



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

**ANÁLISIS DEL PROCESO ENSEÑANZA-APRENDIZAJE DEL MODELO  
ENTIDAD RELACIÓN COMO BASE PARA DISEÑAR BASES DE DATOS,  
DESARROLLADO EN LA ESCUELA DE INGENIERÍA EN CIENCIAS Y  
SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE  
SAN CARLOS DE GUATEMALA**

**Roberto Estuardo Ruiz Cruz**

Asesorado por la Inga. Floriza Ávila Pesquera de Medinilla

Guatemala, agosto de 2010







UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ANÁLISIS DEL PROCESO ENSEÑANZA-APRENDIZAJE DEL MODELO  
ENTIDAD RELACIÓN COMO BASE PARA DISEÑAR BASES DE DATOS,  
DESARROLLADO EN LA ESCUELA DE INGENIERÍA EN CIENCIAS Y  
SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE  
SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR:

**ROBERTO ESTUARDO RUIZ CRUZ**

ASESORADO POR LA INGA. FLORIZA ÁVILA PESQUERA DE MEDINILLA

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, AGOSTO DE 2010



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Maritza Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Luis Pedro Ortiz de León
VOCAL V	Agr. José Alfredo Ortiz Herincx
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Sydney Alexander Samuels Milson
EXAMINADOR	Ing. Rolando Haroldo Alonzo Ordoñez
EXAMINADOR	Ing. Byron Wosbely López López
EXAMINADOR	Ing. Otto Amilcar Rodríguez Acosta
SECRETARIO	Ing. Pedro Antonio Aguilar Polanco





**HONORABLE TRIBUNAL EXAMINADOR**

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**ANÁLISIS DEL PROCESO ENSEÑANZA-APRENDIZAJE DEL MODELO  
ENTIDAD RELACIÓN COMO BASE PARA DISEÑAR BASES DE DATOS,  
DESARROLLADO EN LA ESCUELA DE INGENIERÍA EN CIENCIAS Y  
SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE  
SAN CARLOS DE GUATEMALA,**

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, en marzo de 2008.

  
**Roberto Estuardo Ruiz Cruz**



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **ANÁLISIS DEL PROCESO ENSEÑANZA-APRENDIZAJE DEL MODELO ENTIDAD RELACIÓN COMO BASE PARA DISEÑAR BASES DE DATOS, DESARROLLADO EN LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por el estudiante universitario **Roberto Estuardo Ruiz Cruz**, autoriza la impresión del mismo.

IMPRÍMASE.

Ing. Murphy Olympo Paiz Recinos  
DECANO



Guatemala, agosto de 2010

/cc  
c.c. archivo.

E  
S  
C  
U  
E  
L  
A  
  
D  
E  
  
C  
I  
E  
N  
C  
I  
A  
S  
  
Y  
  
S  
I  
S  
T  
E  
M  
A  
S

UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado "ANÁLISIS DEL PROCESO ENSEÑANZA-APRENDIZAJE DEL MODELO ENTIDAD RELACIÓN COMO BASE PARA DISEÑAR BASES DE DATOS, DESARROLLADO EN LA ESCUELA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA", presentado por el estudiante ROBERTO ESTUARDO RUIZ CRUZ, aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAD A TODOS"

  
Ing. Marlon Antonio Pérez Turk  
Director, Escuela de Ingeniería en Ciencias y Sistemas



Guatemala, 20 de julio 2010



Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 07 de Junio de 2010


Ingeniero  
**Marlon Antonio Pérez Turk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación-EPS del estudiante **ROBERTO ESTUARDO RUIZ CRUZ** carné **85-11655**, titulado: **"ANÁLISIS DEL PROCESO ENSEÑANZA-APRENDIZAJE DEL MODELO ENTIDAD RELACIÓN COMO BASE PARA DISEÑAR BASES DE DATOS, DESARROLLADO EN LA ESCUELA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

  
**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación





UNIDAD DE E.P.S.

Guatemala, 27 de mayo de 2010.  
REF.EPS.D.415.05.10.

Ing. Marlon Antonio Pérez Turck  
Director Escuela de Ingeniería Ciencias y Sistemas  
Facultad de Ingeniería  
Presente

Estimado Ingeniero Perez Turck.

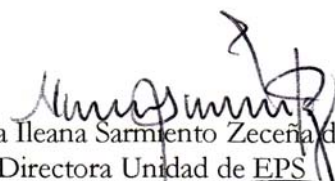
Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **“ANÁLISIS DEL PROCESO ENSEÑANZA-APRENDIZAJE DEL MODELO ENTIDAD RELACIÓN COMO BASE PARA DISEÑAR BASES DE DATOS, DESARROLLADO EN LA ESCUELA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, que fue desarrollado por el estudiante universitario **Roberto Estuardo Ruíz Cruz** Carné No. **8511655** quien fue debidamente asesorado y supervisado por la Inga. Floriza Felipa Ávila Pesquera de Medinilla

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte de la Asesora-Supervisora de EPS, en mi calidad de Directora apruebo su contenido solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

*“Id y Enseñad a Todos”*

  
Inga. Norma Ileana Sarmiento Zecena de Serrano  
Directora Unidad de EPS

NISZ/ra





UNIDAD DE E.P.S.

Guatemala, 27 de mayo de 2010.  
REF.EPS.DOC.677.05.10.

Inga. Norma Ileana Sarmiento Zeceña de Serrano  
Directora Unidad de EPS  
Facultad de Ingeniería  
Presente

Estimada Ingeniera Sarmiento Zeceña.

Por este medio atentamente le informo que como Asesora-Supervisora de la Práctica del Ejercicio Profesional Supervisado, (E.P.S) del estudiante universitario de la Carrera de Ingeniería en Ciencias y Sistemas, **Roberto Estuardo Ruíz Cruz** Carné No. **8511655** procedí a revisar el informe final, cuyo título es **“ANÁLISIS DEL PROCESO ENSEÑANZA-APRENDIZAJE DEL MODELO ENTIDAD RELACIÓN COMO BASE PARA DISEÑAR BASES DE DATOS, DESARROLLADO EN LA ESCUELA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**.

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

*“Id y Enseñad a Todos”*

Inga. Floriza Felipa Avila Pesquera de Medinilla  
Asesora-Supervisora de EPS  
Área de Ingeniería en Ciencias y Sistemas

FFAPdM/RA





## **ACTO QUE DEDICO A:**

### **DIOS**

Por permitirme llegar a este momento de mi vida y poder compartirlo con quienes me han apoyado y lo siguen haciendo hasta el día de hoy.

### **MI FAMILIA**

Por todo su amor y apoyo durante todas las etapas de mi vida, y sobre todo por el apoyo para lograr culminar esta carrera.

### **MIS AMIGOS**

Por cada una de las vivencias y aprendizajes que he podido compartir con cada una de las personas que Dios ha puesto en mi camino a lo largo de mi vida.

Roberto Estuardo Ruiz Cruz



# ÍNDICE GENERAL

<b>GLOSARIO</b> .....	<b>III</b>
<b>RESUMEN</b> .....	<b>VII</b>
<b>OBJETIVOS</b> .....	<b>IX</b>
<b>INTRODUCCIÓN</b> .....	<b>XI</b>
<b>1. MARCO TEÓRICO</b> .....	<b>1</b>
1.1 Antecedentes.....	1
1.2 Importancia del modelo de datos.....	2
1.3 Descripción del curso.....	5
<b>2. ORGANIZACIÓN Y COORDINACIÓN DEL CURSO</b> .....	<b>7</b>
2.1 Recurso humano .....	7
2.2 Horarios.....	7
2.3 Metodología de trabajo .....	8
2.3.1 Fase de servicio técnico profesional .....	8
2.3.2 Fase de investigación.....	9
2.3.3 Fase de docencia .....	9
2.4 Programa del curso.....	10
<b>3. MATERIAL DE APOYO DISEÑADO PARA EL CURSO</b> .....	<b>13</b>
3.1 Laboratorio.....	13
3.1.1 Proyectos .....	13
3.1.1.1 Enunciado del primer proyecto .....	13
3.1.1.2 Diagrama de solución del primer proyecto.....	17
3.1.1.3 Enunciado del segundo proyecto.....	18
3.1.1.4 Solución del segundo proyecto .....	22
3.1.2 Examen corto .....	33
3.2 Exámenes.....	35
3.2.1 Examen parcial, práctica de SQL.....	35
3.2.2 Examen final.....	36
3.2.2.1 Solución: Diagrama de Compañía de Buses.....	38
3.2.3 Examen de retrasada .....	39
3.3 Tareas.....	42

<b>4. DESARROLLO DEL CURSO.....</b>	<b>43</b>
4.1 Distribución de la calificación .....	43
4.2 Asistentes .....	44
4.2.1 Estadísticas .....	45
4.2.2 Casos especiales .....	47
<b>5. EVALUACIÓN DEL CURSO .....</b>	<b>49</b>
5.1 Resultados.....	49
5.2 Dificultades encontradas.....	50
5.3 Propuesta de mejoras.....	50
<b>CONCLUSIONES .....</b>	<b>53</b>
<b>RECOMENDACIONES .....</b>	<b>55</b>
<b>BIBLIOGRAFÍA.....</b>	<b>57</b>

## GLOSARIO

<b>Álgebra Relacional</b>	Se refiere al conjunto de operadores que toman relaciones como sus operandos y regresan una relación como su resultado.
<b>Base de Datos</b>	Es un conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna organización dada.
<b><i>Data Warehouse</i></b>	Sistema de información centralizado que contiene toda la información relevante para una organización, y que permite de una forma ágil y flexible la consulta de información. Está orientado a convertirse en la única fuente de información para todas las áreas del negocio y ser la fuente prioritaria para todas las herramientas de inteligencia de negocio que se desee utilizar en una organización.
<b>DDL</b>	Consiste de instrucciones declarativas que son necesarias para declarar objetos de base de datos (lenguaje de definición de datos).

**Diagrama de Entidad / Relación** Dibujo que intenta explicar el diseño lógico de una base de datos a un nivel de abstracción en el cual muchos detalles, particularmente de tipos y restricciones de integridad, son omitidos.

**Diseño de Base de Datos** Dado cierto cuerpo de datos que deben ser representados en una base de datos, son las decisiones que se toman sobre la estructura lógica conveniente para esos datos.

**DML** Consiste en aquellas instrucciones ejecutables que transfieren información hacia y desde la base de datos (lenguaje de manipulación de datos).

**Metadatos** Datos acerca de los datos. Son tablas del sistema que contienen información sobre la base de datos.

**Modelo de Entidad / Relación** Conjunto de convenciones para dibujar diagramas de entidad / relación

**Modelo Relacional** La teoría formal o fundamento sobre el cual las bases de datos relacionales en particular y la tecnología relacional en general están basadas.

<b>Normalización</b>	Proceso de descomposición sin pérdida de datos para reemplazar una relación o tabla por sus proyecciones, con el objetivo de reducir la redundancia.
<b>Integridad</b>	El término integridad se refiere a la exactitud o corrección de los datos en la base de datos.
<b>Redundancia</b>	Una base de datos dada despliega redundancia, sí y solo sí incluye dos o más representaciones distintas, ya sea directas o indirectas, para la misma proposición.
<b>Relación</b>	Es básicamente el término matemático para "tabla".
<b>SQL</b>	Su nombre proviene de la abreviatura (en inglés) de Lenguaje Estructurado de Consultas. Es el intento mejor conocido de la realización de las ideas abstractas del modelo relacional a una forma sintáctica concreta.



## RESUMEN

Los modelos de datos entendidos como el diseño de las estructuras de datos y la documentación de dicho diseño, constituyen una de las etapas críticas dentro de del desarrollo de un sistema de información computarizado. Dentro de este contexto es de suma importancia que el estudiante conozca, practique y adquiera experiencia en el uso del modelo de entidad relación para diseñar bases de datos.

El concepto de diagrama entidad relación es introducido en la carrera de Ingeniería en Ciencias y Sistemas en el curso de Introducción a la Programación y Computación 2, y es en el curso Sistemas de Bases de Datos 1 donde el tema se trata con mayor profundidad y debe constituirse en la base para que el estudiante adquiera el dominio completo de este instrumento de modelado tan importante en el diseño de sistemas de información.

Para hacer el análisis del proceso de enseñanza aprendizaje del modelo de entidad relación, se dio seguimiento durante el primer semestre de 2008 a todas las actividades relacionadas con el curso Sistemas de Bases de Datos 1, que de aquí en adelante podrá ser llamado como "el curso", en una de las dos secciones habilitadas para dicho semestre, en este caso, la sección B. Las actividades de seguimiento del curso van desde el inicio del semestre, hasta la entrega de notas finales del curso incluyendo los exámenes de primera y segunda retrasada.

El desarrollo del curso incluye la planificación y el desarrollo de clases magistrales, asignación de tareas e investigaciones, la elaboración de enunciados de proyectos y la elaboración de evaluaciones (exámenes cortos, exámenes parciales, examen final y exámenes de primera y segunda retrasada). Así como la calificación de cada uno de los elementos que forman parte de las notas del curso y laboratorio y el reporte de las mismas a Control Académico de la Facultad.

Se presenta este informe con el detalle del contenido del curso desarrollado en el primer semestre de 2008. Se incluye en el informe el programa del curso, la distribución de la nota del laboratorio y del curso en general, incluyendo también estadísticas de la población estudiantil por año de ingreso a la Universidad, promedios de notas de exámenes y proyectos.

Para cumplir con el objetivo del Ejercicio Profesional Supervisado, se dan conclusiones de cómo se está cumpliendo con los objetivos del curso en cuanto a la enseñanza-aprendizaje del diagrama de entidad-relación y recomendaciones de cómo puede enriquecerse y mejorarse el contenido del curso.



## **OBJETIVOS**

### **GENERAL:**

Preparar a los estudiantes con un buen nivel de conocimientos en el curso "Sistemas Administrativos de Bases de Datos I", para que en el futuro sean profesionales con un alto nivel de abstracción y conocimiento del modelo entidad-relación.

### **ESPECÍFICOS:**

1. Recomendar mejoras al contenido del curso, estas mejoras pueden ser dentro del programa oficial del curso, bibliografía, contenido del laboratorio y/o materiales de apoyo como tareas exámenes cortos, exámenes parciales y proyectos.
2. Motivar al estudiante de la carrera de Ingeniería en Ciencias y Sistemas, al hacer conciencia de las tendencias en informática y de cómo deben prepararse para su carrera profesional.



## INTRODUCCIÓN

La población estudiantil de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, que ha tenido un índice de crecimiento bastante considerable, lo cual ocasiona que se dificulte cubrir las plazas de docencia que se requieren, principalmente en los cursos del área profesional. Esto se debe a diversos factores, ya que quienes tienen experiencia en las áreas profesionales de la carrera de Ingeniería en Ciencias y Sistemas, muchas veces no tiene el tiempo o la disposición de dedicarse a la docencia, tomando en cuenta que se necesita dedicar el tiempo suficiente para investigar, preparar y desarrollar el material que se requiere para impartir un curso durante un semestre. Además, los horarios que se requieren para impartir una clase presencial, muchas veces provocan que se parta la jornada laboral para desarrollar un trabajo de tiempo completo en otra institución, en la que probablemente se podría obtener una mejor remuneración.

Debido a la importancia del Modelo de Entidad Relación, surge la necesidad de apoyar a la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, revisando el contenido impartido, los métodos de evaluación como exámenes parciales, exámenes cortos y tareas, y en general haciendo un análisis del proceso de enseñanza y demás elementos utilizados en el curso Sistemas de Bases de Datos 1.

Además, el proyecto se presenta como Ejercicio Profesional Supervisado EPS, como una alternativa al trabajo de graduación, pudiendo así optar al examen de graduación de la carrera mencionada.

El proyecto consiste en apoyar de forma directa e inmediata a la Escuela de Ingeniería en Ciencias y Sistemas, dando seguimiento a todas las actividades desde el inicio del semestre hasta la finalización del mismo, incluyendo la segunda oportunidad de retrasada para el curso Sistemas de Bases de Datos 1 durante el primer semestre del 2008, en la sección B. Se hace un análisis del contenido del curso, de los estudiantes que se asignaron el curso en la sección B, se describe la forma en que se desarrolló y al final se dan recomendaciones de cómo puede mejorarse el contenido del mismo.

De este modo se sienta un precedente, para que otros profesionales puedan apoyar a la carrera de la misma manera, con cursos que sean de su dominio e interés, y que además quieran cumplir con su Ejercicio Profesional Supervisado para optar a graduarse.

# 1. MARCO TEÓRICO

## 1.1 Antecedentes

El modelo Entidad Relación es una de las herramientas mejor conocidas para el diseño lógico de bases de datos. Chen (1976) introdujo la idea de los diagramas de entidad relación (Elsmani y Navathe, 2000), y la mayoría de los libros de texto de bases de datos utilizan alguna variante de dicho modelo. Chen y otros han mejorado el proceso de Entidad Relación a través de los años y actualmente no existe un estándar para el uso de modelos de diagramas de entidad relación, por lo que en los libros de textos y para fines didácticos se utilizan diferentes modelos. Algunos de los más utilizados son el modelo Chen, el modelo Barker y el modelo Barker/*Oracle*, además de las variantes de cada uno de ellos.

Una de las ventajas del modelo Chen es el hecho de que con este modelo no es necesario saber cómo va a ser implementada la base de datos. El modelo Barker está más íntimamente ligado al paradigma de las bases de datos relacionales. La corporación *Oracle* utiliza un diagrama de entidad relación que se asemeja más al modelo Barker. También cabe mencionar que en el diagrama entidad relación de Barker y de *Oracle*, no hay lugar para algunas características que si están presentes en el modelo Chen. Por ejemplo, los atributos multivaluados y las entidades débiles, no son parte del proceso de diseño de Barker ni de *Oracle*.

Es importante que el catedrático haga mención de los diferentes modelos y variantes existentes para realizar diagramas de entidad relación y luego ceñirse a uno de ellos para desarrollar el curso. Dada esta variación, debe hacerse más énfasis en la importancia de una buena documentación de la notación y de todos los elementos incluidos en el diagrama de entidad relación.

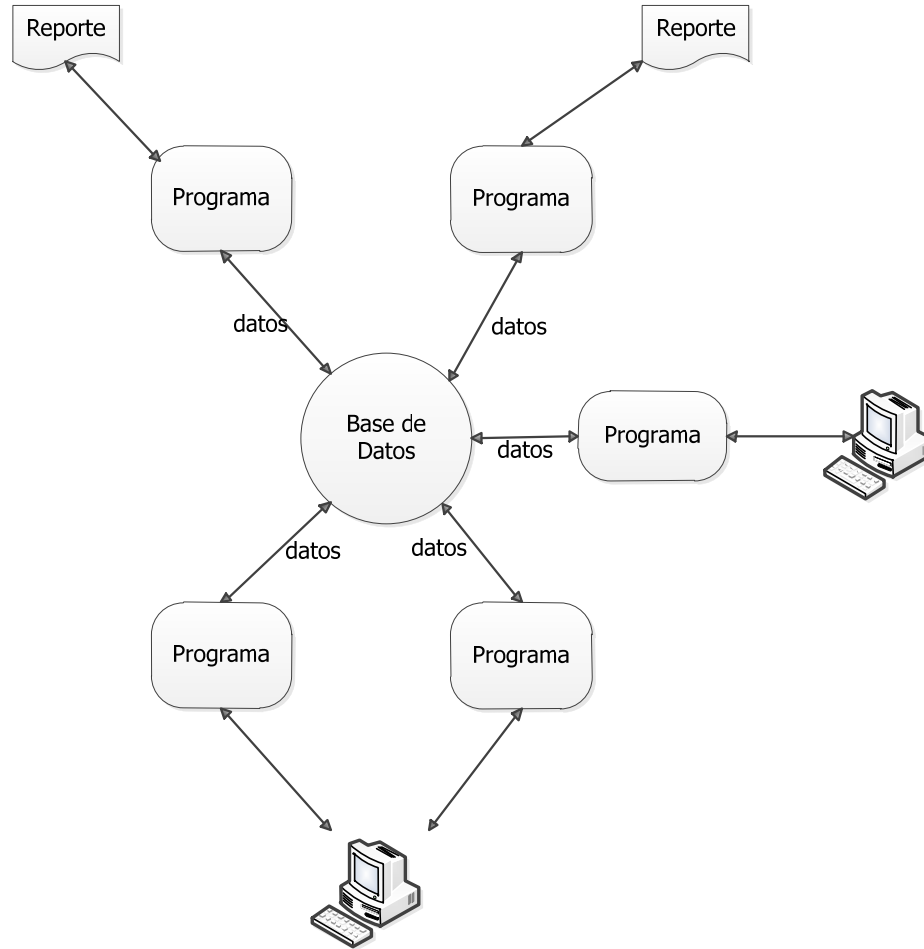
Como se mencionó anteriormente, los diagramas de entidad relación son introducidos en el curso Introducción a la Programación y Computación 2, pero es en el curso Sistemas de Bases de Datos 1 donde se cubre el tema con mayor profundidad, en las siguientes secciones se describe la importancia del modelo de datos dentro del diseño de bases de datos y en los siguientes capítulos se describe el análisis que se hizo por medio del seguimiento al desarrollo del curso durante el primer semestre de 2008.

## **1.2 Importancia del modelo de datos**

Se debe prestar mucha atención a la organización de los datos ya que estos constituyen la base de un sistema, un pequeño cambio al modelo de datos puede significar un cambio al sistema completo. En la mayoría de los sistemas de información comerciales, los programas son mucho más complejos y llevan mucho más tiempo en su especificación y construcción que la base de datos. Pero su contenido y estructura están fuertemente influenciados por el diseño de la base de datos.

En la siguiente figura puede observarse los componentes de un sistema de información.

## Un Sistema de Información



Se puede pensar en un sistema de información formado por una base de datos (conteniendo datos almacenados) junto con programas que capturan, almacenan, manipulan y recuperan esos datos.

Estos programas están diseñados para implementar un modelo de proceso (o especificación funcional), especificando los procesos del negocio que el sistema va a realizar. De la misma manera, la base de datos está especificada por un modelo de datos, describiendo qué tipo de datos va a almacenar y cómo van a estar estos datos organizados.

La mayoría de los programas van a estar interactuando con los datos dentro de la base de datos: almacenando, actualizando, eliminando, manipulando, imprimiendo y desplegando los datos. Su estructura, por lo tanto, necesitará reflejar la forma en que los datos están organizados.

El impacto de la organización de los datos en el diseño de programas tendrá consecuencias prácticas importantes:

- Un modelo de datos bien diseñado puede simplificar y bajar los costos de la programación. Incluso un cambio pequeño al modelo de datos puede llevar a ahorros significativos en el costo total de programación.



- Una organización de datos deficiente puede ser muy cara de corregir. Hacer las correcciones necesarias a un modelo de datos mal definido puede hacerse relativamente rápido. Con el software moderno de administración de bases de datos disponible, la base de datos probablemente puede ser reorganizada para reflejar el nuevo modelo, sin mucha dificultad. Pero el impacto real es en el resto del sistema. Formatos de reportes que necesitan ser rediseñados para mostrar nuevos datos, pantallas que se deben modificar para permitir el ingreso y despliegue de nuevos datos, programas que deben considerar nuevas condiciones de búsqueda, nuevas estructuras, nuevos ciclos, etc. Cambiar el diseño de la base de datos en sí misma puede ser bastante directo, pero los costos vienen de cambiar cada programa que usa la parte afectada. En contraste, arreglar un único programa incorrecto, incluso al punto de reescribirlo completamente, es una tarea (relativamente) simple.

### **1.3 Descripción del curso**

El curso "Sistemas de Bases de Datos 1" es un curso obligatorio de la carrera de Ingeniería en Ciencias y Sistemas, ubicado en el séptimo semestre del pénsum de estudios, corresponde al área de Desarrollo de Software y tiene asignados cinco créditos, su código es 774, es prerrequisito de los cursos "Sistemas de Bases de Datos 2" y "Análisis y Diseño de Sistemas 1"; es post-requisito del curso "Manejo e Implementación de Archivos".

En este curso se introducen los conceptos fundamentales de los sistemas de bases de datos relacionales y comprende tres grandes áreas: (1) el diseño de bases de datos, que incluye la comprensión de la información que manejan las organizaciones, su abstracción, clasificación y diseño en el modelo entidad/relación; (2) la capacidad de abstraer esta representación en términos de un modelo de estructura de datos, concretamente el modelo relacional, estudiando sus aspectos de estructura, integridad y manipulación; y, finalmente, (3) aprender a definir y manipular la información por medio de un lenguaje de programación de bases de datos, específicamente el Lenguaje Estructurado de Consultas (SQL).

## **2. ORGANIZACIÓN Y COORDINACIÓN DEL CURSO**

### **2.1 Recurso humano**

En este semestre, el curso Bases de Datos 1 fue dividido en dos secciones, A y B, las cuales fueron atendidas por un profesor titular, un profesor interino y dos auxiliares o ayudantes de cátedra. En algunos casos, el proyecto se definió conjuntamente para que fuera el mismo proyecto para las dos secciones, en otras ocasiones varió, definiéndolo cada profesor por separado para su sección.

### **2.2 Horarios**

El curso tiene asignados cuatro períodos a la semana, los cuales fueron distribuidos en el caso de la sección B, en dos períodos consecutivos en dos días distintos de la semana. El auxiliar llegó a un acuerdo con los estudiantes para el horario en que habrían de recibir dos períodos adicionales cada semana para cubrir el contenido del laboratorio.

Horario de la sección B durante el primer semestre del 2008:

<b>Día</b>	<b>Horario</b>	<b>Salón</b>
Miércoles	7:10 – 08:50	211, edificio T-3
Sábado	9:10 – 10:50	216, edificio T-3

## **2.3 Metodología de trabajo**

### **2.3.1 Fase de servicio técnico profesional**

En la fase de servicio técnico profesional se motiva y asesora al ayudante de cátedra y a los estudiantes en el uso de los programas comerciales de manejo de bases de datos más populares en Guatemala. De esta manera se introduce a los estudiantes a la tecnología que deben utilizar en su futura actividad laboral.

También se tiene la tarea de realizar evaluaciones como exámenes cortos, parciales y finales, así como proyectos en los cuales el estudiante ponga en práctica la teoría aprendida. Luego el catedrático o el ayudante de cátedra deben calificar estas evaluaciones y proyectos en un tiempo prudencial. Finalmente hacer saber a los estudiantes de la nota, y si el estudiante lo requiere dar revisión sobre la evaluación o proyecto.

Durante todo el semestre se da supervisión y apoyo al ayudante de cátedra y a los estudiantes para el desarrollo del laboratorio del curso y de los proyectos, resolviendo las dudas que pudieran surgir para la realización de los mismos y las herramientas a utilizar.

### **2.3.2 Fase de investigación**

Es importante preparar con suficiente anticipación los temas a impartir en cada clase, investigando además del libro de texto del curso, en otras fuentes de consulta disponibles, como libros, revistas y publicaciones especializadas en el tema, Internet, etc. De tal manera que la exposición sea lo suficientemente clara para los estudiantes, abundando en ejemplos cuando sea necesario y resolviendo todas las dudas que puedan plantear.

También es importante diseñar adecuadamente las evaluaciones y proyectos para que estos puedan medir el nivel de conocimiento adquirido por los estudiantes.

### **2.3.3 Fase de docencia**

Se debe impartir el curso por medio de clase magistral cumpliendo con los días y horarios establecidos, exponiendo los temas correspondientes a cada día de clase, según el programa, dejando espacio para resolver dudas durante y después de la clase, según las necesidades del estudiante.

Se promueven investigaciones y tareas para que los estudiantes puedan poner en práctica los conocimientos adquiridos.

Realizar y calificar evaluaciones parciales, dentro de las fechas establecidas en el calendario de actividades de la Facultad de Ingeniería, así como la publicación de las notas a control académico.

Los enunciados de los proyectos deben estar redactados de una manera clara y comprensible para el estudiante, además de que se entrega el enunciado del mismo por escrito a cada estudiante, puede publicarse en el sitio de la Universidad Virtual para que puedan consultarlo en cualquier momento.

## **2.4 Programa del curso**

### **I. DATOS GENERALES**

CURSO: SISTEMAS DE BASES DE DATOS 1  
PRERREQUISITOS: MANEJO E IMPLEMENTACIÓN DE ARCHIVOS  
SECCIÓN: B  
VIGENCIA: Primer semestre de 2008  
HORARIO: Miércoles 7:10-8:50 y sábado 9:10-10:50

### **II. DESCRIPCIÓN**

Este curso tiene tres grandes áreas: (1) el diseño de bases de datos, que incluye la comprensión de la información que manejan las organizaciones, su abstracción, clasificación y diseño en el modelo entidad/relación; (2) la capacidad de abstraer esta representación en términos de un modelo de estructura de datos, concretamente el modelo relacional, estudiando sus aspectos de estructura, integridad y manipulación; y, finalmente, (3) aprender a definir y manipular la información por medio de un lenguaje de programación de bases de datos, específicamente, el Lenguaje Estructurado de Consultas (SQL).

El programa está estructurado de manera de cubrir estas tres grandes áreas. Para facilitar el proceso de enseñanza-aprendizaje se harán evaluaciones teóricas y prácticas. Se realizarán dos proyectos, el primero

de diseño de una base de datos y el segundo consiste en 15 consultas en SQL. Para el primer proyecto utilizarán una herramienta de diseño y para el segundo un manejador de bases de datos relacional.

### III. CONTENIDO

- |     |                                                          |     |                                                      |
|-----|----------------------------------------------------------|-----|------------------------------------------------------|
| 1.  | INTRODUCCIÓN                                             | 3.3 | Mapeos entre los distintos niveles                   |
| 1.1 | Qué es una base de datos                                 | 3.4 | El Administrador de la Base de Datos (DBA)           |
| 1.2 | Qué es un sistema de bases de datos                      | 3.5 | EL Sistema Manejador de Bases de Datos (DBMS)        |
| 1.3 | Qué es un sistema administrador de bases de datos (DBMS) | 3.6 | El diccionario de datos (Catálogo)                   |
| 1.4 | Por qué utilizar una base de datos                       | 3.7 | La estructura global de un sistema de bases de datos |
| 1.5 | Independencia de los datos                               |     |                                                      |
| 2.  | EL MODELO ENTIDAD/RELACIÓN                               | 4.  | EL NIVEL INTERNO DE LA ARQUITECTURA                  |
| 2.1 | Los datos operacionales                                  | 4.1 | La visión genérica del acceso a la base de datos     |
| 2.2 | Entidades                                                | -   | Manejador de discos                                  |
| 2.3 | Atributos                                                | -   | Manejador de archivos                                |
| 2.4 | Relaciones                                               | -   | DBMS                                                 |
| 2.5 | Reglas de diseño del modelo entidad/relación             | 4.2 | Archivos y conjuntos de páginas                      |
| 2.6 | Ejemplos de diseño                                       | 4.3 | Métodos de acceso                                    |
|     |                                                          | 4.4 | Estructura general del sistema                       |
|     |                                                          | 4.5 | Medios de almacenamiento                             |
| 3.  | ARQUITECTURA DE UN SISTEMA DE BASES DE DATOS             | 5.  | EL MODELO RELACIONAL                                 |
| 3.1 | Los 3 niveles de abstracción de la arquitectura          | 5.1 | El modelo relacional                                 |
| 3.2 | Modelos de datos y sublenguajes de datos                 | 5.2 | Estructura de datos relacional                       |
|     |                                                          | 5.3 | Reglas de integridad relacionales                    |
|     |                                                          | 5.4 | Algebra relacional                                   |
|     |                                                          | 6.  | EL LENGUAJE ESTRUCTURADO DE CONSULTAS                |

- 6.1 El lenguaje SQL
- 6.2 Componentes de un DBMS relacional
- 6.3 Sublenguaje de definición de datos (DDL) relacional
- 6.4 Sublenguaje de manipulación de datos (DML) relacional
  - A. Consultas con proyecciones y restricciones
  - B. Consultas de reunión
  - C. Funciones de Agregados
  - D. Consultas especiales
  - E. Uso de subconsultas
  - F. Uso de subconsultas correlacionadas
  - G. Consultas con operaciones de conjuntos
  - H. Subconsultas con EXISTS
  - I. Consultas jerárquicas
  - J. Inserción
  - K. Actualización
  - L. Eliminación
  - M. Vistas

#### IV. BIBLIOGRAFÍA

1. Introducción a los sistemas de bases de datos. C. J. DATE. Addison Wesley. Séptima edición. 2001.
2. SELECT SQL...SQL The Relational Database Language. Larry Newcommer. Macmillan Publishing Co. 1992



## **3. MATERIAL DE APOYO DISEÑADO PARA EL CURSO**

### **3.1 Laboratorio**

#### **3.1.1 Proyectos**

##### **3.1.1.1 Enunciado del primer proyecto**

#### **Programa de Lealtad Multi-Empresarial (Cliente Frecuente)**

Realizar un diagrama de Entidad-Relación para una empresa que implementa un programa de lealtad multi-empresarial.

El proyecto es individual.

Fecha de entrega: sábado 19 de abril.

Es obligatorio que utilicen una herramienta (a su elección) para hacer el diagrama.

Entregar el diagrama impreso y documentación.

¿En qué consiste el programa de lealtad? El “programa de lealtad” denominado también como programa de “cliente frecuente” consiste en que las personas se identifican como miembros del programa a través de una tarjeta. El cliente compra un “*kit* de afiliación” que incluye la tarjeta y debe de llenar una boleta con sus datos personales: Número de Cédula, Nombre completo; Datos familiares\*: Dirección para recibir correspondencia y otros datos demográficos: Edad (fecha de nacimiento), Sexo, Estado Civil, Ocupación.

Si el cliente extravía su tarjeta o se le deteriora por el uso deberá comprar una nueva, se emite otra tarjeta con una numeración distinta pero sigue siendo el mismo cliente, lo que significa que un cliente podría tener asignada más de una tarjeta.

El objetivo del programa es fomentar la lealtad de los clientes hacia los comercios o empresas afiliadas al programa de cliente frecuente, captar nuevos clientes, hacer análisis de mercadeo y enviar promociones hacia segmentos de mercado específicos. Esto lo pueden hacer por medio de la información histórica de compras, asociada a sus datos personales y demográficos.

Cuando el cliente compra en uno de los comercios afiliados, se le pregunta ¿Posee tarjeta de cliente frecuente? De ser así, la tarjeta se pasa por un lector de banda magnética o pos (punto de venta) en el cual se registra el número de tarjeta, asociado al detalle de la compra que está realizando. A cambio de que el cliente comparta su información, se le "premia" porque con cada compra va acumulando puntos, las reglas de acumulación pueden variar en cada comercio: 4 puntos por cada Q 50.00 de compra, 3 puntos por cada galón de combustible, etc. Promociones especiales: dobles, triples puntos en fechas y horarios especiales, sólo en ciertas sucursales, etc. Al llegar a cierta cantidad de puntos acumulados, el cliente los puede canjear por artículos dentro de otros establecimientos conocidos como "Centros de Canje". Es decir que existen tablas de premios, ejemplo: Viaje a Cancún 10,000 puntos, Horno de Microondas 2,600 puntos, etc.

\*Para que el cliente pueda acceder más rápidamente a los premios, están permitidos los "grupos de acumulación familiar" en los cuales varios miembros de la familia, acumulan puntos en forma conjunta (se debe definir quién es el titular de la cuenta que es el único que está autorizado para canjear).

Para evitar la contingencia (tener reservada una cantidad de dinero demasiado grande para cubrir los premios que se puedan canjear en algún momento) se ha decidido que los puntos no tengan una duración indefinida, sino que tendrán una duración máxima de dos años, después de los cuales los puntos que no se hayan canjeado se vencen. La fecha de vencimiento es el 31 de diciembre de cada año.

Se debe determinar en cualquier momento cuántos puntos vigentes se han acumulado, se han canjeado y están disponibles en cada comercio afiliado por cada uno de los integrantes del grupo de acumulación familiar.

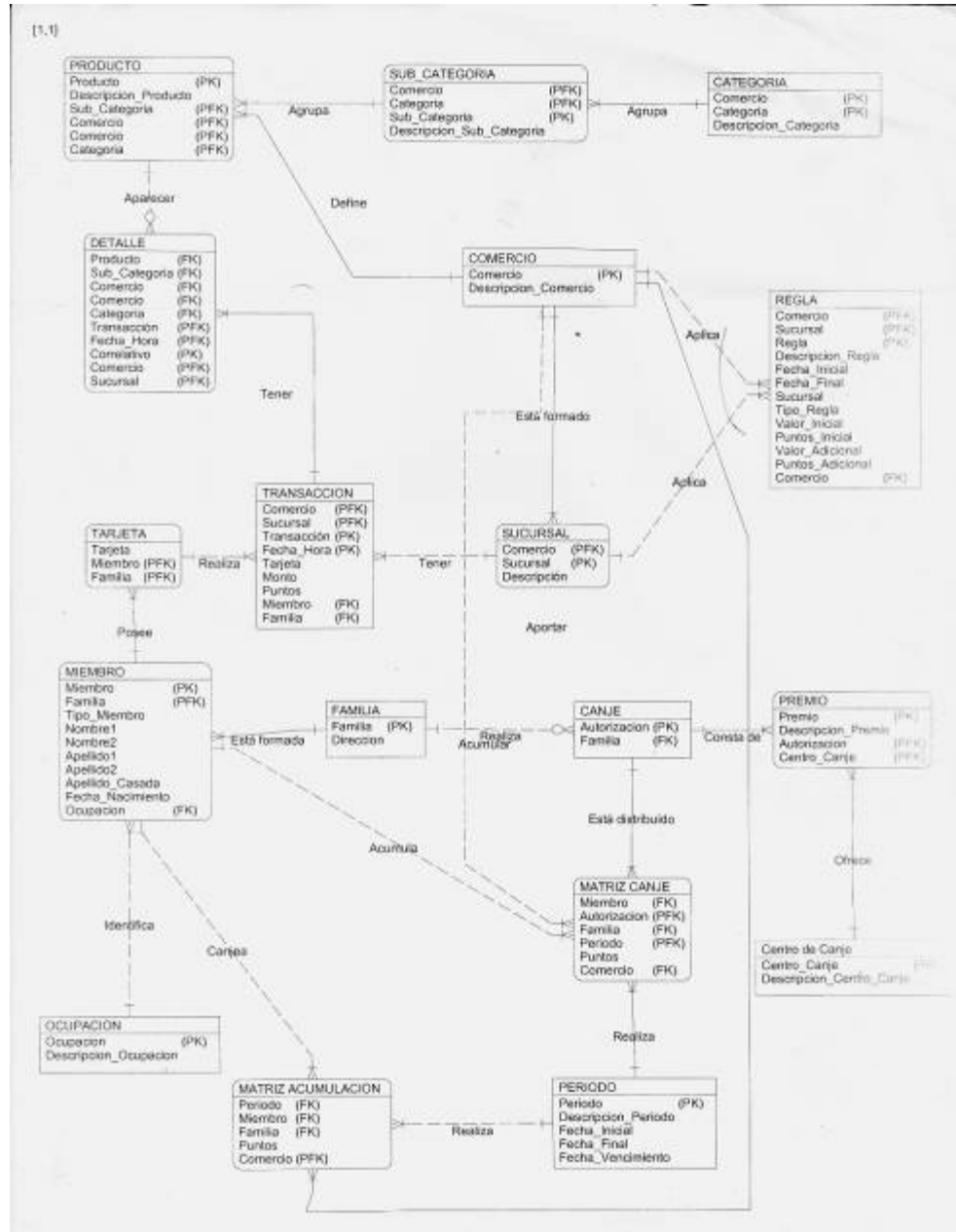
Se debe tener una estructura de hasta dos niveles de productos (categoría y subcategoría), para poder contestar preguntas del tipo:

Total de ventas del producto A-L-026 (ejemplo: litro de leche dos pinos)

Total de ventas de la subcategoría A-L (ejemplo: lácteos)

Total de ventas de la categoría A (ejemplo: abarrotes) en un período, en una sucursal, etc. O para ciertos segmentos: Amas de casa (ocupación), rango de edad, ubicación geográfica.

### 3.1.1.2 Diagrama de solución del primer proyecto



### 3.1.1.3 Enunciado del segundo proyecto

#### PROYECTO #2, Consultas en SQL

Fecha de entrega: 24-may-08

El proyecto es individual, pueden utilizar a su elección cualquiera de los siguientes manejadores de bases de datos que soportan el estándar ANSI de SQL: DB2, Oracle o SQL Server. Las consultas del proyecto involucran las tablas, EMP y DEPT. La tabla EMP es una tabla con 14-tuplas con campos numéricos, cadenas y fechas. La tabla DEPT tiene cuatro-tuplas y campos numéricos y cadena. Hay una relación de muchos a uno de DEPT a EMP. A continuación el contenido de las tablas:

**select \* from emp;**

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-1980	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-1981	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-1981	1250	500	30
7566	JONES	MANAGER	7839	02-APR-1981	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-1981	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-1981	2850		30
7782	CLARK	MANAGER	7839	09-JUN-1981	2450		10
7788	SCOTT	ANALYST	7566	09-DEC-1982	3000		20
7839	KING	PRESIDENT		17-NOV-1981	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-1981	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-1983	1100		20
7900	JAMES	CLERK	7698	03-DEC-1981	950		30
7902	FORD	ANALYST	7566	03-DEC-1981	3000		20
7934	MILLER	CLERK	7782	23-JAN-1982	1300		10

**select \* from dept;**

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Para tener derecho a que se califique el proyecto sobre 100% debe correr satisfactoriamente cuatro consultas que se elegirán al azar el día de entrega. Si tiene una, dos o tres consultas que no corran adecuadamente, la nota será sobre 75%, 50% ó 25% respectivamente.

#### **TAREAS A REALIZAR:**

- Documentación del proyecto que incluya un diseño de la base de datos, diagrama de entidad relación, documentación de las tablas y campos de la base de datos, todos los *scripts* utilizados desde la creación de la base de datos hasta la resolución de las consultas, resultado de cada consulta.
- Crear un esquema llamado RRHH (recursos humanos) en donde se crearán los objetos (tablas) de la base de datos. No está permitido crear tablas auxiliares, resúmenes ni vistas para resolver las consultas, únicamente se pueden utilizar vistas en línea.
- Crear las tablas con sus respectivas restricciones, llaves primarias y foráneas. En general todas las columnas a excepción de MGR y COMM deben de tener una restricción de modo que no permita nullos. Para las llaves foráneas, la acción referencial a utilizar será: *RESTRICT*.
- Poblar las tablas, es decir, crear un *script* que realice las instrucciones correspondientes de *INSERT* a las mismas.
- Generar las sentencias SQL para resolver cada una de las siguientes consultas:

#### ***METADATA***

1. Generar un listado con el nombre de la tabla (una única vez), el nombre de la columna, el tipo de dato y su posición dentro de la tabla, para cada una de las tablas del esquema RRHH.

2. Generar un listado de las restricciones (*CONSTRAINTS*) de la tabla EMP indicando: nombre de la tabla, nombre de la restricción, nombre de la columna y tipo de la restricción.

### **RANGOS, SELECCIÓN Y MANIPULACIÓN DE FECHAS**

3. Reportar el nombre, salario y fecha de contratación de los empleados que ganan menos que el empleado contratado inmediatamente después que él.
4. Obtener un listado de todos los empleados: Nombre del empleado, años completos cumplidos de trabajar en la empresa hasta el día de hoy y Antigüedad: Si se trata del empleado más antiguo, la columna indica "MAS ANTIGUO", de lo contrario especificar el número de años, meses y días de diferencia en relación al empleado más antiguo.

### **AGREGACIÓN, *QUERIES* JERÁRQUICOS y *DATA WAREHOUSE***

5. Listado de cada departamento en donde haya al menos un empleado asignado y una lista de los nombres de empleados de ese departamento separados por comas:

Número de Depto.   Empleados

6. Nombre de cada departamento, cantidad de empleados, cantidad de empleados con comisión, salario mínimo, salario máximo, total de salarios y salario promedio.
7. Listado de empleados ordenado por salario, indicando nombre del empleado, salario y total acumulado. Ejemplo:

<u>Ename</u>	<u>Sal</u>	<u>Total</u>
SMITH	800	800
JAMES	950	1750
...		



8. Generar un *ranking* de los empleados basándose en el salario que devengan, quien tenga el salario mayor será el número uno y así sucesivamente, en caso de salarios iguales “desempatar” por la fecha de contratación, es decir que el que haya sido contratado primero tendrá mejor ranking (número más bajo).
9. Poner el número de departamento en columnas, indicando para cada departamento el número de empleados que hay en el mismo.
10. Poner el nombre del puesto en columnas y listar en filas los empleados que están en ese puesto.
11. Generar una matriz que indique en una columna el nombre del departamento y a continuación varias columnas con los nombres de los puestos: Oficinistas, Administradores, Presidente, Analistas y Ventas. La matriz debe decir ‘SI’ o ‘NO’ indicando si el departamento posee empleado(s) con ese puesto.
12. Crear un reporte como una vista jerárquica de la tabla de empleados mostrando quiénes están bajo el mando de quién, empezando por el empleado KING.
13. Generar un histograma vertical que muestre por medio de asteriscos, la cantidad de empleados que hay en cada departamento. Ejemplo:

```

D10 D20 D30
--- --- ---
      *
      *   *
*     *   *
*     *   *

```

14. Nombre del departamento y Total de salario más comisión de los departamentos en donde no trabaje ningún analista.
15. Nombre del puesto y Total de salario más comisión, más un gran total, de quienes trabajan en DALLAS o en CHICAGO.

### 3.1.1.4 Solución del segundo proyecto

#### ***METADATA***

1. Generar un listado con el nombre de la tabla (una única vez), el nombre de la columna, el tipo de dato y su posición dentro de la tabla, para cada una de las tablas del esquema RRHH.

-- Consulta 1

```
select  
decode(lag(table_name)over(order by table_name), table_name,null,  
table_name) tabla,  
column_name columna,data_type tipo,column_id posicion  
from dba_tab_columns  
where owner='RRHH'  
and table_name in ('DEPT','EMP')  
order by table_name,column_id;
```

TABLA	COLUMNA	TIPO	POSICIÓN
DEPT	DEPTNO	NUMBER	1
	DNAME	VARCHAR2	2
	LOC	VARCHAR2	3
EMP	EMPNO	NUMBER	1
	ENAME	VARCHAR2	2
	JOB	VARCHAR2	3
	MGR	NUMBER	4
	HIREDATE	DATE	5
	SAL	NUMBER	6
	COMM	NUMBER	7
	DEPTNO	NUMBER	8

2. Generar un listado de las restricciones (*CONSTRAINTS*) de la tabla EMP indicando: nombre de la tabla, nombre de la restricción, nombre de la columna y tipo de la restricción.

```
-- Consulta 2
select
table_name,constraint_name,constraint_type,delete_rule
from user_constraints
where table_name='EMP';
```

TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE	DELETE_RULE
EMP	SYS_C002227	P	
EMP	FK_MGR	R	NO ACTION
EMP	FK_EMP_DEPT	R	NO ACTION

### RANGOS, SELECCIÓN Y MANIPULACIÓN DE FECHAS

- Reportar el nombre, salario y fecha de contratación de los empleados que ganan menos que el empleado contratado inmediatamente después que él.

```
-- Consulta 3
select ename nombre,sal salario,
hiredate fecha_contratacion
from (
select deptno, ename, hiredate,sal,
lead(sal)over(order by hiredate) next_sal
from emp
)
where sal < next_sal;
```

NOMBRE	SALARIO	FECHA_CONTRATACIÓN
SMITH	800	17/12/1980
WARD	1250	22/02/1981
MARTIN	1250	28/09/1981
JAMES	950	03/12/1981
MILLER	1300	23/01/1982

4. Obtener un listado de todos los empleados: Nombre del empleado, años completos cumplidos de trabajar en la empresa hasta el día de hoy y Antigüedad: Si se trata del empleado más antiguo, la columna indica "MAS ANTIGUO", de lo contrario especificar el número de años, meses y días de diferencia en relación al empleado más antiguo.

-- Consulta 4

```
select ename nombre,
trunc(months_between(sysdate,hiredate)/12) años,
case when hiredate=(select min(hiredate) from emp)
then 'MAS ANTIGUO'
else to_char(trunc((hiredate-
(select min(hiredate) from emp))/365))||
' año(s), '||
to_char(
trunc(
months_between(hiredate,
(select min(hiredate)
from emp))-12*trunc((hiredate-
(select min(hiredate) from emp))/365))||
' mese(s)'||' y '||
trunc(30*(
months_between(hiredate,
(select min(hiredate)
from emp))-
trunc(
months_between(hiredate,
(select min(hiredate)
from emp))))))||' día(s).'
end antigüedad
from emp;
```

NOMBRE	AÑOS	ANTIGUEDAD
JONES	27	0 año(s), 3 mese(s) y 15 día(s).
KING	26	0 año(s), 11 mese(s) y 0 día(s).
BLAKE	27	0 año(s), 4 mese(s) y 14 día(s).
CLARK	26	0 año(s), 5 mese(s) y 22 día(s).
TURNER	26	0 año(s), 8 mese(s) y 21 día(s).
MILLER	26	1 año(s), 1 mese(s) y 5 día(s).
ALLEN	27	0 año(s), 2 mese(s) y 2 día(s).
WARD	27	0 año(s), 2 mese(s) y 4 día(s).
MARTIN	26	0 año(s), 9 mese(s) y 10 día(s).
SCOTT	25	1 año(s), 11 mese(s) y 22 día(s).
ADAMS	25	2 año(s), 0 mese(s) y 25 día(s).
JAMES	26	0 año(s), 11 mese(s) y 16 día(s).
FORD	26	0 año(s), 11 mese(s) y 16 día(s).
SMITH	27	MAS ANTIGUO

### **AGREGACIÓN, *QUERIES* JERÁRQUICOS y *DATA WAREHOUSE***

- Listado de cada departamento en donde haya al menos un empleado asignado y una lista de los nombres de empleados de ese departamento separados por comas: Número de Depto Empleados

```
-- Consulta 5
select deptno,
       ltrim(sys_connect_by_path(ename,',',';',';')) empleados
from (
select deptno,
       ename,
       row_number() over
         (partition by deptno order by empno) rn,
       count(*) over
         (partition by deptno) cnt
from emp
)
where level = cnt
start with rn = 1
connect by prior deptno = deptno and prior rn = rn-1;
```

DEPTNO	EMPLEADOS
10	CLARK,KING,MILLER
20	SMITH,JONES,SCOTT,ADAMS,FORD
30	ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES

6. Nombre de cada departamento, cantidad de empleados, cantidad de empleados con comisión, salario mínimo, salario máximo, total de salarios y salario promedio.

```
-- Consulta 6
select (select dname from dept where dept.deptno=emp.deptno) depto,
       count(1) empleados,count(comm) con_comision,
       min(sal) min,max(sal) max,sum(sal) total,round(avg(sal),2) prom
from emp
group by deptno;
```

DEPTO	EMPLEADOS	CON_COMISIÓN	MÍN	MÁX	TOTAL	PROM
ACCOUNTING	3	0	1300	5000	8750	2916.67
RESEARCH	5	0	800	3000	10875	2175.00
SALES	6	4	950	2850	9400	1566.67

7. Listado de empleados ordenado por salario, indicando nombre del empleado, salario y total acumulado.

-- Consulta 7

```
select ename nombre, sal salario,
       sum(sal) over (order by sal,empno) as total_acumulado
from emp
order by 2;
```

NOMBRE	SALARIO	TOTAL_ACUMULADO
SMITH	800	800
JAMES	950	1750
ADAMS	1100	2850
WARD	1250	4100
MARTIN	1250	5350
MILLER	1300	6650
TURNER	1500	8150
ALLEN	1600	9750
CLARK	2450	12200
BLAKE	2850	15050
JONES	2975	18025
SCOTT	3000	21025
FORD	3000	24025
KING	5000	29025

8. Generar un ranking de los empleados basándose en el salario que devengan, quien tenga el salario mayor será el número uno y así sucesivamente, en caso de salarios iguales “desempatar” por la fecha de contratación, es decir que el que haya sido contratado primero tendrá mejor ranking (número más bajo).

```
-- Consulta 8
select ename,
       rank() over (order by sal desc,hiredate) as ranking
from emp
order by 2;
```

ENAME	RANKING
KING	1
FORD	2
SCOTT	3
JONES	4
BLAKE	5
CLARK	6
ALLEN	7
TURNER	8
MILLER	9
WARD	10
MARTIN	11
ADAMS	12
JAMES	13
SMITH	14

9. Poner el número de departamento en columnas, indicando para cada departamento el número de empleados que hay en el mismo.

```
-- Consulta 9
select sum(case when deptno=10 then 1 else 0 end) dept_10,
       sum(case when deptno=20 then 1 else 0 end) dept_20,
       sum(case when deptno=30 then 1 else 0 end) dept_30,
       sum(case when deptno=40 then 1 else 0 end) dept_40
from emp;
```

DEPT_10	DEPT_20	DEPT_30	DEPT_40
3	5	6	0



10. Poner el nombre del puesto en columnas y listar en filas los empleados que están en ese puesto.

```
-- Consulta 10
select max(case when job='CLERK'
              then ename else null end) as oficinistas,
       max(case when job='ANALYST'
              then ename else null end) as analistas,
       max(case when job='MANAGER'
              then ename else null end) as admins,
       max(case when job='PRESIDENT'
              then ename else null end) as presidentes,
       max(case when job='SALESMAN'
              then ename else null end) as vendedores
from (
select job,
       ename,
       row_number()over(partition by job order by ename) rn
from emp
) x
group by rn;
```

OFICINISTAS	ANALISTAS	ADMINS	PRESIDENTES	VENDEDORES
ADAMS	FORD	BLAKE	KING	ALLEN
JAMES	SCOTT	CLARK		MARTIN
MILLER		JONES		TURNER
SMITH				WARD

11. Generar una matriz que indique en una columna el nombre del departamento y a continuación varias columnas con los nombres de los puestos: Oficinistas, Administradores, Presidente, Analistas y Ventas. La matriz debe decir 'SI' o 'NO' indicando si el departamento posee empleado(s) con ese puesto.

```

-- Consulta 11
select (select dname from dept d where d.deptno=emp.deptno) depto,
       case when sum(case when job='CLERK'
                          then 1 else 0 end)=0
          then 'NO' else 'SI' end as oficinistas,
       case when sum(case when job='ANALYST'
                          then 1 else 0 end)=0
          then 'NO' else 'SI' end as analistas,
       case when sum(case when job='MANAGER'
                          then 1 else 0 end)=0
          then 'NO' else 'SI' end as admins,
       case when sum(case when job='PRESIDENT'
                          then 1 else 0 end)=0
          then 'NO' else 'SI' end as pres,
       case when sum(case when job='SALESMAN'
                          then 1 else 0 end)=0
          then 'NO' else 'SI' end as vendedor
from emp
group by deptno;

```

DEPTO	OFICINISTAS	ANALISTAS	ADMINS	PRES	VENDEDOR
ACCOUNTING	SI	NO	SI	SI	NO
RESEARCH	SI	SI	SI	NO	NO
SALES	SI	NO	SI	NO	SI

12. Crear un reporte como una vista jerárquica de la tabla de empleados mostrando quiénes están bajo el mando de quién, empezando por el empleado KING.

```

-- Consulta 12
select ltrim(sys_connect_by_path(ename,' - '), ' - ') arbol
from emp
start with mgr is null
connect by prior empno=mgr
order by 1;

```

## ÁRBOL

```

KING
KING - BLAKE
KING - BLAKE - ALLEN
KING - BLAKE - JAMES
KING - BLAKE - MARTIN
KING - BLAKE - TURNER
KING - BLAKE - WARD
KING - CLARK
KING - CLARK - MILLER
KING - JONES
KING - JONES - FORD
KING - JONES - FORD - SMITH
KING - JONES - SCOTT
KING - JONES - SCOTT - ADAMS
  
```

13. Generar un histograma vertical que muestre por medio de asteriscos, la cantidad de empleados que hay en cada departamento.

```

-- Consulta 13
select max(deptno_10) d10,
       max(deptno_20) d20,
       max(deptno_30) d30
  from (select row_number() over(partition by deptno order by empno) rn,
         case when deptno=10 then '*' else null end deptno_10,
         case when deptno=20 then '*' else null end deptno_20,
         case when deptno=30 then '*' else null end deptno_30
        from emp) x
 group by rn
 order by 1 desc, 2 desc, 3 desc;
  
```

D10	D20	D30
		*
	*	*
	*	*
*	*	*
*	*	*

*	*	*
---	---	---

14. Nombre del departamento y Total de salario más comisión de los departamentos en donde no trabaje ningún analista.

-- Consulta 14

```
select (select dname from dept d where d.deptno=emp.deptno) depto,
       sum(sal+coalesce(comm,0)) salario_y_comision
from emp
group by deptno
having sum(case when job='ANALYST' then 1 else 0 end)=0;
```

DEPTO	SALARIO_Y_COMISIÓN
ACCOUNTING	8750
SALES	11600

15. Nombre del puesto y Total de salario más comisión, más un gran total, de quienes trabajan en DALLAS o en CHICAGO.

-- Consulta 15

```
select case grouping(job)
       when 0 then job else 'GRAN TOTAL' end job,
       sum(sal+coalesce(comm,0)) salario_y_comision
from emp
where deptno in (select deptno from dept where loc in ('DALLAS','CHICAGO'))
group by rollup(job);
```

JOB	SALARIO_Y_COMISIÓN
ANALYST	6000
CLERK	2850
MANAGER	5825
SALESMAN	7800
GRAN TOTAL	22475

### 3.1.2 Examen corto

1. Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso
  - a. Tabla
  - b. Tupla
  - c. Base de Datos
  - d. Ninguna
2. Es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos
  - a. DBMS
  - b. Base de datos
  - c. Modelo E-R
  - d. Todas
3. Las entidades definidas en el modelo E-R en la base de datos se convierten en:
  - a. Tuplas
  - b. Relaciones
  - c. Tablas
  - d. *Stored Procedures*
4. ¿Cuál de las siguientes no es una relación que exista en el modelo E-R?
  - a. 1:1
  - b. 1:muchos
  - c. Muchos:muchos
  - d. Varios:1

5. Sean T1 y T2, tablas en una base de datos relacional, si la llave primaria de T1 es la llave extranjera en T2 y viceversa, ambas tablas tienen una relación de tipo:
  - a. Varios:Muchos
  - b. 1:Muchos
  - c. 1:1
6. En el diagrama E-R las entidades se definen:
  - a. Singular
  - b. Plural
  - c. Ninguna
7. Se refiere al grupo de instrucciones que permiten definir los componentes estructurales de la base de datos:
  - a. DML
  - b. SQL
  - c. DLL
8. Este objeto, en la base de datos permite realizar una acción antes o después de que suceda una instrucción DML
  - a. Procedimiento almacenado
  - b. *Trigger*
9. Es la abstracción de características de los privilegios de un grupo de usuarios
  - a. Permisos
  - b. Rol
  - c. Ninguno
  - d. *Login*
10. ¿Por qué deberíamos utilizar procedimientos almacenados?

## 3.2 Exámenes

### 3.2.1 Examen parcial, práctica de SQL

Dadas las siguientes tablas de la base de datos de un estudio cinematográfico:

PELICULA(id, título, año, punteo, votos, director)

ACTOR(id, nombre)

ELENCO(película\_id, actor\_id, ord)

Los campos id son números enteros, el año es de cuatro dígitos, el punteo es un promedio (los usuarios de Internet califican entre 1 y 10 por sus películas favoritas), votos es entero y podría ser nulo cuando la película aún no ha recibido ningún voto, ord es el orden de aparición de un actor (cuando es uno significa que es el protagonista o actor principal), director es un número entero, su nombre se encuentra en la tabla actor. Los demás campos son cadenas de caracteres.

1. Escribir el comando SQL para crear cada una de las tres tablas con las restricciones respectivas de los campos que considere que no deberían de aceptar nulos, así como restricciones de llave primaria y llaves foráneas.
2. Escriba el comando SQL para obtener el elenco (lista de actores) de la película 'Alien' en orden de aparición.
3. Escriba el comando SQL para obtener la lista de todos los campos de la tabla de películas, en las cuales el director sea el mismo que el actor principal.
4. Escriba el comando SQL para obtener la lista de películas en las que ha actuado 'Harrison Ford', pero no como actor principal.

5. Escriba el comando SQL para obtener la lista de películas y votos del año 2003, ordenado descendientemente por número de votos, asegurándose que los valores nulos de votos estén al final en el ordenamiento. El manejador que se está utilizando no dispone de las cláusulas *nulls first* y *nulls last*.
6. Escriba el comando SQL para obtener la lista de películas y año, de la serie 'Star Trek'.
7. Escriba el comando SQL para eliminar de la base de datos todas las películas de la década del 70 que no hayan obtenido ningún voto.
8. Escriba el comando SQL para sumarle un voto a la película identificada con el id 230.
9. Escriba el comando SQL para obtener la suma de punteos de las películas del año 1999 seguido de la suma de punteos de las películas del año 2000.
10. Escriba el comando SQL para obtener el gran total de la consulta anterior.
11. Escriba el comando SQL para obtener el número de id de 'George Lucas'.
12. Escriba el comando SQL para obtener un listado de las películas en las que participó 'George Lucas' y en una segunda columna indicar si participó como actor o como director.

### **3.2.2 Examen final**

PRIMERA PARTE, Diseño. 40 puntos



## 1. Compañía de Buses

Una compañía de buses posee un cierto número de buses. Cada bus está asignado a una ruta en particular, sin embargo algunas rutas podrían tener varios buses. Cada ruta pasa por un número de ciudades. Uno o más conductores están asignados a cada fase de una ruta, la cual corresponde a un trayecto a través de algunas o todas las ciudades de una ruta. Algunas de las ciudades tienen un garaje en donde los buses son estacionados y cada uno de los buses está identificado por un número de registro y puede llevar diferente número de pasajeros, ya que los vehículos varían en tamaño y pueden ser de uno o de dos pisos. Cada ruta está identificada por un número de ruta y se tiene información disponible sobre el número de pasajeros promedio transportado cada día por cada ruta. Los conductores tienen un número de empleado, nombre, dirección y en algunas ocasiones número de teléfono.

Identificar las entidades, relaciones y atributos.

Dibujar el diagrama de entidad-relación.

SEGUNDA PARTE, Teoría. 4 puntos c/u

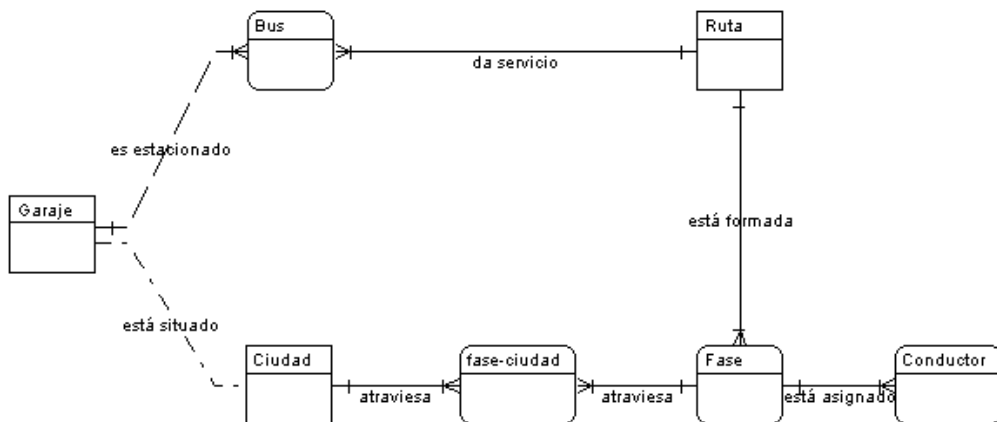
2. ¿Qué es un dominio?
3. Describa los tres niveles de arquitectura de un Sistema de Bases de Datos
4. Describa la independencia lógica de los datos
5. Describa la independencia física de los datos
6. ¿Qué componentes debería tener como mínimo un lenguaje de bases de datos?

TERCERA PARTE, SQL. 8 puntos c/u

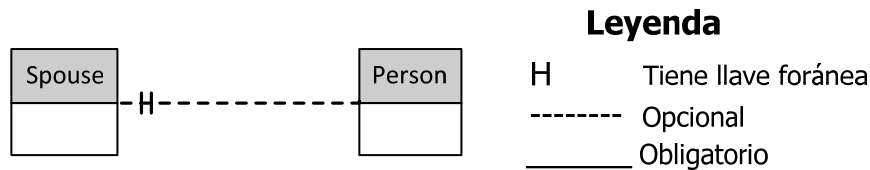
7. En la tabla ACTOR(id, nombre), el id es una secuencia que se genera automáticamente por medio de una aplicación y nombre es el nombre del actor. Por equivocación más de una persona trabajó con el mismo listado de actores ingresándolos. Cuando se dieron cuenta del error, ya solo una persona terminó de ingresar la lista, pero los primeros nombres quedaron repetidos (dos o tres veces). Escriba la instrucción SQL de *DELETE* para eliminar los datos de más.
8. Escriba el comando SQL para obtener la lista de nombres de quienes nunca hayan dirigido una película.
9. Escriba el comando SQL para obtener el nombre del director que haya obtenido más votos por año y la cantidad de votos, incluir también el año.
10. Escriba el comando SQL para insertar el registro de actor para Linda Carter con el número de id 100.

**3.2.2.1 Solución: Diagrama de Compañía de Buses**

[1.1]



### 3.2.3 Examen de retrasada



1. Según el diagrama anterior, cuál de las siguientes oraciones describe con mayor exactitud la relación mostrada:
  - a. Cada persona puede pertenecer a cero o más cónyuges
  - b. Cada persona puede tener cero o un cónyuge
  - c. Cada persona debe de tener uno y sólo un cónyuge
  - d. Cada cónyuge pertenece a cero o una persona
  - e. Cada cónyuge puede pertenecer a una o más personas

2. Considere el siguiente ejemplo desde el punto de vista de una compañía de tarjetas de crédito:

La compañía abre cuentas que pueden ser para un individuo o para una compañía, la cual puede emitir tarjetas para sus empleados. La compañía de tarjeta de crédito emite tres diferentes tipos de tarjeta, con diferentes límites, términos de pago y otras condiciones.

Pueden haber muchas tarjetas autorizadas, ya sean como cuentas personales o corporativas. Es importante saber quién posee cuál. Físicamente esto se logra embozando en la tarjeta, el nombre del tarjeta habiente, junto con el número de cuenta y la fecha de expiración.

Usted podría tener una cuenta, con una tarjeta propia y otra para su cónyuge. Su compañía, además puede tener una cuenta con una tarjeta para usted. ¡Cuidado con las confusiones cuando esté haciendo sus compras navideñas!

Su esposa podría incluso poseer una tercera cuenta, con tarjetas para ustedes dos y para sus hijos – estas tarjetas deberían tener límites bajos. ¿Una precaución sabia quizás? La Compañía de tarjetas de crédito necesita conocer quién posee cuentas, quién posee tarjetas y cuántas tarjetas de los diferentes tipos se poseen en cuentas personales o corporativas.

Identifique Entidades, Atributos y Relaciones y construya el diagrama de entidad relación.

3. Defina qué es un DBMS y un RDBMS
4. ¿Qué es un dominio?
5. ¿Qué es un atributo?
6. De una descripción del lenguaje SQL, qué es, para qué sirve y sus principales características
7. De una descripción de los sublenguajes DDL, DML y DCL que define el estándar ANSI de SQL
8. Para la tabla de empleados (EMP):

Orden\_Céd *char*(1),  
 Registro\_Céd *number*(12),  
 Código\_Depto *number*(5),  
 Número\_Emp *number*(5)

Se han creado por error registros con datos duplicados, con la excepción del número de empleado, escriba un *query* para eliminar los duplicados y que quede únicamente el número de empleado menor. Ejemplo, datos originales con duplicados:

ORDEN_CÉD	REGISTRO_CÉD	CÓDIGO_DEPTO	NÚMERO_EMP
A	12345	1	138
A	12345	1	487
A	12345	1	1486

Datos corregidos después de correr el *query*:

ORDEN_CÉD	REGISTRO_CÉD	CÓDIGO_DEPTO	NÚMERO_EMP
A	12345	1	138

9. ¿Cuántas condiciones se necesitan como mínimo en la cláusula *WHERE* para hacer un *JOIN* de N tablas sin que se produzca un producto cartesiano?
10. Escriba un *query* para obtener el total de salario (SAL) más comisión (COM) de cada departamento (a este total se le llamará presupuesto), ordenado en forma descendente por el monto del presupuesto para esta otra tabla de empleados (EMP):

EMPNO      *not null*      *number(5),*  
ENAME      *not null*      *varchar2(30),*  
SAL          *not null*      *number(12,2),*  
COM                      *number(12,2),*  
DEPTNO      *not null*      *number(5)*

Ponderación    Pregunta 1    10 pts  
                          Pregunta 2    30 pts  
                          Pregunta 3    10 pts  
                          Pregunta 4    05 pts  
                          Pregunta 5    05 pts  
                          Pregunta 6    08 pts  
                          Pregunta 7    08 pts  
                          Pregunta 8    08 pts  
                          Pregunta 9    08 pts  
                          Pregunta 10  08 pts

### 3.3 Tareas

Se asignaron varias tareas e investigaciones para complementar el aprendizaje del estudiante en el curso.

Tareas de investigación:

- 12 reglas de Codd
- Reglas de normalización

Tarea de práctica interactiva de SQL:

- Ejercicios de *join* en la página <http://sqlzoo.net>

## 4. DESARROLLO DEL CURSO

### 4.1 Distribución de la calificación

Durante este semestre, la calificación se distribuyó de la siguiente forma:

1 examen parcial teórico	06 puntos
1 examen parcial teórico práctico	18 puntos
1 examen parcial práctico de SQL	12 puntos
Tareas e investigaciones en clase	04 puntos
Laboratorio	
Proyecto de Diseño	09 puntos
Proyecto de Consultas SQL	19 puntos
Tareas y Cortos	07 puntos
TOTAL Laboratorio	<u>35 puntos</u>
ZONA	75 puntos
Examen final	25 puntos
TOTAL CURSO	100 puntos

Para tener derecho a someterse a examen final, se debe tener aprobado el laboratorio con un mínimo de 61%. La nota del laboratorio tiene validez por dos años, de no aprobar el curso dentro de ese período, se deberá cursar y aprobar nuevamente el laboratorio.

Los proyectos fueron diseñados por el catedrático titular de tal manera que se ponga en práctica y se evalúe el conocimiento adquirido durante el curso. Debido a que los mismos constituyen la parte práctica del curso, fueron incluidos dentro de la nota del laboratorio y fueron calificados por el auxiliar del curso, siempre bajo la supervisión del catedrático titular.

## 4.2 Asistentes

En la sección B del curso Bases de Datos 1, se asignaron 36 personas. Como se mencionó en el capítulo dos, este curso corresponde al séptimo semestre del pénsum de estudios, por lo que los estudiantes que deberían estarlo cursando en el presente semestre (primer semestre del 2008), son quienes ingresaron a la universidad en el año 2005. Sin embargo, de los 36 estudiantes asignados al curso, únicamente 13 de ellos (36%) ingresaron a la universidad en el año 2005.

A continuación se muestra un cuadro de cómo está formada la población estudiantil asignada en el curso, según su año de ingreso:

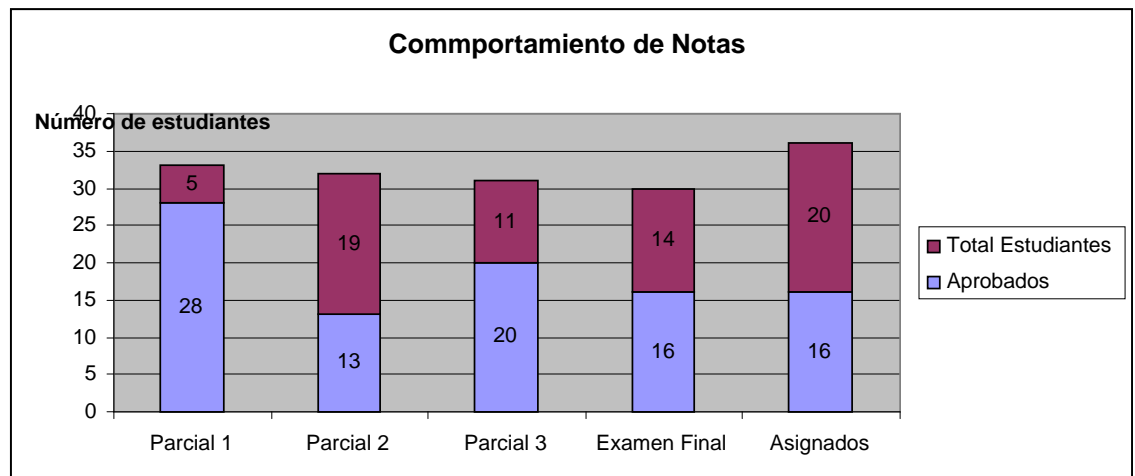
<b>AÑO</b>	<b>ESTUDIANTES</b>
2005 (En tiempo)	13
2004 (Atrasado 1 año)	5
2003 (Atrasado 2 años)	6
2002 (Atrasado 3 años)	1
2001 (Atrasado 4 años)	2
2000 (Atrasado 5 años)	3
Antes de 2000 (Atrasado más de 5 años)	6



En la siguiente sección de este capítulo se muestran más datos estadísticos acerca de los estudiantes que asistieron al curso y al laboratorio.

#### 4.2.1 Estadísticas

De los 36 alumnos que se asignaron el curso en el semestre, 16 aprobaron luego del examen final. La siguiente gráfica muestra la proporción de los estudiantes que se aprobaron (primera parte de la barra) y los que se presentaron (barra acumulada) a cada una de las evaluaciones y de la asignación del curso en general.



Alumnos que repiten el curso o que se atrasaron en curso(s) anterior(es) en la carrera y están cursando Sistemas de Bases de Datos 1 después de su séptimo semestre: 23 de 36 (64%).

Sin tomar en cuenta a los estudiantes que están al día en su p<sup>é</sup>nsu<sup>m</sup> de estudios, es decir, de los estudiantes que se inscribieron antes del 2005, hay cinco de 23 estudiantes (23%) que ya tienen aprobado el laboratorio y est<sup>á</sup>n revalidando el mismo. En el siguiente cuadro se muestra la estadística de revalidación de laboratorio seg<sup>ú</sup>n el a<sup>ñ</sup>o de ingreso a la universidad:

<b>AÑO</b>	<b>% de Estudiantes que revalidan laboratorio</b>
2004 (Atrasado 1 a <sup>ñ</sup> o)	20%
2003 (Atrasado 2 a <sup>ñ</sup> os)	17%
2002 (Atrasado 3 a <sup>ñ</sup> os)	100%
2001 (Atrasado 4 a <sup>ñ</sup> os)	0%
2000 (Atrasado 5 a <sup>ñ</sup> os)	0%
Antes de 2000 (Atrasado m <sup>á</sup> s de 5 a <sup>ñ</sup> os)	33%

Porcentaje de alumnos que aprobaron laboratorio por a<sup>ñ</sup>o de ingreso:

<b>AÑO</b>	<b>% de Estudiantes que aprueban laboratorio</b>
2005 (En tiempo)	85%
2004 (Atrasado 1 a <sup>ñ</sup> o)	50%
2003 (Atrasado 2 a <sup>ñ</sup> os)	20%
2002 (Atrasado 3 a <sup>ñ</sup> os)	No aplica
2001 (Atrasado 4 a <sup>ñ</sup> os)	0%
2000 (Atrasado 5 a <sup>ñ</sup> os)	33%
Antes de 2000 (Atrasado m <sup>á</sup> s de 5 a <sup>ñ</sup> os)	75%

Porcentaje de alumnos que aprobaron el curso por año de ingreso:

<b>AÑO</b>	<b>% de Estudiantes que aprueban curso</b>
2005 (En tiempo)	85%
2004 (Atrasado 1 año)	60%
2003 (Atrasado 2 años)	33%
2002 (Atrasado 3 años)	100%
2001 (Atrasado 4 años)	0%
2000 (Atrasado 5 años)	33%
Antes de 2000 (Atrasado más de 5 años)	50%

#### **4.2.2 Casos especiales**

Inicialmente se pensaba dividir las dos secciones del curso en A- y A+, lo cual significa que la asignación o distribución en cada sección es automática (números de carné impares en la sección A- y números de carné pares en la sección A+). Posteriormente y cuando se hizo la asignación oficial en control académico, las secciones quedaron como A y B, lo que significa que el estudiante tenía libertad de elegir la sección por preferencia, conveniencia de horarios, etc.

Hubo cinco estudiantes que ya tenían aprobada la nota del laboratorio con anterioridad y todavía se encontraba dentro del período de vigencia de dos años, para quienes se les aplicó el proceso de revalidación usual que se sigue en la Facultad. Esto es, en el momento de ingresar las notas finales en control académico, donde aparece la nota vigente del laboratorio y se le da la ponderación que se está usando actualmente (35%) para sumarlo a la nota del estudiante.

El otro caso especial fue el de un estudiante no asignado, esto se debió a que ya se había asignado el curso con anterioridad en tres ocasiones, por lo que el sistema no le permitió asignárselo nuevamente. Luego de que el estudiante completó los requisitos del curso, incluyendo la aprobación del laboratorio y la nota final del curso con 77 puntos, se adjuntó al cuadro de notas finales, una carta informando de dicha situación para autorizar la asignación y aprobación del curso.

## **5. EVALUACIÓN DEL CURSO**

### **5.1 Resultados**

Se cumplió con el análisis del curso como requisito para el EPS académico, dando seguimiento a todos los pasos desde la planificación del curso, desarrollo de clases magistrales, elaboración de material de apoyo, exámenes, proyectos y tareas.

Los contenidos del curso y los proyectos que se propusieron para el semestre apoyan directamente a los objetivos del curso, ya que permiten evaluar el conocimiento adquirido por los estudiantes para realizar un buen diseño a través del modelo relacional.

Se atendieron 36 estudiantes, brindándoles el apoyo para que adquieran el conocimiento teórico y práctico relacionado con el curso de Bases de Datos 1, resolviendo sus dudas y haciendo entrega de cuadros de zonas, notas de exámenes finales y de primera y segunda retrasada en los tiempos y plazos estipulados por la Facultad de Ingeniería.

Además pudo hacerse un análisis de los factores que influyeron, positiva o negativamente, en el desarrollo del curso dentro del semestre y así poder sacar conclusiones y dar recomendaciones sobre cómo puede enriquecerse y mejorarse el contenido del mismo.

## 5.2 Dificultades encontradas

La principal dificultad encontrada para un desarrollo normal y adecuado del curso fueron los retrasos ocasionados por la suspensión de clases, en actividades programadas, como la huelga de dolores, así como no programadas, como se dio en una oportunidad en la que por protestas estudiantiles cerraron algunos edificios y no fue posible impartir la clase.

## 5.3 Propuesta de mejoras

Para hacer mejoras al curso debe hacerse una revisión y actualización periódica del contenido del mismo, hay que tomar en cuenta que la tecnología evoluciona constantemente y al contrario de limitar a que se utilicen mejoras a las herramientas de los manejadores de bases de datos, se debe incentivar a que se conozcan, utilicen y se aprovechen dichas mejoras. Un ejemplo de esto son las funciones de ventana de SQL, citando el libro *SQL Cookbook* de Anthony Molinaro (libro 3 de la bibliografía): “El uso de funciones de ventana, sin duda el cambio más significativo de SQL de la última década (¡se está perdiendo de mucho si aún no hace uso de ella!)”

Se debe planificar cuidadosamente el desarrollo del curso desde el inicio del semestre tomando en cuenta el calendario de actividades de la facultad de ingeniería, de modo que no se den los atrasos que ocasionalmente suceden, cuando se corre la fecha de entrega de proyectos más allá de la fecha de finalización del semestre, interfiriendo con el desarrollo normal de actividades del curso de vacaciones y asignación de cursos del siguiente semestre.

También es importante la planificación de las evaluaciones parciales y proyectos para que se evalúe objetivamente el nivel de conocimientos adquiridos por los estudiantes.





## CONCLUSIONES

1. La teoría relacional continúa vigente en la actualidad, siendo ampliamente utilizada en una gran variedad de software de aplicaciones de bases de datos comerciales.
2. Es importante que los catedráticos tengan experiencia y estén actualizados en las tendencias tecnológicas más recientes para dar un enfoque adecuado al curso.
3. El contenido del curso para los programas vigentes es bastante extenso para poderse cubrir en un semestre y debe hacerse una buena planificación que permita presentar todos los temas de la mejor manera posible.
4. En muchas ocasiones se da el caso de que las fechas de entrega de proyectos se extienden más allá de la fecha oficial de finalización del semestre, lo que suele provocar problemas en el cumplimiento con las fechas establecidas por la Facultad para realizar exámenes finales, de retrasada y entrega de notas, como consecuencia el estudiante también puede tener problemas con su asignación de cursos del siguiente semestre.
5. Como en la mayoría de los cursos del área profesional de la carrera de Ingeniería en Ciencias y Sistemas, para este curso, el conocimiento práctico es un componente muy importante del mismo, por lo que el laboratorio y los proyectos deben contar también con una buena planificación.

6. El que ha sido el libro de texto "oficial" del curso (libro 1 de bibliografía) es un texto bastante completo para cubrir la mayor parte del contenido del curso, ofrece una base sólida de los fundamentos de la tecnología de bases de datos.
7. El contenido del curso ha tendido a ser muy variable dependiendo del catedrático o catedráticos designados para impartir el mismo en las distintas secciones.
8. No existe un documento oficial para el contenido a cubrirse en el laboratorio, sino que se define cada semestre de común acuerdo entre el auxiliar y el catedrático del curso.

## RECOMENDACIONES

1. Revisar y actualizar periódicamente, cada año o cada semestre, el contenido del curso, mismo que debería de ser aprobado por la dirección de la carrera de Ingeniería en Ciencias y Sistemas para que esté unificado y no dependa de la persona que se designe para desempeñarse como catedrático del curso.
2. A pesar de que el texto oficial del curso es bastante completo, se recomienda, principalmente para la parte de SQL, utilizar un libro especializado en el tema que esté actualizado, ya que el texto que aparece en la bibliografía para cubrir el tema de SQL es un libro de 1992 y desde ese año para la fecha, ha habido mejoras y actualizaciones al estándar SQL: ANSI SQL, SQL2, SQL3. En esta ocasión se utilizó el libro 3 de la bibliografía "SQL Cookbok, de Anthony Molinaro, año 2005".
3. De la misma manera, el contenido del laboratorio debe revisarse y crear un documento con el contenido oficial del mismo, apegado a los objetivos definidos para el curso.
4. Se debe hacer una planificación adecuada al principio del semestre tomando en cuenta los contenidos a cubrir, calendario establecido por la facultad de Ingeniería, para cumplir con las fechas establecidas, impartiendo el curso y evaluando apropiadamente todos los contenidos.



## BIBLIOGRAFÍA

1. C. J. Date. "Introducción a los Sistemas de Bases de Datos". Editorial Pearson Educación de México, S.A. de C.V., Séptima edición. 2001.
2. C. J. Date. "*The Relational Database Dictionary*". Editorial O'Reilly. 2006.
3. Molinaro, Anthony. "*SQL Cookbook*". Editorial O'Reilly. 2005.
4. Simsion, Graeme C.; Witt, Graham C. "*Data Modeling Essentials*". Morgan Kaufmann Publishers. Tercera edición. 2003.
5. Feuerstein, Steven; Pribyl, Bill; Dawes, Chip. "*Oracle PL/SQL Language Pocket Reference*". Editorial O'Reilly. Cuarta edición. 2007.
6. Kline, Kevin ; Kline, Daniel. "*SQL in a Nutshell*". Editorial O'Reilly. 2001.
7. Thompson, Kevin; Dicken, Deana. "*Learn SQL in a Weekend*". Editorial Premier Press. 2002.
8. Sperley, Eric. "*The Enterprise Data Warehouse: Planning, Building, and Implementation*". Prentice Hall. 1999.
9. Hobs, Lillian; Smith, Pete. "*Oracle Database 10g DataWarehousing*". Editorial Elsevier Digital Press. Oracle Corporation. 2004.

10. Halpin, Terry. *"Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design"*. Morgan Kaufmann Publishers. 2001.
11. Bagui, Sikha; Earp, Richard. *"Database Design Using Entity-Relationship Diagrams"*. Auerbach Publications. 2003.
12. Celko, Joe. *"Joe Celko's SQL for Smarties: Advanced SQL programming"*. Morgan Kaufmann Publishers. Tercera edición. 2005.
13. Powell, Gain. *"Oracle High Performance Tuning for 9i and 10g"*. Digital Press. 2004.
14. R., Elmasri; S.B., Navathe. *"Fundamentals of Database Systems"*. Addison Wesley. Tercera edición. 2000.