISCUSION

En el presente apartado de la discusión se analizará una situación hipotética de interrogación. El análisis tiene un doble propósito: observar a qué tipo de problemas se enfrentan los usuarios y contrastar la forma en que tres diferentes interfaces para el usuario abordarían esos problemas.

**Interfase 1.** Interfase convencional de SQL, el usuario de ésta cuenta con la configuración mínima: la interfase de SQL (lenguaje) y documentación.

**Interfase 2.** Interfase compuesta por una plantilla de colores y teclado con las cláusulas del SQL, más una plantilla y teclado de nombre de tablas y atributos.

**Interfase 3.** Interfase diseñada en base a menús y activada mediante ratón "mouse". Esta interfase esta siendo desarrollada por Octavio Torres (1987, comunicación personal, véase fig. 11) y por lo tanto no se tiene evidencia empírica de su eficiencia. Sin embargo, esto no es un impedimento para inferir algunas de las propiedades de la interfase dadas las características de su diseño.

# DISCUSION

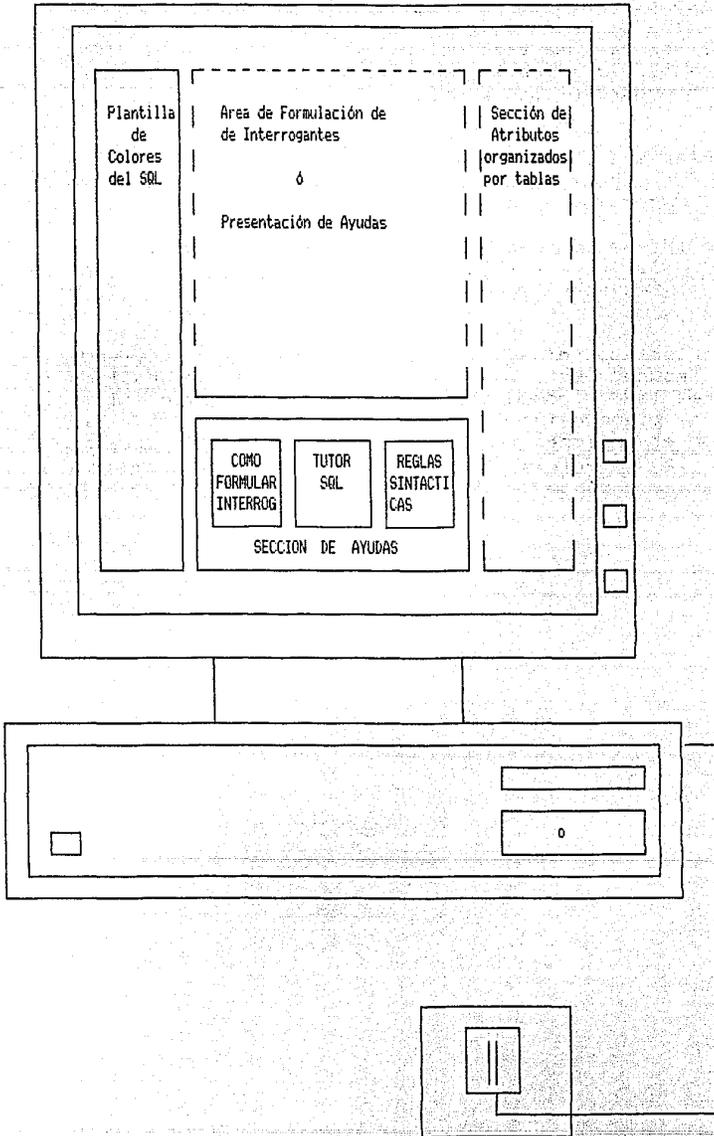


FIGURA 11. Interfase para el usuario del SQL en base a Menús conducido por ratón "mouse"

## DISCUSION

Utilizando el modelo de Gould y Ascher (1985) (véase fig. 2) como guía de análisis. Supóngase la siguiente situación de consulta a una base de datos:

El usuario desea obtener una relación de artículos vendidos por la persona con el RFC BAT0621204 durante el mes de septiembre.

### 4.1 DESCRIPCION GLOBAL DE LA PRODUCCION DEL INTERROGANTE.

A continuación se analizarán las actividades que efectuaría el usuario en cada etapa de la producción de interrogantes, así como los requerimientos y resultado de cada una de las etapas.

#### 4.1.1 ETAPA I. Formulación del Problema.

**Actividad:** El usuario define que es lo que quiere en su lenguaje nativo.

**Requerimiento 1:** Esquema de la vista a la base de datos a la que tiene acceso.

**Requerimiento 2:** Habilidad lingüística para enunciar claramente los datos que quiere consultar.

**Resultado:** ¿Cuáles son los nombres de los artículos que vendió el RFC= BAT0621204 durante el mes de septiembre ?

#### 4.1.2 ETAPA 2. Preparación del Plan.

**Actividad:** El usuario infiere las operaciones que se requieren para obtener los datos que busca.

**Requerimiento 1:** Un modelo mental de operación consistente con el funcionamiento del SQL.

Razonamiento del usuario.

1. "Los datos que debo manejar están en los atributos, rfc, nombre\_art y mes".

ARTICULOS			VENTAS		
clave	nombre_art	precio	rfc	clave	mes
1	LAPIZ	5	BAT0621204	1	JUN
2	BORRADOR	10	CAPA601201	4	JUN
3	HOJA	15	BAT0621204	2	SEP
4	PLUMA	30	ZAPA351203	3	SEP
			BAT0621204	1	SEP
			CAPA601201	2	OCT
			BAT0621204	3	OCT

2. "Los atributos están en las tablas de ventas y artículos".

3. "Dado que los atributos están en tablas diferentes, tengo que unir tablas ¿ Las puedo unir ? si se pueden unir, ya que comparten el atributo de clave".

## DISCUSION

clave	nombre_art	Precio	rfc	clave	mes
1	LAPIZ	5	BAT0621204	1	JUN
4	PLUMA	30	CAPA601201	4	JUN
2	BORRADOR	10	BAT0621204	2	SEP
3	HOJA	15	ZAPA351203	3	SEP
1	LAPIZ	5	BAT0621204	1	SEP
2	BORRADOR	10	CAPA601201	2	OCT
3	HOJA	15	BAT0621204	3	SEP

4. "De la tabla que se formó me interesan los registros que cumplen la condición de que el mes sea septiembre y el rfc= BAT0621204".

clave	nombre_art	Precio	rfc	clave	mes
1	LAPIZ	5	BAT0621204	1	JUN
4	PLUMA	30	CAPA601201	4	JUN
2	BORRADOR	10	BAT0621204	2	SEP
3	HOJA	15	ZAPA351203	3	SEP
1	LAPIZ	5	BAT0621204	1	SEP
2	BORRADOR	10	CAPA601201	2	OCT
3	HOJA	15	BAT0621204	3	SEP

v

clave	nombre_art	Precio	rfc	clave	mes
2	BORRADOR	10	BAT0621204	2	SEP
1	LAPIZ	5	BAT0621204	1	SEP
3	HOJA	15	BAT0621204	3	SEP

BORRADOR  
LAPIZ  
HOJA

[el usuario determina el tipo de interrogante que resuelve su problema]

**Resultado:** "Luego entonces, requiero hacer una proyección condicional".

1. "Tengo que pedir el nombre del artículo"
2. "Indicar la procedencia de los atributos involucrados en el interrogante".
3. "Unir las tablas".
4. "Establecer la condición para seleccionar registros".

## DISCUSION

### 4.1.3 ETAPA III. Codificación del Problema

Actividad: El usuario expresa las operaciones que infirió en terminos de la sintaxis del SQL.

Requerimiento 1: Conocimiento de las reglas sintácticas del SQL.

Requerimiento 2: Escribir el Interrogante.

Resultado:

```
select nombre_articulo
from ventas, articulo
where articulo.clave_articulo=ventas.clave_articulo
and (rfc= 'BAT0621204 and mes='SEP') /
```

## 4.2 COMPARACION DE TRES INTERFASES PARA EL USUARIO DEL SQL.

4.2.1.1 Etapa 1. formulación del problema (requerimiento 1. esquema de la vista de la base de datos)

Interfase 1. El usuario recurre a un esquema que ilustra su vista correspondiente a la base de datos, este esquema incluye los nombres de las tablas y atributos.

Interfase 2. Igual que en la interfase 1. Adicionalmente se requiere que los nombres de tablas y atributos esten declarados en un formato especial, de tal manera que esos nombres se puedan predefinir en la plantilla de nombres de tablas y atributos.

Interfase 3. Cada usuario tiene acceso interactivo a la vista de base de datos que le corresponda mediante la ayuda correspondiente. (véase fig. 12)

## DISCUSION

El usuario elige la ayuda de Como formular interrogantes, tiene la opción de pedir:

1. Su vista particular de la Base de Datos.
2. Reglas de redacción de preguntas.

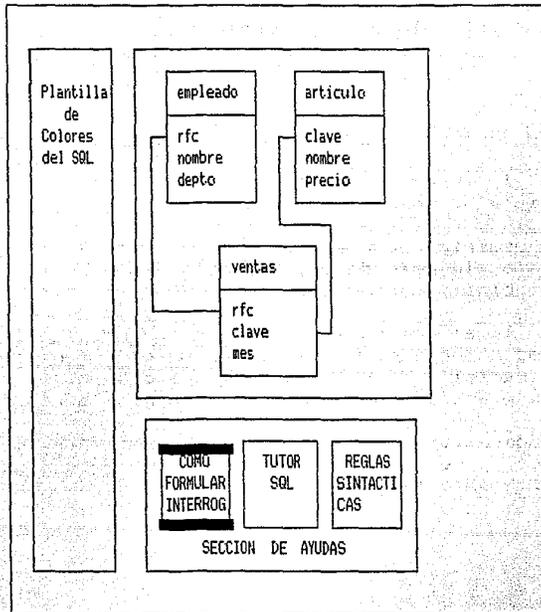


FIGURA 12. Ayuda para formular interrogantes de la interfase para el usuario del SQL a base de menús y conducida por ratón.

## DISCUSION

### 4.2.1.2 Etapa 1, requerimiento 2 (habilidad lingüística del usuario).

Dada la complejidad de este recurso, la ayuda que pueden ofrecer las tres interfases es limitada. Independientemente de la interfase en uso, los usuarios deben entrenarse para enunciar preguntas adecuadamente. En este contexto, se puede disponer de un documento de reglas de redacción en papel para las interfases 1 y 2, y en línea para la interfase 3.

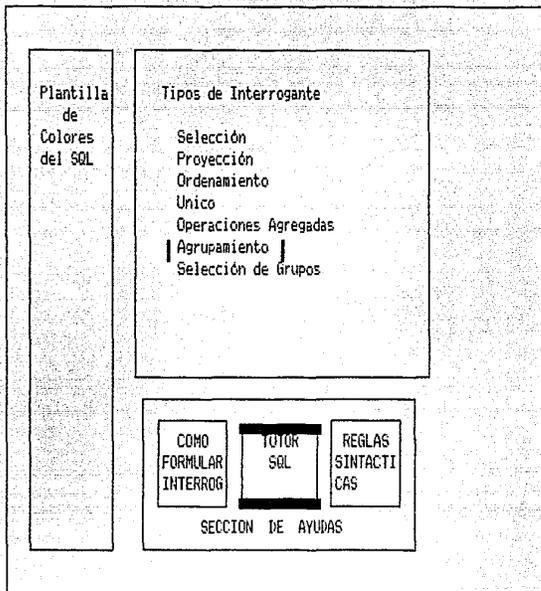
En esta etapa, la interfase 3 se diferencia de las interfases 1 y 2 en términos de el medio de presentación de las ayudas.

### 4.2.2 Etapa 2, preparación del plan.

Interfases 1 y 2. El usuario tiene a su disposición un manual que ilustra gráficamente las operaciones involucradas en cada tipo de interrogante. (ver apéndice E)

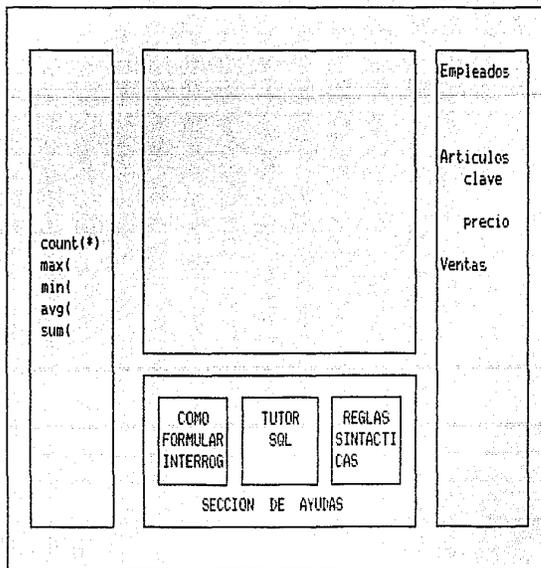
Interfase 3. El usuario puede acceder a un programa tutor que ilustra interactivamente las operaciones involucradas en el interrogante. (véase fig. 13)

## DISCUSSION 21

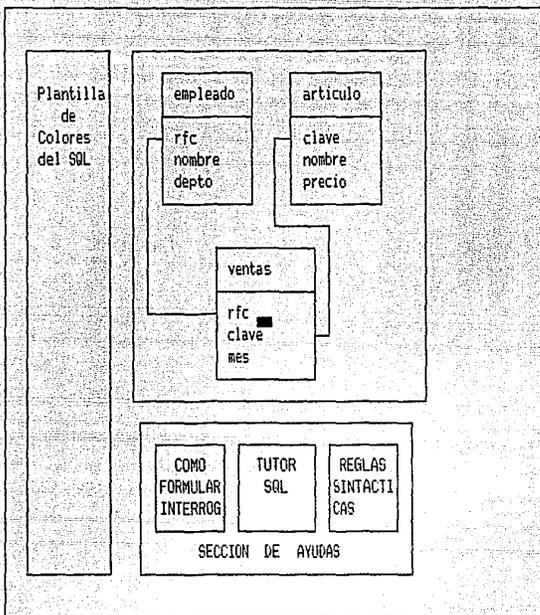


El usuario elige el interrogante de agrupamiento, La interfase pregunta por la operación agregada que se quiere aplicar:

- 1) En la plantilla de colores aparecen las operaciones agregadas.
  - 2) En la sección de atributos aparecen los atributos numericos.
- El usuario elige la operación de count(\*).



El tutor pregunta por el atributo por el cual se formarán los grupos. El usuario elige el atributo de rfc de la tabla ventas.



Area de presentación de ayudas

rfc	clave	mes
BAT0621204	1	JUN
CAPA601201	4	JUN
BAT0621204	2	SEP
ZAPA351203	3	SEP
BAT0621204	1	SEP
CAPA601201	2	OCT
BAT0621204	3	OCT

En primer termino se ordenan los registros por el contenido del rfc

rfc	clave	mes
BAT0621204	1	JUN
BAT0621204	2	SEP
BAT0621204	1	SEP
BAT0621204	3	OCT
CAPA601201	4	JUN
CAPA601201	2	OCT
ZAPA351203	3	SEP

Se agrupan los registros que tengan valores de rfc en común

BAT0621204	1	JUN
BAT0621204	2	SEP
BAT0621204	1	SEP
BAT0621204	3	OCT

CAPA601201	4	JUN
CAPA601201	2	OCT

ZAPA351203	3	SEP
------------	---	-----

Se aplica la operación agregada a los grupos formados

count(\*)

BAT0621204	1	JUN	4
BAT0621204	2	SEP	
BAT0621204	1	SEP	
BAT0621204	3	OCT	

CAPA601201	4	JUN	2
CAPA601201	2	OCT	

ZAPA351203	3	SEP	1
------------	---	-----	---

Obtenemos:

rfc	count(*)
BAT0621204	4
CAPA601201	2
ZAPA351203	1

Figura 13. Ayuda para la preparación del plan en la interfase para el usuario de SQL mediante menús.

## DISCUSION

### 4.2.3 Etapa 3; codificación del problema.

#### Interfase 1.

Requerimiento 1 (Reglas sintácticas). El usuario debe tener presente las reglas sintácticas del SQL, para este efecto cuenta con el manual del SQL.

Requerimiento 2 (escribir el interrogante). El usuario debe de programar el interrogante en SQL caracter por caracter, esto incluye cláusulas, nombres de tablas y atributos y signos de puntuación.

#### Interfase 2

Requerimiento 1 (Reglas sintácticas). El usuario utiliza la plantilla de colores como guía perceptual para tener presente las reglas sintácticas del SQL, adicionalmente cuenta con el manual del SQL.

Requerimiento 2 (escribir el interrogante). El usuario recurre al teclado de cláusulas y al teclado de nombres de tablas y atributos para escribir el interrogante de SQL. Los signos de puntuación, así como algunos nombres de tablas y atributos tienen que ser generados con el teclado normal.

#### Interfase 3

Requerimiento 1 (reglas sintácticas). Esta interfase, además de ofrecer ayuda perceptual de la sintaxis del SQL, (mediante la plantilla de colores) funciona como guía operativa para el usuario, ya que la interfase tiene almacenadas las reglas sintácticas del SQL, lo cual implica que en esta interfase no se permiten combinaciones sintácticas ajenas a las permitidas. Adicionalmente, la interfase proporciona ayudas de tipo sintáctico (por cláusula y por tipo de interrogante) mediante la ayuda correspondiente. (véase fig. 14)

Requerimiento 2 (escribir el interrogante). Para generar el interrogante, nombres de las cláusulas, nombres de tablas y atributos y signos de puntuación la tarea del usuario se reduce a deslizar el ratón (mouse) por la pantalla de la terminal. En la figura 15 se ilustra mas detalladamente esta facilidad.

El usuario elige la ayuda de Reglas sintacticas. Tiene la opción de pedir:

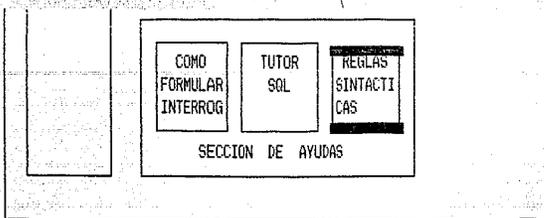
1. Ayuda sobre alguna cláusula en particular
2. Ayuda sobre algún tipo de interrppgante.

El usuario pide ayuda acerca de la cláusula max()

1. En la sección de ayudas se presenta el texto de la ayuda.
2. En la plantilla de colores se ubica contextualmente a la cláusula

select	<p>max(), es una operación agregada. Se utiliza para obtener el maximo valor de un atributo numerico.</p> <p>Se utiliza en conjunción de las cláusulas select</p> <pre>select max(atributo)</pre>						
count(*) max() min() avg() sum()							
<table border="1"><tr><td>COMO FORMULAR INTERROG</td><td>TUTOR SQL</td><td>REGLAS SINTACTI CAS</td></tr><tr><td colspan="3" style="text-align: center;">SECCION DE AYUDAS</td></tr></table>		COMO FORMULAR INTERROG	TUTOR SQL	REGLAS SINTACTI CAS	SECCION DE AYUDAS		
COMO FORMULAR INTERROG	TUTOR SQL	REGLAS SINTACTI CAS					
SECCION DE AYUDAS							

select from where between and or	<p>O junto a la cláusula de having para establecer el criterio de selección de grupos</p>
count(*) max() min() avg() sum() group by having	<pre>select max(atributo),atributoi from tabla group by atributo_i having count(*) &gt;2/</pre>



El usuario pide ayuda del interrogante de Único.

1. En la plantilla se presentan todas las cláusulas permitidas en este interrogante.

2. En la sección de ayudas se presentan las variantes del interrogante Único con su respectiva explicación.

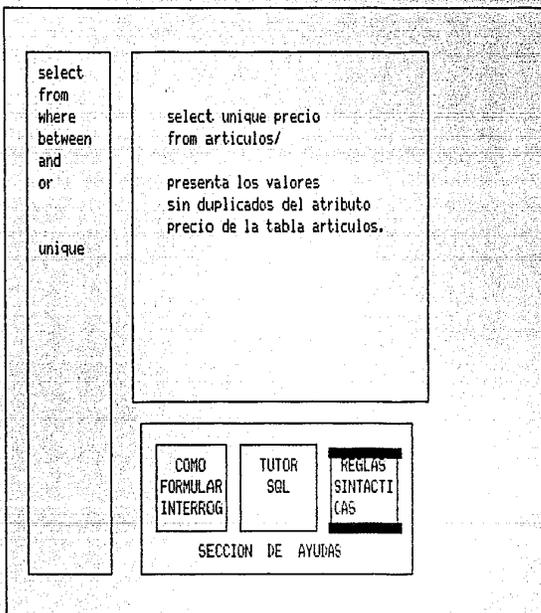
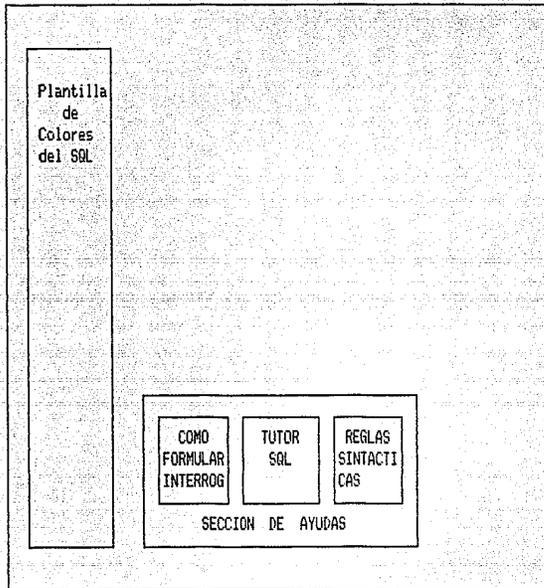


Figura 14. Ayuda para la Codificación del Interrogante

Suponga que el usuario desea saber:

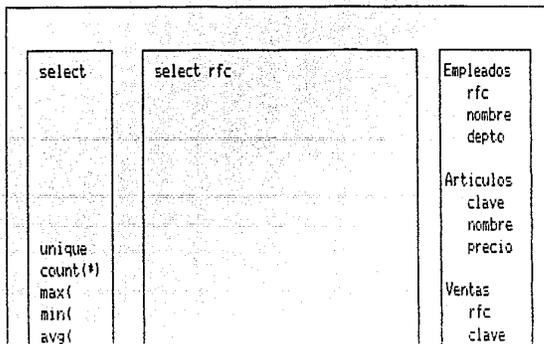
La cantidad total que ha vendido cada RFC,  
la meta es generar el siguiente interrogante de SQL.

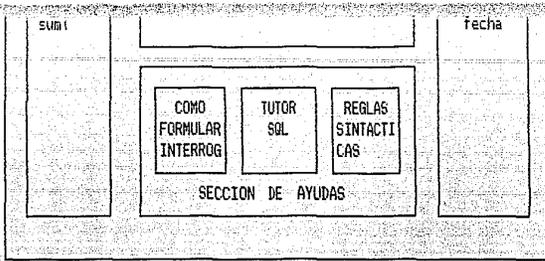
```
select rfc, sum(precio)
from ventas, articulos
where ventas.clave_articulo= articulos.clave_articulo
group by rfc /
```



Al iniciar el modo de interrogación, aparece en la plantilla de colores la cláusula de select, en la parte media de la plantilla aparecen también las cláusulas con las que se puede combinar y en la sección de atributos los atributos que se pueden escoger.

[el usuario elige el atributo de rfc]

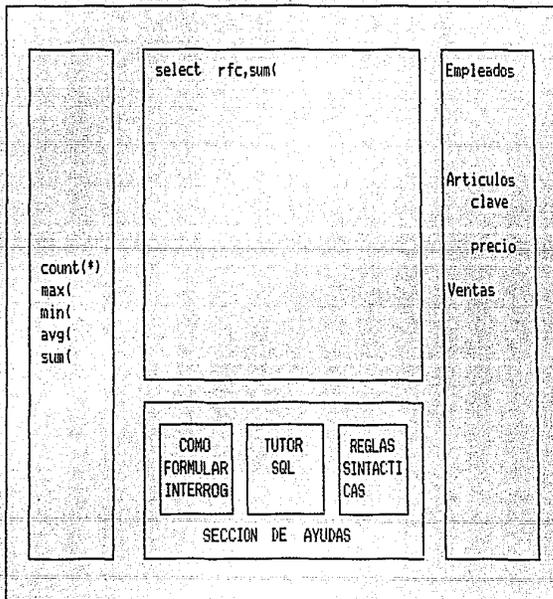




[el usuario elige la operación agregada de suma]

observe que:

1. Aparece una coma enseguida de rfc, (se agregan comas mientras se declaren atributos y/o operaciones agregadas.
2. En la plantilla desaparece la cláusula de unique, la razón consiste en que unique y las operaciones agregadas se pueden combinar con la cláusula de select, pero son excluyentes. (ver apéndice E)
3. En la sección de atributos aparecen únicamente los atributos que se definieron como numericos.



1. Al elegir el atributo al que se le aplicará la operación agregada, aparece automáticamente el parentesis que cierra.

2. Aparecen de nueva cuenta todos los atributos, para el caso de

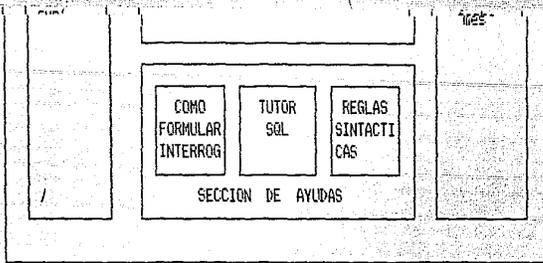
que el usuario, desea escoger el contenido de otro(s) atributos. En el caso de que eligiera otra operación agregada, aparecerían únicamente atributos numéricos.

<p>count(*) max( min( avg( sum(  </p>	<pre>select rfc,sum(precio)</pre>	<p>Empleados rfc nombre depto</p> <p>Articulos clave nombre precio</p> <p>Ventas rfc clave mes</p>						
<table border="1"> <tr> <td data-bbox="417 560 484 642">COMO FORMULAR INTERROG</td> <td data-bbox="496 560 563 642">TUTOR SQL</td> <td data-bbox="574 560 641 642">REGLAS SINTACTI CAS</td> </tr> <tr> <td colspan="3" data-bbox="451 657 596 672">SECCION DE AYUDAS</td> </tr> </table>			COMO FORMULAR INTERROG	TUTOR SQL	REGLAS SINTACTI CAS	SECCION DE AYUDAS		
COMO FORMULAR INTERROG	TUTOR SQL	REGLAS SINTACTI CAS						
SECCION DE AYUDAS								

Al darse por terminado el renglón de select:

1. Se genera automáticamente el renglón de from, dado que la interfase tiene identificado la procedencia de los atributos elegidos.
2. En virtud de que los atributos son de tablas distintas, la interfase también establece automáticamente la unión de tablas.
3. En la plantilla aparecen las cláusulas que se utilizan con la cláusula de where, por si el usuario desea establecer alguna condición de selección de registros.
4. En el fondo de la plantilla de colores aparece una diagonal (signo para dar por terminado el interrogante)

<p>between and or</p>	<pre>select rfc,sum(precio) from ventas, articulos where ventas.clave=articulos.clave</pre>	<p>Empleados rfc nombre depto</p> <p>Articulos clave nombre precio</p> <p>Ventas rfc clave</p>
-------------------------------	---------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------



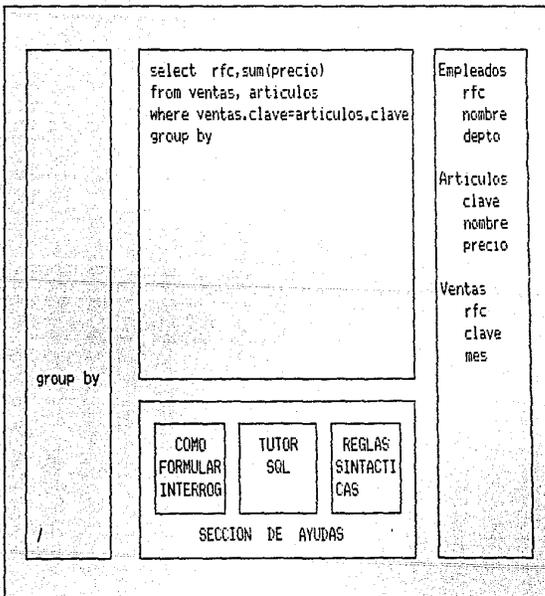
El usuario pasa al siguiente renglón.

1. En la plantilla aparece la cláusula de group by, dado que es la única que puede ser declarada inmediatamente después de where, tomando en cuenta que se declaró anteriormente una operación agregada. (ver apéndice E)

2. Aparecen los atributos por los que se puede agrupar.

El usuario elige agrupar por el atributo de rfc.

1) En el caso de que el usuario hubiese elegido un atributo no contemplado en las tablas previamente definidas, la interfase reajustaría los renglones de from y where.



El usuario pasa al siguiente renglón.

1) El usuario tiene la opción de utilizar ahora la cláusula de having ó dar por terminado el interrogante.

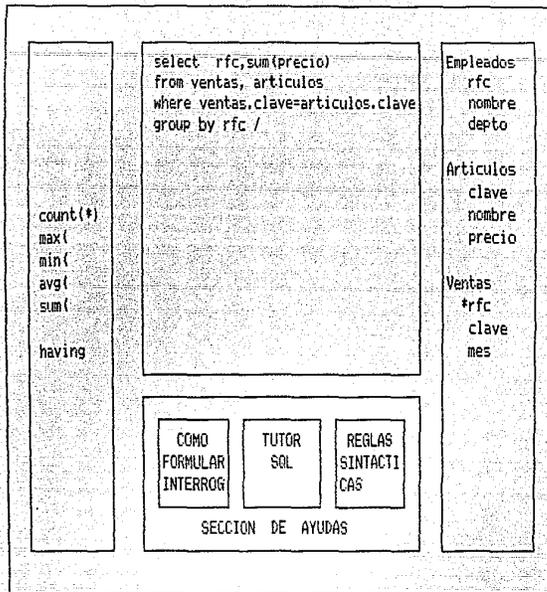


Figura 15. Construcción del Interrogante mediante la Facilidad por Menús conducido por ratón "mouse".

## DISCUSSION

## DISCUSION

En la interfase 1 el usuario está sujeto a cometer errores de tipo sintáctico y lexicográfico con relativa facilidad. La interfase 2 proporciona ayuda mediante la plantilla de colores para prevenir errores de tipo sintáctico, más no los puede evitar (véase tabla 5), por otra parte los errores de tipo lexicográfico si son evitados, lo cual no ocurre con los errores de puntuación. (véase tabla 3). Por otro lado, el usuario de la interfase 3 prácticamente no tendría de que preocuparse por cometer errores ya sean sintácticos, lexicográficos o de puntuación, en virtud de que la interfase guiaría al usuario en la codificación del interrogante.

## DISCUSION

### 5.0 CONCLUSIONES.

5.1 Las Interfases para el Usuario son necesarias para una interacción óptima sistema-usuario. El diseñador de interfases para el usuario debe considerar las capacidades y limitaciones del humano en el diseño de su interfase. Para este objeto, el diseñador debe apoyarse en un modelo que describa, explique y prediga las actividades que realizaria el usuario en la tarea que se pretenda facilitar. En este estudio se utilizó el modelo de producción de interrogantes de Gould y Ascher (1975), en el cual se ubica la interfase para el usuario de SQL presentada en este trabajo.

5.2 La interfase estándar del SQL (el lenguaje) es fácil de aprender. Sin embargo, debe proporcionarse al usuario de una interfase externa al lenguaje que incremente aún más su facilidad de uso. En el presente trabajo se presenta una interfase para el usuario del SQL e introduce una nueva alternativa.

1. Interfase para el usuario del SQL consistente en un teclado con las cláusulas del SQL y un segundo teclado con los nombres de tablas y atributos. Esta interfase esta diseñada para reducir los errores de tipo sintáctico y lexicografico en la etapa de codificación de interrogantes. En el presente estudio se encontró que la interfase ayuda a reducir algunos errores de tipo sintáctico y lexicográficos. Sin embargo, algunos errores estuvieron fuera del alcance de la interfase, por ejemplo errores de puntuación (intercambiar signos), errores de tipo lógico (omisión de una cláusula determinante en el interrogante), errores de tipo sintáctico (omisión del atributo por el que se quiere ordenar, invertir el orden unique\_atributo).

2. Interfase para el usuario del SQL en base a menús y conducido por ratón "mouse" (Torres, 1987 comunicación personal). Esta interfase esta en desarrollo y esta diseñada para abordar el problema de interrogación a bases de datos de una forma integral. Esto implica que la interfase esta orientada a reducir los errores de tipo sintáctico, lexicográfico y de puntuación del SQL (etapa de codificación), así como los errores potenciales de las etapas previas (etapas de formulación del problema y preparación del plan).

## REFERENCIAS BIBLIOGRAFICAS

- Brown, P. J. (1983). "Error Messages: The Neglected Area of the Man/Machine Interface?", Communications of the ACM, Vol. 26, (4), 246-249.
- Cohill, A. M. y Williges, R. C. (1985). "Retrieval of HELP Information for Novice Users of Interactive Computer Systems". Human Factors, Vol. 27, (3), 335-343.
- Ehrenreich, S. L. (1981). "Query Languages: Design Recommendations derived from the Human Factors Literature". Human Factors, Vol. 23, (6), 709-725.
- Entrevista a H. Michael Braude: "Analizando las Estrategias Futuras de Software". Computer World/Mexico, Año 6, No. 151, Marzo 31, 1986, 28.
- Gould, J. H. y Grischkowsky, N. (1984). "Doing the Same Work with Hard Copy and with Cathode-Ray Tube (CRT) Computer Terminales". Human Factors, Vol. 26, (3), 323-337.
- Gould, J. y Ascher, R. (1975). "Use of an IQF-like query language by nonprogrammers: Yorktown Heights, N. Y.: IBM Thomas J. Watson Research Center, Report RC 5279, February, 1965.
- Hill, I. D. (1972). "Wouldn't it be nice if we could write programs in ordinary English or would it?". The Computer Bulletin, Vol. 16, 306-312.
- Hirsch, R. S. (1981). "Procedures of the Human Factors at San Jose". IBM Systems Journal, Vol. 20, No. 2, 123-171.
- Just, M. A., Carpenter, P. A. y Woolley, J. D. (1982). "Paradigms and Processes in Reading Comprehension", Journal of Experimental Psychology: General, Vol. 111, (2), 228-236.
- Norman, D. (1987). "Design Rules Based on Analyses of Human Error". Communications of the ACM, Vol. 26, (4), 254-258.
- Reisner, P. (1977). "Use of Psychological Experimentation as an Aid to development of a Query Language". IEEE Transactions on Software Engineering, Vol. SE-3, (3), 218-229.
- Schlager, M., S. y Ogden, W. C. (1986). "A cognitive Model of Database Querying: A Tool for Novice Instruction", CHI'86 Proceedings, 107-113.

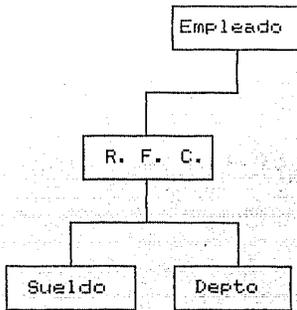
## REFERENCIAS BIBLIOGRAFICAS

- Small, D. W., Weldon, L. J. (1983). "An Experimental Comparison of Natural and Structured Query Language". Human Factors, Vol. 25, (3), 253-263.
- Torres, O. F. (1984a). "Factores Humanos en el Diseño de Interrogadores a Bases de Datos". Revista de Computación 010, Vol. 4, (8), 35-42.
- Torres, O. F. (1984b). "El Usuario de Editores de Texto: Un Analisis de sus Errores". Revista de Computación 010, Vol. 4, (6), 14-21.
- Torres, O. F., Arao, J. (1986). "La Interfaz con el Usuario de una Base de Datos Académicas: Reporte 2". Comunicaciones Técnicas de la Facultad de Psicología, No. 25.
- Torres, O. F., Arao, J. (1987). "Two Cases of Improvement of the User Interface to Data Bases". Manuscrito enviado a publicación.
- Torres, O. F., Bibriesca, R. (1986). "Sobre un Procedimiento de Evaluación Ergonómica de Manuales para el Usuario", Segunda Conferencia Internacional Las Computadoras en Instituciones de Educación Superior, pp. 312-314.
- Sandberg, G. (1981). "A Primer on Relational Data Base Concepts", IBM SYST J., Vol. 20, (1), 23-40.
- Sternberg, S. (1969). "The Discovery of Processing Stages: Extensions of Donders' Method. Acta Psychologica 30 Attention and Performance II (W. G. Koster Ed.).
- Williges, R. C. y Williges, B. H. (1983). "Human-Computer Dialogue Considerations", Automatica, Vol. 19, (6), 767-773.

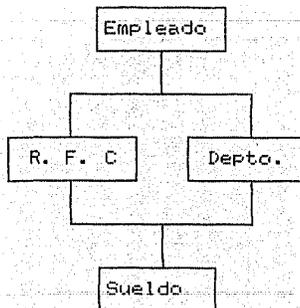
1.0 MODELOS DE SISTEMAS DE BASES DE DATOS.

Hay tres modelos de Bases de Datos: Jerárquico, de Red y Relacional.

1.1 Modelo Jerárquico, en este modelo los datos se almacenan en una estructura arborea, en la que cada nodo padre tiene un conjunto de nodos hijos, estos nodos hijos pueden ser padres de otros nodos hijos, la característica importante en estas relaciones es que cada nodo hijo tiene un solo nodo padre.



1.2 Modelo de Red, en el modelo de red, existen existen los nodos padres y nodos hijos, con la propiedad de que un nodo hijo puede tener más de un padre.



## APENDICE A

1.3 Modelo Relacional, en este modelo el usuario conceptualiza los datos en forma de tablas bidimensionales, en donde los renglones representan registros u ocurrencias de alguna entidad (por ejemplo empleado) y, columnas que representan atributos de alguna entidad (R. F. C., nombre, sueldo).

rfc	depto	sueldo
UISF501203	Almacén	500,000.00
TCC0550703	Véntas	600,000.00
AATJ621204	Almacén	300,000.00
CAHG601215	Compras	600,000.00
HUMM580812	Véntas	300,000.00
LUZB621201	Véntas	400,000.00

### 2.0 OPERACIONES RELACIONALES.

2.1 Selección, se utiliza para escoger registros de una tabla (secciona horizontalmente la tabla), los registros seleccionados dan lugar a una nueva tabla; es decir de una tabla dada se genera un subconjunto de la misma, el criterio para escoger registros de una tabla puede ser que uno o más atributos tengan ciertos valores.

Por ejemplo, deseamos obtener los registros de los empleados que esten en el almacén.

	rfc	depto	sueldo
> *	UISF501203	Almacén	500,000.00
	TCC0550703	Véntas	600,000.00
> *	AATJ621204	Almacén	300,000.00
	CAHG601215	Compras	600,000.00
	HUMM580812	Véntas	300,000.00
	LUZB621201	Véntas	400,000.00

rfc	depto	sueldo
UISF501203	Almacén	500,000.00
AATJ621204	Almacén	300,000.00

## APENDICE A

2.2 Proyección, esta operación se utiliza para escoger atributos de una tabla (sección verticalmente la tabla), los atributos escogidos con sus respectivos datos pasan a formar una nueva tabla.

Por ejemplo interesa obtener los rfc y deptos de los empleados

↓	↓	
v	v	
rfc	depto	sueldo
UISF501203	Almacén	500,000.00
TOC0550703	Ventas	600,000.00
AATJ621204	Almacén	300,000.00
CAHG601215	Compras	600,000.00
HUMM580812	Ventas	300,000.00
LUZB621201	Ventas	400,000.00

rfc	depto
UISF501203	Almacén
TOC0550703	Ventas
AATJ621204	Almacén
CAHG601215	Compras
HUMM580812	Ventas
LUZB621201	Ventas

Comunmente, selección y proyección trabajan conjuntamente, por ejemplo se desea saber el rfc y depto. de los empleados que tengan un sueldo mayor a 300,000.

Con la operación de selección se escogen los registros que tengan en el atributo de sueldo valores mayores a 300,000.

	rfc	depto	sueldo
> *	UISF501203	Almacén	500,000.00
> *	TOC0550703	Ventas	600,000.00
> *	AATJ621204	Almacén	300,000.00
> *	CAHG601215	Compras	600,000.00
> *	HUMM580812	Ventas	300,000.00
> *	LUZB621201	Ventas	400,000.00

APENDICE A

Se forma una tabla con registros que cumplen con la condición.

rfc	depto	suelo
UISF501203	Almacén	500,000.00
TOCQ550703	Véntas	600,000.00
CAHG601215	Compras	600,000.00
LUZB621201	Véntas	400,000.00

De esta tabla se proyectan los atributos de rfc y depto.

v                      v

rfc	depto	suelo
UISF501203	Almacén	500,000.00
TOCQ550703	Véntas	600,000.00
CAHG601215	Compras	600,000.00
LUZB621201	Véntas	400,000.00

rfc	depto
UISF501203	Almacén
TOCQ550703	Véntas
CAHG601215	Compras
LUZB621201	Véntas

2.3 Unión. mediante esta operación es posible formar una tabla nueva a partir de dos o más tablas que compartan algún atributo en común.

rfc	depto	suelo	rfc	nombre
UISF501203	Almacén	500,000.00	UISF501203	Uribe
TOCQ550703	Véntas	600,000.00	TOCQ550703	Torres
AATJ621204	Almacén	300,000.00	AATJ621204	Arango
CAHG601215	Compras	600,000.00	CAHG601215	Carpizo
HUMMS80812	Véntas	300,000.00	HUMMS80812	Humble
LUZB621201	Véntas	400,000.00	LUZB621201	Lucas

## APENDICE A

rfc	depto	sueldo	nombre
UISF501203	Almacén	500,000.00	Uribe
TOCO550703	Ventas	600,000.00	Torres
AATJ621204	Almacén	300,000.00	Arango
CAHG601215	Compras	600,000.00	Carpizo
HUMM580812	Ventas	300,000.00	Humble
LUZB621201	Ventas	400,000.00	Lucas

A esta tabla resultante se le pueden aplicar las operaciones de selección y proyección.

### 3.0 LENGUAJES RELACIONALES.

3.1 Lenguajes Algebraicos Relacionales. estos lenguajes proporcionan explícitamente las operaciones de Selección, Proyección y Unión, estos lenguajes trabajan con conjuntos de registros es decir con tablas; el SQL (Structured Query Language - Lenguaje de Interrogación Estructurado) es un ejemplo de lenguaje Algebraico, lo cual implica que es un lenguaje diseñado expresamente para interrogar bases de dato de tipo relacional. (véase las notas del SQL, Apéndice D).

3.2 Lenguajes Orientados Visualmente. en este tipo de lenguajes el usuario especifica las operaciones relacionales manipulando símbolos gráficos en la pantalla de la terminal. El Query-By-Example (QBE Interrogación por Ejemplo) es un ejemplo de estos lenguajes.

### 4.0 COMPARACION DE MODELOS DE BASES DE DATOS.

4.1 En el modelo relacional, el usuario no necesita conocer la estructura interna de la base, en contraste el usuario del modelo jerárquico o de red si debe de conocer la estructura interna de las bases; es decir mientras que el primer usuario ve tablas de datos, los otros usuarios ven estructuras complejas de datos en las que se requiere establecer trayectorias de acceso (navegación).

4.1.1 El usuario del modelo relacional manipula más fácilmente los datos de la Base, ya que para el usuario no existen trayectorias las de acceso, más bien, el usuario puede definir infinidad de trayectorias de acceso potenciales al unir tablas; por otro lado, el usuario del modelo jerárquico o de red debe de moverse por trayectorias de acceso predefinidas, es decir el usuario debe de navegar a lo largo o a lo ancho de la estructura jerárquica o de red, esto implica en terminos operativos mayor programación y menor facilidad de uso. Basicamente es por esta razón por la cual los modelos relacionales de Bases de Datos Relacionales son los más populares actualmente en el mercado.

## APENDICE B

### CLAUSULAS DEL SQL

#### CLAUSULA

#### FUNCION

#### select

declara la lista de atributos de los que se desea seleccionar datos.

#### from

indica el lugar (tabla o tablas) de procedencia de los atributos.

(\* estas clausulas conforman la estructura \*)

(\* sintáctica básica de un programa en SQL \*)

---

#### where

declara la condición para que se seleccionen o se rechazen registros.

la condición se basa en el resultado de una expresión booleana.

(\* funcionalmente, la clausula where es muy utilizada en los programas de SQL, por lo cual se asumira como básica a la estructura

```
select
from
where *)
```

#### and, or

operadores lógicos, conectan expresiones booleanas simples para formar expresiones compuestas.

El resultado de la expresión compuesta establece la condición de selección de registros.

---

#### expresiones simples

```
depto='Ventas*'
salario > 1000
```

#### expresiones compuestas

```
depto= 'Ventas*' and salario> 1000
depto= 'Ventas*' or salario> 1000
```

los operadores and y or se utilizan exclusivamente con la cláusula where.

## APENDICE B

### CLAUSULA

### FUNCION

#### between

especifica un rango de valores numérico, dentro del cual deben de caer el valor del campo numérico (de los registros) para que sean seleccionados.

la cláusula between se utiliza junto a la cláusula where, y requiere del operador lógico and para especificar el rango.

---

#### order by

ordena los registros en base al contenido de uno o mas atributos en orden ascendente o descendente.

por omisión order by ordena en forma ascendente.

para especificar el ordenamiento descendente se añade desc a la clausula order by.

order by atributo desc

para ordenar por mas de un atributo

order by atributo\_1, atributo\_2

---

#### unique

elimina los datos duplicados, selecciona solo los datos unicos de algún atributo

la clausula unique se declara junto a select

---

APENDICE B

CLAUSULA

FUNCION

\*\*\*\*\*

Operaciones Agregadas

\*\*\*\*\*

Las operaciones agregadas se declaran  
junto a la clausula de select o junto  
a la clausula de having.

-----  
count(\*)

cuenta registros

sum

suma los datos de algun atributo  
numérico.

avg

obtiene el promedio de los valores  
de algún atributo numérico.

max

obtiene el máximo de los valores  
de algún atributo numérico.

min

obtiene el mínimo de los valores  
de algún atributo numérico.

-----  
\*\*\*\*\*

group by

agrupa registros que tengan alguna  
caracterisitica en común (en base al  
contenido de uno o mas atributos),  
el objeto de aplicar alguna  
operación agregada a los mismos.

-----  
having

selecciona o rechaza grupos de  
registros formados previamente con  
group by.

la selección de grupos se basa en el  
resultado de alguna operación  
agregada.

## APENDICE C

### TIPOS DE INTERROGANTE DE SQL

TIPO DE INTERROGANTE

FUNCION

CLAUSULAS IMPLICADAS

---

SELECCION

Presenta todos los registros de una tabla.

select

from

---

PROYECCION

Presenta todo el contenido de un atributo o de varios atributos.

select

from

---

SELECCION CONDICIONAL

Presenta solo aquellos registros que que cumplan con alguna condición.

select

from

where y sus opciones

and

or

between

APENDICE C

TIPO DE INTERROGANTE  
CLAUSULAS IMPLICADAS

FUNCION

PROYECCION CONDICIONAL

Presenta los datos de una lista de atributos siempre y cuando se cumpla con alguna condición.

select

from

where y sus opciones  
and  
or  
between

UNION

Une dos o mas tablas con el objeto de extraer el segmento de información que deseemos.

select

from

where y la opción  
and

opcionalmente  
.....  
. or .  
. between .  
.....

ORDENAMIENTO

Ordena en forma ascendente o descendente los registros en base al contenido de uno o mas atributos.

select

from

opcionalmente  
.....  
. where y sus opciones .  
. and .  
. or .  
. between .  
.....

APENDICE C

order by y su opción

desc  
TIPO DE INTERROGANTE                      FUNCION

CLAUSULAS IMPLICADAS

---

UNICO                                              Presenta los datos unicos de algún atributo.

select            unique

from

opcionalmente

.....  
· where y sus opciones  
·                                              and  
·                                              or  
·                                              between  
.....

---

OPERACIONES AGREGADAS                      count(\*), cuenta registros  
                                                    las operaciones restantes efectuan  
                                                    calculos con los atributo de tipo  
                                                    numérico

select count(\*), avg(atributo), sum(atributo),  
          max(atributo), min(atributo)

from tabla

opcionalmente

.....  
· where y sus opciones  
·                                              and  
·                                              or  
·                                              between  
.....

TIPO DE INTERROGANTE      FUNCION

CLAUSULAS IMPLICADAS

**AGRUPAMIENTO**

Agrupar registros en base al contenido de algún atributo con el objeto de aplicar a los grupos alguna operación agregada.

select op agregada 1,.. op agregada n, atributo\_i  
from tabla

opcionalmente  
 .....  
 . where y sus opciones .  
 .                                    and                                    .  
 .                                    or                                    .  
 .                                    between                                    .  
 .....

group by atributo\_i

**SELECCION DE GRUPOS**

Selecciona grupos de registros formados previamente por el interrogante de agrupamiento.

El criterio de selección o rechazo de grupos se fundamenta en el resultado de alguna operación agregada.

select op agregada 1,.. op agregada n, atributo\_i  
from tabla

opcionalmente  
 .....  
 . where y sus opciones .  
 .                                    and                                    .  
 .                                    or                                    .  
 .                                    between                                    .  
 .....

group by atributo\_i  
having operacion agregada

## APENDICE D

### INSTRUCTIVO DE COLORES

El presente instructivo, forma parte de un programa de entrenamiento en SQL para usuarios sin experiencia en programación.

El programa de entrenamiento consta de dos fases :

#### 1. Ambientación de los usuarios con la interfase de interrogación.

En esta fase, los usuarios aprenden la sintáxis básica del SQL mediante la manipulación de colores.

#### 2. Entrenamiento en SQL.

En esta fase, los usuarios aprenden SQL manipulando directamente las palabras-cláusula y los colores.

Este instructivo se utiliza en la primera fase, y tiene por objetivo el facilitar el aprendizaje de las reglas sintácticas básicas de SQL de una forma rápida e intuitiva.

## APENDICE D

### INSTRUCTIVO

EN ESTE INSTRUCTIVO USTED APRENDERA A FORMAR COMBINACIONES DE COLORES UTILIZANDO LA PLANTILLA DE COLORES QUE TIENE A SU DISPOSICION.

#### 1.0 LA PLANTILLA ESTA COMPUESTA POR:

- 1.1 Tres colores primarios (ROJO, AMARILLO y AZUL).
- 1.2 Tres colores opcionales:

- CAFE
- ROJO CLARO/VERDE.
- VERDE/ROJO CLARO.

#### 2.0 COLORES PRIMARIOS.

A los colores primarios les corresponden ciertas VARIANTES.

##### 2.1 Al color ROJO le corresponden DOS variantes:

ROJO MEDIO (un cuadrado) y ROJO CLARO (5 cuadrados).  
Estas variantes solo pueden ser combinados con el color rojo.

##### 2.2 El color AMARILLO NO tiene variante.

##### 2.3 Al color AZUL le corresponde una variante:

AZUL MEDIO (3 cuadrados).  
Esta variante solo puede ser combinado con el color azul.

#### 3.0 COMBINACIONES MINIMAS DE COLORES

##### 3.1 La primer combinaci3n minima es

rojo
amarillo

##### 3.2 La segunda combinaci3n minima es

rojo
amarillo
azul

Observe que se utilizan renglones separados para cada color primario, la posici3n de estos colores es INALTERABLE

## APENDICE D

4.0 A partir de estas combinaciones se pueden generar otras combinaciones válidas.

La forma de proceder es:

4.1 A la DERECHA del color primario se agregan uno o varios cuadritos de la variante que le corresponda.

4.2 Recuerde que cada color primario ocupa un solo renglón y que el orden es INALTERABLE.

Ejemplos de combinaciones válidas:

rojo	rojo
amarillo	medio

o

rojo	rojo	
amarillo	claro	
azul	azul	azul
	medio	medio

Ejemplos de combinaciones INVALIDAS

rojo	azul
amarillo	medio

esta es una combinación INVALIDA porque el AZUL MEDIO solo puede añadirse al color AZUL

Otro ejemplo de combinación INVALIDA es:

rojo	rojo
claro	amarillo

esta combinación es INVALIDA porque el cuadrito (o cuadritos) de una variante se colocan a la DERECHA del color primario

APENDICE D

Otra combinación INVALIDA es:

azul	azul
rojo	medio
amarillo	

- ->

esta combinación también es INVALIDA porque se altera el orden de los colores primarios.

4.3 Al color ROJO primario SOLO se le puede(n) añadir cuadrito(s) de solo UNA de sus variantes a la vez, es decir:

ó se utiliza

rojo	rojo
amarillo	claro

O

rojo	rojo
amarillo	medio

PERO NO ES VALIDO

rojo	rojo	rojo*
amarillo	claro	medio
azul		

Esta es una combinación inválida debido a que aparecen cuadrillos de ambas VARIANTES del color ROJO a la vez (del rojo CLARO y del rojo MEDIO).

## APENDICE D

### COLORES OPCIONALES.

Los colores OPCIONALES (CAFE, ROJO CLARO/VERDE y VERDE/ROJO CLARO) se pueden añadir a las combinaciones mencionadas anteriormente.

Al igual que los colores primarios, los colores opcionales utilizan RENGLONES POR SEPARADO.

#### 5.0 Color CAFE

5.1 Al color CAFE le corresponde una variante:

CAFE MEDIO (un cuadrito).

Esta variante solo puede utilizarse con el color café.

El color café y su variante se pueden añadir a las combinaciones de colores anteriores siempre y cuando:

5.2 El color café se añada inmediatamente después del último color primario (ya sea el AMARILLO o el AZUL).

5.3 Se utilicen los colores de la plantilla que estén a su IZQUIERDA, pero NO los colores que se encuentren a su DERECHA (a partir del cuadrito rojo medio). (ver plantilla)

Ejemplos de combinaciones válidas.

rojo	
amarillo	
cafe	cafe medio

o

rojo		
amarillo		
azul	azul medio	azul medio
cafe	cafe medio	

Ejemplos de combinaciones INVALIDAS.

rojo	rojo medio
amarillo	*
cafe	cafe medio

Esta combinación es inválida porque, cuando se utiliza el color café no se pueden agregar colores que se encuentren a su DERECHA.

APENDICE D

Otra combinación INVALIDA

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

rojo	rojo claro	
amarillo		
azul	azul medio	azul medio
cafe	cafe medio	

Esta combinación es inválida porque se utiliza el color rojo claro el cual está colocado a la DERECHA del color café. (recurra a la plantilla)

6.0 Para usar el color ROJO CLARO/VERDE es necesario que:

6.1 Se añada inmediatamente después del último color primario declarado, y que,

6.2 Algún cuadrito de la variante del ROJO CLARO se haya combinado PREVIAMENTE con el rojo primario.

Lo cual significa que: el color ROJO CLARO/VERDE NO puede combinarse con el color ROJO CLARO en el MISMO RENGLON

Ejemplos de combinaciones válidas.

rojo	rojo claro
amarillo	
rojo claro	verde

O

rojo	rojo claro
amarillo	
azul	
rojo claro	verde

## APENDICE D

Ejemplos de combinaciones INVALIDAS.

rojo		*
amarillo		
rojo claro		
_verde		

Esta combinación es inválida porque se omite la adición del ROJO CLARO con el rojo primario.

Otra combinación INVALIDA es:

rojo	+ rojo medio
amarillo	
azul	
rojo claro	
_verde	

Esta combinación es inválida porque se añade erróneamente un cuadrado de la variante rojo MEDIO con el rojo primario.

Otro ejemplo de combinación inválida es la siguiente:

rojo	rojo claro
amarillo	
azul	
rojo claro	+ rojo claro
_verde	

Esta combinación es inválida debido a que se añade el color rojo claro a el color rojo claro/verde.

APENDICE D

7.0 El color VERDE/ROJO CLARO se puede añadir a las combinaciones de colores siempre y cuando:

7.1 Se halla añadido previamente el color ROJO CLARO/VERDE (lo que significa que también debe contemplar las reglas para el rojo claro/verde).

7.2 Se le añade en su renglón un cuadrado del grupo ROJO CLARO.

Ejemplos de combinaciones válidas.

rojo	rojo claro
amarillo	
rojo claro	
verde	
verde	rojo claro
rojo claro	

o

rojo	rojo claro
amarillo	
azul	azul medio
rojo claro	
verde	
verde	rojo claro
rojo claro	

APENDICE D

Ejemplos de combinaciones INVALIDAS.

rojo	rojo claro
amarillo	
azul	
* verde rojo claro	rojo claro

Esta es una combinación inválida ya que, para añadir el color verde/rojo claro se requiere añadir previamente el color rojo claro/verde.

Otra combinación INVALIDA es:

rojo	rojo claro	
amarillo		
azul		
rojo claro verde	verde rojo claro	rojo claro

:> . . . . . |

Esta es una combinación es inválida, recuerde que los colores opcionales ocupan un renglón por separado.

# APENDICE E

## NOTAS DEL SQL

1.0	Introducción .....	85
1.1	Conceptos básicos .....	85
1.1.1	Atributos .....	85
1.1.2	Tabla .....	85
1.1.3	Base de Datos .....	86
1.2	Estructura de SQL .....	86
1.2.1	Claúsulas .....	86
1.2.2	Tipos de Interrogante .....	87
2.0	Selección .....	88
2.1	Claúsulas implicadas .....	88
2.2	Función .....	88
2.3	Como opera ? .....	88
2.4	Sintáxis .....	89
3.0	Proyección .....	90
3.1	Claúsulas implicadas .....	90
3.2	Función .....	90
3.3	Como opera ? .....	90
3.4	Sintáxis .....	90
4.0	Selección Condicional .....	91
4.1	Claúsulas implicadas .....	91
4.2	Función .....	91
4.3	Como opera ? .....	91
4.4	Sintáxis .....	92
5.0	Proyección Condicional .....	93
5.1	Claúsulas implicadas .....	93
5.2	Función .....	93
5.3	Como opera ? .....	93
5.4	Sintáxis .....	94
6.0	Unión .....	95
6.1	Claúsulas implicadas .....	95
6.2	Función .....	95
6.3	Como opera ? .....	95
6.4	Sintáxis .....	96

APENDICE E

7.0	Ordenamiento .....	97
7.1	Claúsulas implicadas .....	97
7.2	Función .....	97
7.3	Como opera ? .....	97
7.4	Sintáxis .....	98
8.0	Unico .....	99
8.1	Claúsulas implicadas .....	99
8.2	Función .....	99
8.3	Como opera ? .....	100
8.4	Sintáxis .....	101
9.0	Operaciones Agregadas .....	102
9.1	Claúsulas implicadas .....	102
9.2	Función .....	102
9.3	Como operan? .....	103
9.4	Sintáxis .....	105
10.0	Agrupamiento .....	106
10.1	Claúsulas implicadas .....	106
10.2	Función .....	106
10.3	Como opera ? .....	107
10.4	Sintáxis .....	109
11.0	Selección de Grupos .....	110
11.1	Claúsulas implicadas .....	110
11.2	Función .....	110
11.3	Como opera ? .....	111
11.4	Sintáxis .....	112

## APENDICE E

### 1.0 INTRODUCCION.

#### 1.1 CONCEPTOS BASICOS.

##### 1.1.1 ATRIBUTO.

La unidad básica una Base de Datos se denomina atributo, ejemplos de atributos son: el número del académico, el nombre del académico.

##### 1.1.2 TABLA.

Es una colección de atributos que representan a una entidad, por ejemplo la entidad de académicos se caracteriza por el número, nombre, por el RFC, grado y número de hijos.

Una tabla se representa de la forma siguiente:

tabla académicos

: acanum :	: acareg :	: acagra :	: acanom :	: acahij :
: 10 :	: CAJK621204 :	: PAS :	: CHAVEZ :	: 2 :
: 20 :	: PETE561209 :	: LIC :	: PEREZ :	: 3 :
: 30 :	: MAFG231231 :	: PAS :	: MACIAS :	: 2 :
: 40 :	: DERT341208 :	: DR :	: DELGADO :	: 1 :
: 50 :	: AIRE621224 :	: LIC :	: ABRIL :	: 5 :
: 60 :	: BATE121202 :	: DR :	: BASURTO :	: 4 :
: 70 :	: AATJ341023 :	: PAS :	: ARMAS :	: 2 :

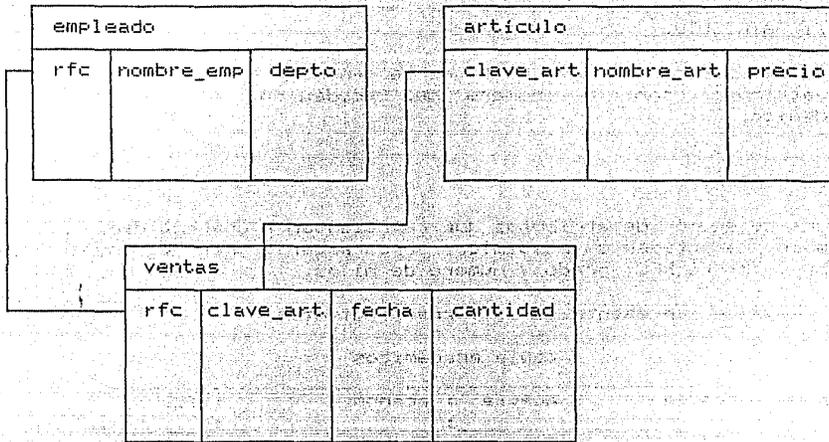
Las columnas de la tabla se llaman atributos, por ejemplo:

: acareg :	=	R. F. C. del académico
: CAJK621204 :		datos (o contenido) del atributo
: PETE561209 :		
: MAFG231231 :		
: DERT341208 :		
: AIRE621224 :		
: BATE121202 :		
: AATJ341023 :		

Los renglones de la tabla se denominan registros, por ejemplo:

: 10 :	: CAJK621204 :	: PAS :	: CHAVEZ :	: 2 :
--------	----------------	---------	------------	-------

1.1.3 Base de Datos. Regularmente una base de datos se compone por varias tablas relacionadas entre si.



En resumen:

- a las columnas de la tabla se denomina atributos.
- a los renglones de la tabla se les llama registros.
- una base de datos se compone por varias tablas relacionadas entre si

## 1.2. Estructura del SQL.

El SQL, (Structured Query Language) es un lenguaje parecido al idioma inglés diseñado para interrogar o consultar bases de datos.

Al SQL lo compone un conjunto de cláusulas (palabras) y un sistema estructurado de reglas sintácticas.

### 1.2.1 CLAUSULAS

Las cláusulas mas usuales del SQL son las siguientes:

CLAUSULA	FUNCION
select	declara la lista de atributos de los que se desea seleccionar datos.
from	indica el lugar (tabla o tablas) de procedencia de los atributos.
where	declara la condición para que se seleccionen o se rechazen registros.  la condición se basa en el resultado de una expresión booleana.

## APENDICE E

Dirijase al Apéndice B para obtener una referencia rápida de todas las cláusulas y sus funciones.

### 1.2.2. TIPOS DE INTERROGANTE.

SQL opera con estructuras de interrogación definidas, por medio de las cuales se pueden resolver toda una gama de necesidades de consulta de información.

Cada estructura de interrogación se compone por varias cláusulas.

TIPO DE INTERROGANTE	FUNCION	CLAUSULAS
SELECCION	Presenta	todos los registros de una tabla.
PROYECCION	Presenta	todo el contenido de un atributo o de varios atributos.
ORDENAMIENTO	Ordena	en forma ascendente o descendente los registros por el contenido de uno o mas atributos.

Remítase al Apéndice C si desea tener una referencia rápida de todos los tipos de interrogante con los que trabaja SQL.

A partir del siguiente capítulo estudiaremos cada tipo de interrogante.

2.0 SELECCION

2.1 CLAUSULAS IMPLICADAS

select  
from

remítase al Apéndice B para una descripción de las cláusulas.

2.2 FUNCION

El interrogante de selección presenta el contenido de todos los atributos (registros) de alguna tabla.

2.3 COMO OPERA

Suponga que desea obtener todos los registros de la tabla aca

ac anum	ac areg	ac agra	ac anom	ac ahij
10	CAJK621204	PAS	CHAVEZ	2
20	PETE561209	LIC	PEREZ	3
30	MAFG231231	PAS	MACIAS	2
40	DETR341208	DR	DELGADO	1
50	AIRE621224	LIC	ABRIL	5
60	BATE121202	DR	BASURTO	4
70	AATJ341023	PAS	ARMAS	2

El interrogante de selección secciona horizontalmente la tabla, es decir presenta los registros de la tabla.

ac anum	ac areg	ac agra	ac anom	ac ahij
---> 10	CAJK621204	PAS	CHAVEZ	2
---> 20	PETE561209	LIC	PEREZ	3
---> 30	MAFG231231	PAS	MACIAS	2
---> 40	DETR341208	DR	DELGADO	1
---> 50	AIRE621224	LIC	ABRIL	5
---> 60	BATE121202	DR	BASURTO	4
---> 70	AATJ341023	PAS	ARMAS	2

El listado que obtendriamos seria el siguiente:

ac anum	ac areg	ac agra	ac anom	ac ahij
10	CAJK621204	PAS	CHAVEZ	2
20	PETE561209	LIC	PEREZ	3
30	MAFG231231	PAS	MACIAS	2
40	DETR341208	DR	DELGADO	1
50	AIRE621224	LIC	ABRIL	5
60	BATE121202	DR	BASURTO	4
70	AATJ341023	PAS	ARMAS	2

observe que el listado es idéntico a todo el contenido de la tabla.

2.4 SINTAXIS

select \* (el \* significa todos los atributos)  
from tabla

selecciona los registros  
 de la tabla

select \*  
from aca/

presenta los registros de la tabla (o el contenido de todos  
 los atributos) de la tabla académicos.

## APENDICE E

### 3.0 PROYECCION

#### 3.1 CLAUSULAS IMPLICADAS

select  
from

remítase al Apéndice B para una descripción de las cláusulas.

#### 3.2 FUNCION

El interrogante de proyección presenta el contenido de los atributos que deseemos.

#### 3.3 COMO OPERA

Suponga que desea obtener un listado de todos los académicos, pero únicamente quiere saber el grado y nombre de cada uno de ellos. Para este efecto tendríamos que extraer el contenido de los atributos que deseemos.

<u>;</u> <u>v</u>	<u>;</u> <u>v</u>	<u>;</u> <u>v</u>	<u>;</u> <u>v</u>	<u>;</u> <u>v</u>
: acanum	: acareg	: acagra	: acanom	: acahij:
: 10	: CAJK621204	: PAS	: CHAVEZ	: 2 ;
: 20	: PETE561209	: LIC	: PEREZ	: 3 ;
: 30	: MAFG231231	: PAS	: MACIAS	: 2 ;
: 40	: DERT341208	: DR	: DELGADO	: 1 ;
: 50	: AIRE621224	: LIC	: ABRIL	: 5 ;
: 60	: BATE121202	: DR	: BASURTO	: 4 ;
: 70	: AATJ341023	: PAS	: ARMAS	: 2 ;

acanum acanom

.....  
10 CHAVEZ  
20 PEREZ  
30 MACIAS  
40 DELGADO  
50 ABRIL  
60 BASURTO  
70 ARMAS

Observe que tuvimos que seccionar verticalmente la tabla para obtener el contenido de los atributos solicitados.

Mediante el interrogante de proyección podemos obtener el contenido de solo aquellos atributos que indiquemos.

#### 3.4 SINTAXIS

select atributo 1.. atributo n  
from tabla

presenta el contenido de el (los) atributo(s)  
de la tabla

select acanum, acagra ó select acagra, acanum  
from aca/ from aca/

presenta el contenido de los atributos acanum y acagra  
de la tabla académicos. El orden en que declaramos los  
atributos en la cláusula de select

## APENDICE E

### 4.0 SELECCION CONDICIONAL

#### 4.1 CLAUSULAS IMPLICADAS

```
select  
from  
where y sus opciones  
and  
or  
between
```

remítase al Apéndice B para una descripción de las cláusulas.

#### 4.2 FUNCION

La selección básica presenta todos los registros de la tabla.

```
select *  
from tabla
```

La selección condicional nos permite obtener solo aquellos registros que cumplan con alguna condición.

#### 4.3 COMO OPERA ?

Suponga que desea saber el número, RFC, grado, nombre y número de hijos de aquellos académicos que tengan el grado de pasante (PAS).

Si utilizamos el interrogante de selección,

```
select *  
from aca/
```

```
-----  
:acanum : acareg      : acagra : acanom      : acahij :  
:-----: :-----: :-----: :-----: :-----: :  
---> : 10   : CAJK621204 : PAS       : CHAVEZ   : 2       :  
---> : 20   : PETE561209 : LIC       : PEREZ    : 3       :  
---> : 30   : MAFG231231 : PAS       : MACIAS   : 2       :  
---> : 40   : DERT341208 : DR        : DELGADO  : 1       :  
---> : 50   : AIRE621224 : LIC       : ABRIL    : 5       :  
---> : 60   : BATE121202 : DR        : BASURTO  : 4       :  
---> : 70   : AATJ341023 : PAS       : ARMAS    : 2       :
```

## APENDICE E

Tendríamos que elegir manualmente solo aquellos registros que cumplieran con la condición, grado= pasante:

```

-----
:acanum : acareg      : acagra : acanom      :acahij:
-----
----> : 10 : CAJK621204 : PAS * : CHAVEZ      : 2 :
      : 20 : PETE561209 : LIC   : PEREZ       : 3 :
----> : 30 : MAFG231231 : PAS * : MACIAS      : 2 :
      : 40 : DERT341208 : DR    : DELGADO     : 1 :
      : 50 : AIRE621224 : LIC   : ABRIL       : 5 :
      : 60 : BATE121202 : DR    : BASURTO    : 4 :
----> : 70 : AATJ341023 : PAS * : ARMAS       : 2 :

```

```

-----
:acanum : acareg      : acagra : acanom      :acahij:
-----
      10      CAJK621204      PAS      CHAVEZ      2
      30      MAFG231231      PAS      MACIAS      2
      70      AATJ341023      PAS      ARMAS      2

```

Nos podemos ahorrar este trabajo si indicamos la condición de selección de registros.

```

select *
from aca
where acagra= 'PAS*'

```

selecciona los registros  
de la tabla aca  
con la condición de que el grado sea pasante

### 4.4 SINTAXIS

```

select *
from tabla
where condición

```

selecciona los registros  
de la tabla  
que cumplan con la condición

## APENDICE E

### 5.0 PROYECCION CONDICIONAL

#### 5.1 CLAUSULAS IMPLICADAS

```

select
from
where y sus opciones
and
or
between
    
```

remítase al Apéndice B para una descripción de las cláusulas.

#### 5.2 FUNCION

La proyección básica presenta todos los datos del atributo o atributos especificados

```

select atributo 1.. atributo n
from tabla
    
```

Mediante la proyección condicional podemos pedir que se presenten los datos de uno o más atributos, siempre y cuando se cumpla con alguna condición.

#### 5.3 COMO OPERA ?

Suponga que desea saber los grados y los nombres de los académicos que tengan el grado de doctor (DR).

Si utilizamos el interrogante de proyección:

```

select acanom,acagra
from aca/
    
```

aca:	aca:	aca:	aca:	aca:
acanom:	acareg:	acagra:	acanom:	acahij:
10	CAJK621204	PAS	CHAVEZ	2
20	PETE561209	LIC	PEREZ	3
30	MAFG231231	PAS	MACIAS	2
40	DETR341208	DR	DELGADO	1
50	AIRE621224	LIC	ABRIL	5
60	BATE121202	DR	BASURTO	4
70	AATJ341023	PAS	ARMAS	2

Obtendríamos un listado de los grados y nombres existentes en la tabla.

```

acagra acanom
.....
PAS CHAVEZ
LIC PEREZ
PAS MACIAS
DR DELGADO
LIC ABRIL
DR BASURTO
PAS ARMAS
    
```

## APENDICE E

De este listado tendríamos que elegir manualmente los renglones que correspondan al grado de DR.

```
acagra acanom
```

```
.....
```

```
DR      DELGADO
```

```
DR      BASURTO
```

La proyección condicional nos permite elegir datos de uno o mas atributos que cumplan con alguna condición.

```
select acagra, acanom  
from aca  
where acagra = 'DR*'/
```

presenta aquellos grados y nombres de académicos con la condición de que el grado sea el de DR.

### 5.4 SINTAXIS

```
select lista de atributos  
from tabla  
where condición
```

selecciona los datos de una lista de atributos de alguna tabla que cumplan con cierta condición

6.0 UNION

6.1 CLAUSULAS IMPLICADAS

```

select
from
where y la opción
and

opcionalmente
.....
. or .
. between .
.....
    
```

remitase al Apéndice B para una descripción de las cláusulas.

6.2 FUNCION

El interrogante de unión une dos o mas tablas con el objeto de obtener de tablas separadas datos que estén relacionados.

6.3 COMO OPERA ?

El requisito para la unión de tablas consiste en que las tablas compartan al algún atributo en común.

```

-----
: acanum : acareg : acagra : acanom :
:-----:
: 10 : CAJK621204 : PAS : CHAVEZ :
: 20 : PETE561209 : LIC : PEREZ :
: 30 : MAFG231231 : PAS : MACIAS :
: 40 : DERT341208 : DR : DELGADO :
: 50 : AIRE621224 : LIC : ABRIL :
: 60 : BATE121202 : DR : BASURTO :
: 70 : AATJ341023 : PAS : ARMAS :
:-----:
    
```

\*

```

-----
: depnum : acanum : nombra 1 : nombra 2 :
:-----:
: 30 : 30 : TEC ACAD: TEC ACAD :
: 20 : 10 : INV : PROF TIT :
: 30 : 20 : PROF TIT: :
: 10 : 40 : TEC ACAD: :
: 20 : 10 : INV TIT : :
: 10 : 70 : PROF TIT: :
: 10 : 60 : TEC ACAD: TEC ACAD :
:-----:
    
```

## APENDICE E

Las tablas se pueden unir en virtud de que comparten el atributo de acanum.

```

-----
:depnum:acanum:nombra 1:nombra 2 :acanum:acareg :acgra:acanom :
:-----:-----:-----:-----:-----:-----:-----:-----:
: 30 :30 :TEC ACAD:TEC ACAD : 30 :MAFG231231 PAS : MACIAS:
: 20 :10 :INV :PROF TIT : 10 :CAJK621204 PAS : CHAVEZ:
: 30 :20 :PROF TIT: : 20 :PETE561209 LIC : PEREZ :
: 10 :40 :TEC ACAD: : 40 :BERT341208 DR :DELGADO:
: 20 :10 :INV TIT : : 10 :CAJK621204 PAS : CHAVEZ:
: 10 :70 :PROF TIT: : 70 :AATJ341023 PAS : ARMAS :
: 10 :60 :TEC ACAD:TEC ACAD : 60 :BATE121202 DR :BASURTO:
:-----:-----:-----:-----:-----:-----:-----:-----:

```

La tabla resultante de la unión puede ser utilizada para obtener registros o datos de algun o algunos atributos que cumplan con alguna condición.

### 6.4 SINTAXIS

select \* ó lista de atributos

from tabla1, tabla2

[1] where tabla1.atributo i = tabla2.atributo i

ó

select \* ó lista de atributos

from tabla1, tabla2

[1] where tabla1.atributo\_i = tabla2.atributo\_i and

[2] condición

[1] Se establece la relación existente entre las tablas

[2] En la unión al igual que en la selección y proyección podemos condicionar la presentación de ciertos datos.

select \*

from aca, acadep

[1] where aca.acanum= acadep.acanum

[2] and acagra= 'PAS#'/

presenta los registros de la unión de las tablas aca y acadep, con la condición de que el atributo acanum es común a ambas tablas y que el grado sea igual a pasante.

7.0 ORDENAMIENTO

7.1 CLAUSULAS IMPLICADAS

select  
from

```
opcionalmente
.....
.where y sus opciones .
.      and .
.      or .
.      between .
.....
```

order by y su opción  
desc

remítase al Apéndice B para una descripción de las cláusulas.

7.2 FUNCION

El interrogante de ordenamiento, ordena los registros de una tabla (o unión de dos o mas tablas) en base a los datos de uno o mas atributos.

7.3 COMO OPERA ?

Suponga que desea obtener un listado de los grados y nombres de los académicos, pero ahora usted desea el listado ordenado alfabeticamente por el nombre del académico.

Para obtener este listado tendríamos que seleccionar los registros de la tabla académicos

```
select *
from aca/
```

```
      : acanum : acareg      : acagra : acanom      : acahij:
---> : 10      : CAJK621204 : PAS    : CHAVEZ      : 2 :
---> : 20      : PETE561209 : LIC    : PEREZ       : 3 :
---> : 30      : MAFG231231 : PAS    : MACIAS      : 2 :
---> : 40      : DERT341208 : DR     : DELGADO     : 1 :
---> : 50      : AIRE621224 : LIC    : ABRIL       : 5 :
---> : 60      : BATE121202 : DR     : BASURTO     : 4 :
---> : 70      : AATJ341023 : PAS    : ARMAS       : 2 :
```

ordenamos los registros en base a los datos de acagra

```
      acanum      acareg      acagra      acanom      acahij
: 50      : AIRE621224 : LIC        : ABRIL       : 5 :
: 70      : AATJ341023 : PAS        : ARMAS       : 2 :
: 60      : BATE121202 : DR         : BASURTO     : 4 :
: 10      : CAJK621204 : PAS        : CHAVEZ      : 2 :
: 40      : DERT341208 : DR         : DELGADO     : 1 :
: 30      : MAFG231231 : PAS        : MACIAS      : 2 :
: 20      : PETE561209 : LIC        : PEREZ       : 3 :
```

Y de este listado proyectamos los atributos de acagra y acanom.

```

acagra      acanom
: LIC       : ABRIL      :
: PAS       : ARMAS       :
: DR        : BASURTO     :
: PAS       : CHAVEZ      :
: DR        : DELGADO     :
: PAS       : MACIAS      :
: LIC       : PEREZ       :
    
```

Este trabajo lo podemos omitir si utilizamos el interrogante de ordenamiento,

```

select acagra, acanom
from aca
order by acanom/
    
```

por omisión order by ordena en forma ascendente (A..Z, 1..100). si queremos ordenar en forma descendente (Z..A, 100..1) agregamos la cláusula desc a order by .

order by atributo desc.

#### 7.4 SINTAXIS

```

select * ó lista de atributos
from tabla1
[1] order by atributo_1, ... atributo_n
    
```

ó

```

select * ó lista de atributos
from tabla1, tabla2
[2] where condición
[1] order by atributo_1, ...atributo_n
    
```

- [1] En el orden en el que se especifican los atributos, se ordenan y reordenan los registros.

Para especificar el ordenamiento descendente se anade la cláusula desc al atributo

order by atributo\_1 desc

- [2] Al igual que los interrogantes anteriores, se pueden ordenar solo aquellos registros que cumplan con alguna condición

8.0 UNICO

8.1 CLAUSULAS IMPLICADAS

```
select    unique
from
```

opcionalmente

```
.....
. where y sus opciones .
.           and         .
.           or          .
.           between    .
.....
```

remítase al Apéndice B para una descripción de las cláusulas.

8.2 FUNCION

Suponga que desea obtener un listado de los diferentes grados académicos, si utilizamos el interrogante de proyección:

```
select acagra
from aca/
```

:

aca/		V	aca/	
aca/	aca/	aca/	aca/	aca/
10	CAJK621204	PAS	CHAVEZ	2
20	PETE561209	LIC	PEREZ	3
30	MAFG231231	PAS	MACIAS	2
40	DETR341208	DR	DELGADO	1
50	AIRE621224	LIC	ABRIL	5
60	BATE121202	DR	BASURTO	4
70	AATJ341023	PAS	ARMAS	2

obtendríamos el siguiente listado

```
PAS
LIC
PAS
DR
LIC
DR
PAS
```

este listado nos sería útil, pero de este tendríamos que desechar los grados que se repitan, y elegir los grados únicos

## APENDICE E

Ahora bien, la tarea se podria simplificar si ordenamos los registros por a los grados en forma ascendente.

```

ac anum   ac areg   ac agra   ac anom   ac ahij
: 50      : AIRE621224   : LIC    : ABRIL    : 5 :
: 70      : AATJ341023   : PAS    : ARMAS    : 2 :
: 60      : BATE121202   : DR     : BASURTO  : 4 :
: 10      : CAJK621204   : PAS    : CHAVEZ   : 2 :
: 40      : DERT341208   : DR     : DELGADO  : 1 :
: 30      : MAFG231231   : PAS    : MACIAS   : 2 :
: 20      : PETE561209   : LIC    : PEREZ    : 3 :
    
```

proyectamos acagra

```

DR
DR
LIC
LIC
PAS
PAS
PAS
    
```

aun asi, nuestra tarea de localizar los diferentes grados sigue siendo tardada, ya que debemos de eliminar los grados duplicados

```

DR
DR * duplicado
LIC
LIC * duplicado
PAS
PAS * duplicado
PAS * duplicado
    
```

para obtener un listado de los grados unico

```

DR
LIC
PAS
    
```

El interrogante unico, ordena los datos de algun atributo en forma ascendente y elimina los datos duplicados.

### 8.3 COMO OPERA ?

Desea saber los grados unicos de la tabla academicos.

```

select unique acagra
from aca/
    
```

```

-----
: ac anum: ac areg   : ac agra   : ac anom   : ac ahij:
:-----:-----:-----:-----:-----:
: 10     : CAJK621204 : PAS      : CHAVEZ    : 2 :
: 20     : PETE561209 : LIC      : PEREZ     : 3 :
: 30     : MAFG231231 : PAS      : MACIAS    : 2 :
: 40     : DERT341208 : DR       : DELGADO   : 1 :
: 50     : AIRE621224 : LIC      : ABRIL     : 5 :
: 60     : BATE121202 : DR       : BASURTO   : 4 :
: 70     : AATJ341023 : PAS      : ARMAS     : 2 :
:-----:-----:-----:-----:-----:
    
```

## APENDICE E

Se ordenan en forma ascendente los registros por el atributo del que se quiere saber los datos unicos.

acahum	acareg	acagra	acahom	acahij
: 50	: AIRE621224	: LIC	: ABRIL	: 5
: 70	: AATJ341023	: PAS	: ARMAS	: 2
: 60	: BATE121202	: DR	: BASURTO	: 4
: 10	: CAJK621204	: PAS	: CHAVEZ	: 2
: 40	: DERT341208	: DR	: DELGADO	: 1
: 30	: MAFG231231	: PAS	: MACIAS	: 2
: 20	: PETE561209	: LIC	: PEREZ	: 3

Proyecta el atributo elegido

acagra

: DR :  
: DR :  
: LIC :  
: LIC :  
: PAS :  
: PAS :  
: PAS :

elimina los datos duplicados

acagra

: DR :  
: DR \* :  
: LIC :  
: LIC \* :  
: PAS :  
: PAS \* :  
: PAS \* :

y da como resultado

acagra

: DR :  
: LIC :  
: PAS :

### 8.4 SINTAXIS

[1] select unique atributo  
from tabla

ó

[1] select unique atributo  
from tabla  
[2] where condición

[1] La cláusula unique se declara junto a la cláusula select

[2] La presentación de datos unicos se puede limitar a alguna condición

## APENDICE E

### 9.0 OPERACIONES AGREGADAS

#### 9.1 CLAUSULAS IMPLICADAS

Las funciones agregadas se utilizan con las cláusulas select y having (vease 11.0)

```
select count(*), avg(atributo), sum(atributo),  
max(atributo), min(atributo)
```

```
from tabla
```

opcionalmente

```
.....  
. where y sus opciones .  
. and .  
. or .  
. between .  
.....
```

remitase al Apéndice B para una descripción de las cláusulas.

#### 9.2 FUNCION

Suponga que desea saber la cantidad de académicos existente. Para obtener esta cifra podríamos utilizar el interrogante de selección

```
select *  
from aca
```

	:acanum :	acareg	: acagra :	acanon	: acanhij:
---> :	10	: CAJK621204	: PAS	: CHAVEZ	: 2 :
---> :	20	: PETE561209	: LIC	: PEREZ	: 3 :
---> :	30	: MAFG231231	: PAS	: MACIAS	: 2 :
---> :	40	: DERT341208	: DR	: DELGADO	: 1 :
---> :	50	: AIRE621224	: LIC	: ABRIL	: 5 :
---> :	60	: BATE121202	: DR	: BASURTO	: 4 :
---> :	70	: AATJ341023	: PAS	: ARMAS	: 2 :

y de este listado tendríamos que contar manualmente los registros para saber la cifra que estamos buscando

La operación agregada count(\*) se emplea para contar registros.

APENDICE E

Suponga ahora que desea saber el numero de hijos que tienen en promedio los académicos que son pasantes.

```
select acagra,acahij
from aca
where acagra= 'PAS*'/
```

para que nos presentara el numero de hijos de cada pasante

		V			V
: acanum:	acareg	: acagra	: acanom	: acahij	:
: 10	: CAJK621204	: PAS	: CHAVEZ	: 2	:
: 20	: PETE561209	: LIC	: PEREZ	: 3	:
: 30	: MAFG231231	: PAS	: MACIAS	: 2	:
: 40	: DERT341208	: DR	: DELGADO	: 1	:
: 50	: AIRE621224	: LIC	: ABRIL	: 5	:
: 60	: BATE121202	: DR	: BASURTO	: 4	:
: 70	: AATJ341023	: PAS	: ARMAS	: 2	:
		PAS		2	
		PAS		2	
		PAS		2	

de tal manera que podamos obtener el total de hijos de los pasantes

2
2
2
6

contamos el numero de pasantes (3), y dividimos el total de hijos entre el numero de pasantes para obtener el promedio de hijos.

total de hijos/ numero de pasantes      6/3= 2

Las operaciones agregadas avg(), sum(), max(), min() se utilizan para realizar cálculos con el contenido de atributos de tipo numerico (enteros, fechas, punto flotante).

9.3 COMO OPERAN ?

count(\*), cuenta registros, ya sea el total de registros de una tabla (o unión de dos o mas tablas) asi como una cierta cantidad de registros que cumplan con alguna condición.

Cuantos académicos son pasantes.

```
select count(*), acagra
from aca
where acagra= 'PAS*'/
```

APENDICE E

seleccionamos los registros que tengan el grado de PAS

aca-num:	aca-reg	aca-gra	aca-nom	aca-hij:
10	CAJK621204	* PAS	CHAVEZ	2
20	PETE561209	LIC	PEREZ	3
30	MAFG231231	* PAS	MACIAS	2
40	DERT341208	DR	DELGADO	1
50	AIRE621224	LIC	ABRIL	5
60	BATE121202	DR	BASURTO	4
70	AATJ341023	* PAS	ARMAS	2

y los contamos

	aca-num:	aca-reg	aca-gra	aca-nom	aca-hij:
-	10	CAJK621204	* PAS	CHAVEZ	2
-	30	MAFG231231	* PAS	MACIAS	2
-	70	AATJ341023	* PAS	ARMAS	2

= 3

avg(atributo), sum(atributo), max(atributo), min(atributo) efectúan cálculos de todo el contenido de algún atributo numérico, o de algún segmento de datos numérico que cumplan con alguna condición.

Cuántos hijos en total tienen todos los académicos

```
select sum(aca-hij)
from aca/
```

aca-num:	aca-reg	aca-gra	aca-nom	aca-hij:
10	CAJK621204	PAS	CHAVEZ	2
20	PETE561209	LIC	PEREZ	3
30	MAFG231231	PAS	MACIAS	2
40	DERT341208	DR	DELGADO	1
50	AIRE621224	LIC	ABRIL	5
60	BATE121202	DR	BASURTO	4
70	AATJ341023	PAS	ARMAS	2

= 19

## APENDICE E

### 9.4 SINTAXIS

[1] select op agregada 1, op agregada n,  
from tabla

ó

[1] select op agregada 1, op agregada n, atributo  
from tabla  
[2] where condición

[1] La(s) operación(es) agregada(s) se declara junto a la cláusula de select.

Las operaciones agrgadas y unique son incompatibles en virtud de que tienen funciones distintas, es por esta razón por la que no pueden aparecer al mismo tiempo junto a la cláusula de select.

[2] La aplicación de la operación agregada la podemos limitar a un segmento de datos que cumplan con alguna condición.



APENDICE E

agrupamos los datos comunes, y obtenemos el promedio de cada grupo.

```

: acagra :acahij:
: DR : 1 :
: DR : 4 : DR
                    5/2= 2.5

: LIC : 3 : LIC
: LIC : 5 :
                    8/2= 4

: PAS : 2 :
: PAS : 2 : PAS
: PAS : 2 :
                    6/2= 3
    
```

ambas estrategias nos resuelven el problema pero de forma muy laboriosa.

10.3 COMO OPERA ?

El interrogante de agrupamiento ordena y agrupa registros en base al contenido de algun atributo, con el objeto de aplicar alguna operación agregada sobre los grupos formados.

cual es el numero maximo de hijos por cada grado académico.

```

select max(acahij), acagra
from aca
group by acagra/
    
```

en virtud de que se requiere agrupar por el grado, el interrogante de agrupamiento, ordena los registros por este atributo.

```

: acanum: acareg : acagra :acanom :acahij:
:-----:-----:-----:-----:-----:
: 40 : DERT341208 : DR : DELGADO : 1 :
: 60 : BATE121202 : DR : BASURTO : 4 :
: 20 : PETE561209 : LIC : PEREZ : 3 :
: 50 : AIRE621224 : LIC : ABRIL : 5 :
: 10 : CAJK621204 : PAS : CHAVEZ : 2 :
: 30 : MAFG231231 : PAS : MACIAS : 2 :
: 70 : AATJ341023 : PAS : ARMAS : 2 :
    
```

y los agrupa

```

: acanum: acareg : acagra :acanom :acahij:
:-----:-----:-----:-----:-----:
: 40 : DERT341208 : DR : DELGADO : 1 :
: 60 : BATE121202 : DR : BASURTO : 4 :
: 20 : PETE561209 : LIC : PEREZ : 3 :
: 50 : AIRE621224 : LIC : ABRIL : 5 :
: 10 : CAJK621204 : PAS : CHAVEZ : 2 :
: 30 : MAFG231231 : PAS : MACIAS : 2 :
: 70 : AATJ341023 : PAS : ARMAS : 2 :
    
```

APENDICE E

Una vez formados los grupos se procede a aplicar la operación agregada sobre los mismos.

acanam:	acareg	acagra	acanom	acahij:
				max
: 40	: DERT341208	: DR	: DELGADO	: 1 :
: 60	: BATE121202	: DR	: BASURTO	: 4 :
		DR		4
				max
: 20	: PETE561209	: LIC	: PEREZ	: 3 :
: 50	: AIRE621224	: LIC	: ABRIL	: 5 :
		LIC		5
				max
: 10	: CAJK621204	: PAS	: CHAVEZ	: 2 :
: 30	: MAFG231231	: PAS	: MACIAS	: 2 :
: 70	: AATJ341023	: PAS	: ARMAS	: 2 :
		PAS		2

obtenemos como resultado,

acagra	max(acahij)
DR	4
LIC	5
PAS	2

## APENDICE E

### 10.4 SINTAXIS

- (1) select op\_agregada\_1, op\_agregada\_n, atributo\_j  
from tabla  
(2) group by atributo\_j

ó

- (1) select op\_agregada\_1, op\_agregada\_n, atributo\_j  
from tabla  
(3) where condición  
(2) group by atributo\_j

- (1) Para utilizar la cláusula group by se requiere forzosamente la declaración de alguna operación agregada.

Este requisito es fundamental ya que no tiene ningún caso ordenar y agrupar si no se aplica alguna operación agregada.

- (2) El agrupamiento de registros se hace en base al contenido de algún atributo.

				max	
: 20	: PETE561209	: LIC	: PEREZ	: 3	:
: 50	: AIRE621224	: LIC	: ABRIL	: 5	:
		LIC		5	

				max	
: 10	: CAJK621204	: PAS	: CHAVEZ	: 2	:
: 30	: MAFG231231	: PAS	: MACIAS	: 2	:
: 70	: AATJ341023	: PAS	: ARMAS	: 2	:
		PAS		2	

Observe que el atributo por el que se agrupa, en este caso grado debe de declararse junto a la cláusula de select.

```
select max(aca hij), acagra
from aca
group by acagra
```

de tal manera que podamos saber el nombre del grupo (caracterizado por el contenido del atributo) y su resultado correspondiente a la operación agregada.

En virtud de que group by ordena y agrupa registros, la cláusula de order by sale sobrando, por esta razón la cláusula de order by no debe de utilizarse en un interrogante de agrupamiento.

- (3) Al igual que en todos los interrogantes anteriores el agrupamiento se puede condicionar a un segmento de la tabla.

11.0 SELECCION DE GRUPOS

11.1 CLAUSULAS IMPLICADAS.

```
select op agregada 1... op agregada n, atributo_i
from tabla
```

```
opcionalmente
.....
. where y sus opciones .
.         and           .
.         or            .
.         between       .
.....
```

```
group by atributo_i
having operación agregada
```

remítase al Apéndice B para una descripción de las cláusulas.

11.2 FUNCION.

Suponga que desea saber que grados tienen mas de dos académicos.

Este problema lo podríamos atacar mediante el interrogante de agrupamiento.

```
select count(*), acagra
from tabla
group by acagra
```

Para saber la cantidad de académicos que hay en cada grado.

```
      : acanum: acareg      : acagra : acanom      : acahij:
count(*)
-      : 40      : DERT341208 : DR      : DELGADO     : 1      :
-      : 60      : BATE121202 : DR      : BASURTO     : 4      :
2
      DR
count(*)
-      : 20      : PETE561209 : LIC     : PEREZ       : 3      :
-      : 50      : AIRE621224 : LIC     : ABRIL       : 5      :
2
      LIC
count(*)
-      : 10      : CAJK621204 : PAS     : CHAVEZ      : 2      :
-      : 30      : MAFG231231 : PAS     : MACIAS      : 2      :
-      : 70      : AATJ341023 : PAS     : ARMAS       : 2      :
3
      PAS
```

de tal manera que podamos elegir aquellos grados que tengan mas de dos registros.

```
acagra count(*)
DR      2      2 mayor 2 ? no
LIC     2      2 mayor 2 ? no
PAS     3      3 mayor 2 ? si
```

APENDICE E

finalmente obtenemos

```
acagra count(*)
PAS      3
```

en virtud de que solo el grado de PAS tiene mas de 2 registros.

11.3 COMO OPERA.

El interrogante de selección de grupos acepta o rechaza grupos de registros formados previamente con group by, en base al resultado de alguna operación agregada.

Que grados tienen un promedio mayor a 3 hijos

```
select acagra, avg(acahij)
from aca
group by acagra
having avg(acahij) > 3/
```

En primer lugar se deben de formar los grupos de grados. para este efecto se ordenan y agrupan los registros por el atributo de acagra.

acanam:	acareg	acagra	acanam	acahij:
40	DETR341208	DR	DELGADO	1
60	BATE121202	DR	BASURTO	4
20	PETE561209	LIC	PEREZ	3
50	AIRE621224	LIC	ABRIL	5
10	CAJK621204	PAS	CHAVEZ	2
30	MAFG231231	PAS	MACIAS	2
70	AATJ341023	PAS	ARMAS	2

se obtiene el promedio de hijos por grupo

acanam:	acareg	acagra	acanam	acahij:	avg
40	DETR341208	DR	DELGADO	1	(4+1)/2= 2.5
60	BATE121202	DR	BASURTO	4	
		DR			
20	PETE561209	LIC	PEREZ	3	(3+5)/2= 4
50	AIRE621224	LIC	ABRIL	5	
		LIC			
					avg
10	CAJK621204	PAS	CHAVEZ	2	(2+2+2)/3= 2
30	MAFG231231	PAS	MACIAS	2	
70	AATJ341023	PAS	ARMAS	2	
		PAS			

se obtiene como resultado,

```
acagra avg(acahij)
DR      2.5
LIC     4
PAS     2
```

y se chequea que grupo(s) tiene(n) un promedio mayor a 3

```
acagra avg(acahij)
DR      2.5      2.5 > 3 ? no
LIC     4        4 > 3 ? si
PAS     2        2 > 3 ? no
```

se obtiene como resultado

```
acagra avg(acahij)
LIC     4
```

#### 11.4 SINTAXIS.

```
select op_agregada_1, op_agregada_n, atributo_j
from tabla
(1) group by atributo_j
(2) having operación agregada
```

ó

```
select op_agregada_1, op_agregada_n, atributo_j
from tabla
(3) where condición
(1) group by atributo_j
(2) having operación agregada
```

- (1) El requisito para declarar la cláusula having consiste en que se halla declarado previamente la cláusula group by.

Este requisito es fundamental, ya que no tiene ningún caso seleccionar o rechazar grupos si estos no se formaron con anterioridad.

- (2) La cláusula group by utiliza el resultado de la operación agregada para establecer el criterio para la selección o rechazo de grupos.
- (3) Al igual que todos los interrogantes anteriores, la selección de grupos se puede condicionar a un subconjunto de la tabla

## APENDICE F

### PROGRAMA DE ENTRENAMIENTO

Duración: 8 hrs= 2 hrs por 4 días consecutivos.

TEMARIO.

SQL, Lenguaje de Interrogación Estructurado a Bases de Datos.

#### I. PRIMER DIA.

=====

##### 1.0 Introducción

###### 1.1 Conceptos básicos

- 1.1.1 Atributos
- 1.1.2 Tabla

###### 1.2 Estructura de SQL

- 1.2.1 Cláusulas
- 1.2.2 Tipos de Interrogante

#### TIPOS DE INTERROGANTE.

##### 2.0 Selección

- 2.1 Cláusulas implicadas
- 2.2 Función
- 2.3 Como opera ?
- 2.4 Sintaxis

##### 3.0 Proyección

- 3.1 Cláusulas implicadas
- 3.2 Función
- 3.3 Como opera ?
- 3.4 Sintaxis

#### II. SEGUNDO DIA

=====

##### 4.0 Selección Condicional

- 4.1 Cláusulas implicadas
- 4.2 Función
- 4.3 Como opera ?
- 4.4 Sintaxis

##### 5.0 Proyección Condicional

- 5.1 Cláusulas implicadas
- 5.2 Función
- 5.3 Como opera ?
- 5.4 Sintaxis

## APENDICE F

### 6.0 Unión

- 6.1 Cláusulas implicadas
- 6.2 Función
- 6.3 Como opera ?
- 6.4 Sintáxis

### III. TERCER DIA

=====

### 7.0 Ordenamiento

- 7.1 Cláusulas implicadas
- 7.2 Función
- 7.3 Como opera ?
- 7.4 Sintáxis

### 8.0 Unico

- 8.1 Cláusulas implicadas
- 8.2 Función
- 8.3 Como opera ?
- 8.4 Sintáxis

### 9.0 Operaciones Agregadas

- 9.1 Cláusulas implicadas
- 9.2 Función
- 9.3 Como operan?
- 9.4 Sintáxis

### IV. CUARTO DIA

=====

### 10.0 Agrupamiento

- 10.1 Cláusulas implicadas
- 10.2 Función
- 10.3 Como opera ?
- 10.4 Sintáxis

### 11.0 Selección de Grupos

- 11.1 Cláusulas implicadas
- 11.2 Función
- 11.3 Como opera ?
- 11.4 Sintáxis

**ESTRATEGIA PARA EL DISEÑO DE INTERFASES PARA EL USUARIO DE LENGUAJES DE INTERROGACION A BASES DE DATOS**

1. Defina el conjunto de cláusulas más usuales del lenguaje de interrogación a utilizar.

por ejemplo las cláusulas del SQL.

select, from, where, between, and, or, order by, desc, unique, count(\*), max(), min(), avg(), sum(), group by, having.

2. Defina las estructuras sintácticas (combinaciones de cláusulas válidas) del lenguaje. (véase Apéndices B y C)

**PROYECCION**

```
select
from
opcionalmente where y sus opciones and , or, between
```

**ORDENAMIENTO**

```
select
from
opcionalmente where y sus opciones and , or, between
order by y su opción desc
```

**UNICO**

```
select unique
from
opcionalmente where y sus opciones and , or, between
```

**OPERACIONES AGREGADAS**

```
select count(*), avg(atributo), sum(atributo),
max(atributo), min(atributo)
from tabla
opcionalmente where y sus opciones and , or, between
```

**AGRUPAMIENTO**

```
select op_agregada_1,... op_agregada_n, atributo_i
from tabla
opcionalmente where y sus opciones and , or, between
group by atributo_i
```

**SELECCION DE GRUPOS**

```
select op_agregada_1,... op_agregada_n, atributo_i
from tabla
opcionalmente where y sus opciones and , or, between
group by atributo_i
having operacion agregada
```

## APENDICE G

3. Agrupe las cláusulas en términos de las estructuras sintácticas del lenguaje.

select from where (and,or,between)	order by (desc)	unique	operaciones agregadas  group by having... op_agregadas
---------------------------------------------	--------------------	--------	-----------------------------------------------------------------

4. Genere un código de preferencia mnemónico que refleje la compatibilidad e incompatibilidad de las cláusulas. (véase Fig. 3 y Apéndice D)

## APENDICE H

### INTERROGANTES EN SQL UTILIZADOS EN LA TAREA DE VERIFICACION.

#### INTERROGANTES SIN ERRORES DE SINTAXIS

##### INTERROGANTE DE SELECCION

```
select *  
from academ  
where num_academico between 10 and 50/
```

```
select *  
from academ, acaddepto  
where academ.num_academico=acaddepto.num_academico  
and hijos > 1/
```

```
select *  
from academ, acaddepto  
where academ.num_academico=acaddepto.num_academico/
```

##### INTERROGANTE DE PROYECCION

```
select nom_academico, grado  
from academ  
where hijos > 2/
```

```
select nom_academico, nombramiento_1  
from academ, acaddepto  
where academ.num_academico=acaddepto.num_academico  
and num_academico > 10/
```

```
select rfc, salario  
from academ, acaddepto  
where academ.num_academico=acaddepto.num_academico/
```

##### INTERROGANTE DE ORDENAMIENTO

```
select rfc, hijos  
from academ  
where grado= 'dr*'  
order by rfc/
```

```
select nombramiento_1, nombramiento_2  
from acaddepto, academ  
where acaddepto.num_academico=academ.num_academico  
and hijos > 1  
order by nombramiento_1/
```

```
select rfc, salario  
from academ, acaddepto  
where academ.num_academico=acaddepto.num_academico  
order by salario/
```

## APENDICE H

### INTERROGANTES SIN ERRORES DE SINTAXIS

#### INTERROGANTE DE UNICO

```
select unique salario
from acaddepto
where nombramiento_1= 'tec acad*'/
```

```
select unique hijos
from acaddepto,academ
where acaddepto.num_academico=academ.num_academico
and salario > 10000 /
```

```
select unique grado
from acaddepto,academ
where acaddepto.num_academico=academ.num_academico
and nombramiento_1= 'tec acad*'/
```

#### INTERROGANTE DE OPERACIONES AGREGADAS

```
select count(*), grado
from academ
where grado= 'lic*'/
```

```
select avg(hijos), avg(salario)
from academ, acaddepto
where academ.num_academico=acaddepto.num_academico/
```

```
select max(salario), grado
from acaddepto,academ
where acaddepto.num_academico=academ.num_academico
and grado= 'lic*'/
```

#### INTERROGANTE DE AGRUPAMIENTO

```
select avg(hijos), max(hijos), hijos
from academ
where num_academico between 10 and 40
group by hijos/
```

```
select max(salario),grado
from academ,acaddepto
where academ.num_academico=acaddepto.num_academico
group by grado/
```

```
select sum(hijos),nombramiento_1
from academ,acaddepto
where academ.num_academico=acaddepto.num_academico
and nombramiento_1= 'tec acad*'
group by nombramiento_1/
```

## APENDICE H

### INTERROGANTES SIN ERRORES DE SINTAXIS

#### INTERROGANTE DE SELECCION DE GRUPOS

```
select nombramiento_1, count(*)  
from acaddepto  
where salario > 10000  
group by nombramiento_1  
having count(*) > 2/
```

```
select grado, max(salario)  
from academ, acaddepto  
where academ.num_academico=acaddepto.num_academico  
group by grado  
having max(salario) > 10000/
```

```
select nombramiento_1, min(hijos)  
from acaddepto, academ  
where acaddepto.num_academico=academ.num_academico  
and grado= 'lic#'  
group by nombramiento_1  
having avg(hijos) > 2/
```

## APENDICE H

### INTERROGANTES CON ERRORES DE SINTAXIS

#### INTERROGANTE DE SELECCION

```
from academ
select *
where num_academico between 10 and 50/

select *
where academ.num_academico=acadepto.num_academico
and hijos > 1/
from academ, acadepto

where academ.num_academico=acadepto.num_academico/
select *
from academ, acadepto
```

#### INTERROGANTE DE PROYECCION

```
from academ
select nom_academico, grado
where hijos > 2/

select nom_academico, nombramiento_1
where academ.num_academico=acadepto.num_academico
and num_academico > 10/
from academ,acadepto

where academ.num_academico=acadepto.num_academico/
select rfc, salario
from academ,acadepto
```

#### INTERROGANTE DE ORDENAMIENTO

```
select avg(hijos), hijos
from academ
where grado= 'dr*'
group by hijos
having avg(hijos)>2
order by hijos/

select unique nombramiento_1
from acadepto, academ
where acadepto.num_academico=academ.num_academico
and hijos > 1
order by nombramiento_1/

select rfc, salario
from academ, acadepto
order by salario/
where academ.num_academico=acadepto.num_academico
```

## APENDICE H

### INTERROGANTES CON ERRORES DE SINTAXIS

#### INTERROGANTE DE UNICO

```
select unique salario *
from acaddepto
where nombramiento_1= 'tec acad*'
group by salario
having unique salario= 10,000/
```

```
select unique hijos
from acaddepto,academ
where acaddepto.num_academico=academ.num_academico
and salario > 10000
group by hijos/
```

```
select unique grado
where acaddepto.num_academico=academ.num_academico
and nombramiento_1= 'tec acad*'
from acaddepto,academ/
```

#### INTERROGANTE DE OPERACIONES AGREGADAS

```
select count(*), grado
from academ
where grado= 'lic*'
order by grado/
```

```
select avg(hijos), avg(salario)
where academ.num_academico=acaddepto.num_academico
from academ, acaddepto/
```

```
select max(salario), unique salario, grado
from acaddepto,academ
where acaddepto.num_academico=academ.num_academico
and grado= 'lic'*/
```

#### INTERROGANTE DE AGRUPAMIENTO

```
select avg(hijos), max(hijos), grado
from academ
where num_academico between 10 and 40
order by grado
group by grado/
```

```
select grado
from academ,acaddepto
where academ.num_academico=acaddepto.num_academico
group by grado/
```

```
select sum(hijos),nombramiento_1
from academ,acaddepto
group by nombramiento_1
where academ.num_academico=acaddepto.num_academico
and nombramiento_1= 'tec acad'*/
```

## APENDICE H

### INTERROGANTES CON ERRORES DE SINTAXIS

#### INTERROGANTE DE SELECCION DE GRUPOS

```
select nombramiento_1, count(*)
from acadepto
group by nombramiento_1
having count(*) > 2/
where salario > 10000
```

```
select grado, max(salario)
from academ,acadepto
where academ.num_academico=acadepto.num_academico
group by grado
having salario > 10000/
```

```
select nombramiento_1, min(hijos)
from acadepto,academ
where acadepto.num_academico=academ.num_academico
and grado= 'lic*'
having min(hijos) > 2/
```

## APENDICE I

### PREGUNTAS UTILIZADAS EN LA TAREA DE CONSTRUCCION

#### INTERROGANTE DE SELECCION

Lista los rfc, nombres y número de hijos de los académicos con la condición de que el número del académico este entre el 20 y 40.

Lista el rfc, grado, nombre, primer nombramiento y segundo nombramiento de todos los académicos

#### INTERROGANTE DE PROYECCION

Lista los números y rfc de los académicos que tengan entre 1 y 4 hijos.

Lista el grado, nombre, primer nombramiento y segundo nombramiento de los académicos cuyo primer nombramiento sea 'tec acad\*'

#### INTERROGANTE DE ORDENAMIENTO

Lista los nombres y grados de los académicos con la condición de que el grado sea 'pas\*', ordenados por el grado en forma descendente

Lista los grados, nombres y primer nombramiento de los académicos que tengan 2 hijos, ordenados por el nombre de los académicos.

#### INTERROGANTE DE UNICO

Lista los salarios sin repetición (únicos) de los académicos que tengan el grado de 'pas\*'

Lista los grados sin repetición (únicos) de los académicos que tengan más de dos hijos.

## APENDICE I

### INTERROGANTE DE OPERACIONES AGREGADAS

Calcula el máximo y mínimo de los salarios con la condición de que el primer nombramiento sea 'tec acad\*'.  
.

Calcula el promedio de hijos de los académicos cuyo primer nombramiento sea 'tec acad\*'.  
.

### INTERROGANTE DE AGRUPAMIENTO

Calcula el mínimo de hijos por cada grado, con la condición de que los números de los académicos estén entre 1 y 4.  
.

Calcula el promedio de hijos por cada primer nombramiento  
.

### INTERROGANTE DE SELECCION DE GRUPOS

Lista los grados (grupos) que tengan en promedio más de dos hijos con la condición de que el número del académico sea mayor que 10  
.

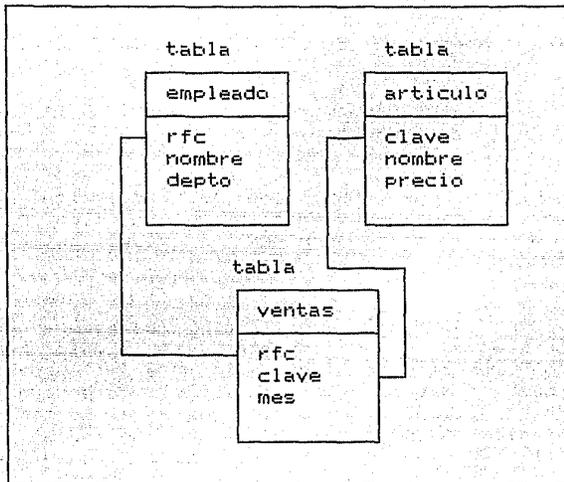
Lista los nombramientos (grupos) que tengan en promedio más de 4 hijos  
.

## GLOSARIO

**ATRIBUTO.** Unidad básica de una base de datos, ejemplos de atributos son el RFC; nombre; sueldo.

**BASE DE DATOS.** Dispositivo electrónico que almacena los datos. Existen tres modelos de bases de datos, relacional, jerárquico y de red (ver apéndice A). El usuario de bases de datos relacionales no tiene porque preocuparse de la representación física de la base, más bien conceptualiza a la base de datos compuesta por tablas y por relaciones entre tablas.

### BASE DE DATOS



**CODIFICACION DEL PROBLEMA.** Etapa última de la producción de interrogantes, en esta etapa el usuario expresa su consulta de datos en términos de la sintaxis de un lenguaje de interrogación.

**FORMULACION DEL PROBLEMA.** Primer etapa de la producción de interrogantes en la que el usuario define que datos desea consultar a la base de datos

**INTERFASE.** Inter: entre, fase: estado, dispositivo de interacción entre dos entidades, por ej. un sistema de cómputo y el usuario. Ejemplos de interfases sistema-usuario son: documentación del sistema, "ventanas", etc.

**INTERROGACION POR MENU.** Forma de diálogo entre el sistema y usuario, esta forma de interrogación es la más fácil de usar, y adecuada cuando se tienen bien identificadas las opciones de

## GLOSARIO

interrogación.

LENGUAJE DE INTERROGACION O INTERROGADOR A BASES DE DATOS. forma de diálogo entre el sistema y el usuario, ejemplos de interrogadores son: menú de opciones, lenguaje formal de interrogación y lenguaje natural de interrogación.

INTERROGANTE. estructura sintáctica predefinida de interrogación (programa), por ejemplo el SQL cuenta con los interrogantes de selección, proyección, ordenamiento, único, operaciones agregadas, agrupamiento, selección de grupos.

LEFI. LEnguaje Formal de Interrogación, este lenguaje esta compuesto por un vocabulario restringido y por un conjunto estructurado de reglas sintácticas. El SQL es un ejemplo de LEFI.

LENI. LEnguaje Natural de Interrogación, este lenguaje es muy parecido al lenguaje nativo del usuario.

PREPARACION DEL PLAN. Segunda etapa de la producción de interrogantes, en esta etapa el usuario traza un plan para obtener los datos que se definieron en la etapa anterior (formulación del problema).

PRODUCCION DE INTERROGANTES. Modelo que describe las etapas que ocurren en la elaboración de un interrogante, este modelo toma como entrada una consulta de datos (que es lo que el usuario desea obtener de la base de datos) y como salida el interrogante en un lenguaje de interrogación. Este modelo consta de las etapas siguientes: a) Formulación del problema, el usuario define en su lenguaje materno el tipo de datos que desea consultar, b) Preparación del plan, el usuario elige una estrategia para recuperar los datos y c) codificación del problema, el usuario expresa su consulta en términos de la sintáxis de un lenguaje de interrogación en particular. La relación de las etapas es secuencial, es decir el resultado de una etapa sirve como entrada a la subsecuente.

RATON. "mouse", dispositivo que se utiliza para controlar el los movimientos del cursor o para ejecutar cierta acción.

## GLOSARIO

REGISTRO. Renglón de la tabla, cada registro tiene un número fijo de atributos y hace corresponde a cierta ocurrencia de la tabla por ej. el empleado Luther Zepeda Belinda se representa con el siguiente registro:

rfc	depto	sueldo
LUZB621201	Ventas	400,000.00

SABAD. Sistema Administrador de Base de Datos, conjunto de lenguajes y utilerías diseñados para administrar una base de datos, es decir registrar, procesar y recuperar datos. El lenguaje de interrogación es un componente del sistema administrador de bases de datos y tiene particular importancia dado que es una herramienta muy utilizada por los usuarios del sistema.

SQL. Structured Query Language, lenguaje de interrogación estructurado a base de datos de tipo relacional.

TABLA. La tabla hace referencia a cierta entidad de la base de datos, por ejemplo al empleado. Una tabla puede conceptualizarse como un archivo, en donde las columnas corresponden a los atributos del empleado (R.F.C.) y los renglones a los registros (algún empleado).

tabla empleado

<u>atributo</u>	<u>atributo</u>	<u>atributo</u>	
rfc	depto	sueldo	
UISF501203	Almacén	500,000.00	-> registro
TOC0550703	Ventas	600,000.00	-> registro
AATJ621204	Almacén	300,000.00	-> registro
CAHG601215	Compras	600,000.00	-> registro
HUMM580812	Ventas	300,000.00	-> registro
LUZB621201	Ventas	400,000.00	-> registro

TAREA DE CONSTRUCCION. En esta tarea el sujeto tiene que elaborar un interrogante en SQL dada una pregunta en español, ej.

Pregunta en español:

¿Cuál es el nombre del empleado con el R.F.C.= VAHE621204 ?

SQL

```
select nombre
from empleado
where rfc= 'VAHE621204'/'
```

En esta tarea se registran los tiempos, aciertos y el tipo de errores. Para esta tarea se debe contar idealmente con una cámara de video para registrar tiempos y errores de los sujetos.

## GLOSARIO

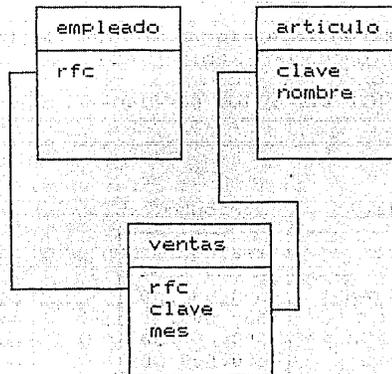
**TAREA DE VERIFICACION.** En esta tarea el sujeto decide el valor de verdad de un interrogante, por ej.

```
select nombre
from empleado
where rfc= 'VAHE621204' /
```

Respuesta [ V / F ]

Para esta tarea se requiere un programa que presente los interrogantes al azar para cada sujeto, registre los tiempos de verificación y las respuestas.

**VISTA.** Término que se utiliza para referir a la porción de la base de datos a la que tiene acceso el usuario. Es común que para cada usuario exista una vista en particular.



tablas= empleado, articulo, ventas  
atributos= rfc, nombre, clave, mes