

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

REDES COMPUTACIONALES

## Práctica 1: Archivos de Ordenamiento

Grupo: 3CM3

Equipo: CompilandoConocimiento.com

*Integrantes:*

Morales López Laura Andrea

*Profesora:*

Nidia Cortez

Práctica n: Título

3CM3  
ESCOM-IPN

14 de junio de 2018

# Índice general

<b>1. Teoría</b>	<b>3</b>
1.1. Definiciones . . . . .	3
1.1.1. Dato . . . . .	3
1.1.2. Información . . . . .	3
1.1.3. Sistema . . . . .	3
1.1.4. Software . . . . .	3
1.1.5. Base de datos . . . . .	3
1.2. SGGD . . . . .	4
1.2.1. Motor Evaluador de consultas . . . . .	4
1.2.2. Gestión de almacenamiento . . . . .	4
1.2.3. Módulo de control de concurrencia ACID . . . . .	5
1.2.4. Administración de tolerancia a fallos . . . . .	5
1.3. Clasificación de los SGBD . . . . .	6
1.3.1. Modelo de datos . . . . .	6
1.3.2. Número de Usuarios . . . . .	7
1.3.3. Proposito . . . . .	7
1.3.4. Costos . . . . .	7
1.3.5. Número de sitios . . . . .	7
1.4. Clasificación de Modelo de Datos . . . . .	7
1.5. Elementos de un Modelo de Datos . . . . .	8
1.5.1. Arquitectura de un SGBD . . . . .	8
1.5.2. SQL Lenguaje Estructurado de Consultas . . . . .	8
1.5.3. Personas involucradas en un SBD . . . . .	9

1.6. Normalización . . . . .	10
1.6.1. Que es? . . . . .	10
1.6.2. Descripcion . . . . .	10
1.6.3. Dependencias Funcionales . . . . .	10
1.6.4. Forma Normal . . . . .	10
<b>2. SQL</b>	<b>12</b>
2.1. Introducción . . . . .	12
2.2. Aldunas cosas a tener en cuenta . . . . .	12
2.3. Algunas sentencias . . . . .	12
2.3.1. Crear un respaldo de la BD . . . . .	14
2.3.2. Crear un repado por un puerto diferente . . . . .	14
2.3.3. Crear una bd a partir de un respaldo . . . . .	14
<b>Appendices</b>	<b>15</b>

# Capítulo 1

## Teoría

### 1.1. Definiciones

#### 1.1.1. Dato

Representación gráfica con significado asignado.

#### 1.1.2. Información

Conjunto de datos ordenado.

#### 1.1.3. Sistema

Conjunto de elementos que se asocian para lograr un objetivo dentro de un entorno de cierto alcance.

#### 1.1.4. Software

Conjunto de programas con objetivo y estructura.

#### 1.1.5. Base de datos

- Representa aspectos del mundo real, a esto se le llama Mini Universo.
- Datos almacenados que tienen un significado interesante.
- Diseñar, manipular y construir.

## 1.2. SGGD

Es un programa para crear o manipular bases de datos. Algunos ejemplos son: MySQL, MariaDB, Server, Oracle, Informix, DB2, Sybase, MiniSQL, SQLite, Postgress, etc.

### 1.2.1. Motor Evaluador de consultas

#### El compilador

- Lexico
- Sintáctico
- Semántico (Congruencia)

#### Optimizador de consultas

- Basada en costos:
  - Para número de reuniones pequeños es aceptable, para números grandes es difícil de manipular.
  - Se pueden manipular en subconjuntos y calcular individualmente el mejor orden de reunión eliminando los más costosos de cada subconjunto.  
Usando esta técnica se puede implementar el algoritmo de programación dinámica para la optimización del orden de reunión óptimo.
  - Una desventaja es el costo de la misma optimización. El número de los planes de evaluación distintos para una consulta puede ser grande y encontrar el plan óptimo lleva mucho trabajo de cómputo.
- Heurística:
  - Suele a ayudar a reducir costo.
  - Realizar las operaciones de selección tan pronto como sea posible"
  - Se dice que la regla anterior es heurística porque ayuda a reducir el costo, aunque no lo haga siempre.
  - Realizar las proyecciones tan pronto como sea posible"

### 1.2.2. Gestión de almacenamiento

Depende del sistema operativo.

- Archivos y métodos de acceso (Búsqueda)
- Administrador de página (Gestiona Caché y RAM)
- Administrador de espacio en disco (RW)

### 1.2.3. Módulo de control de concurrencia ACID

- *A* tomicidad: Se ejecuta todo o nada.
- *C* onsistencia: Es el estado coherente de la información o datos que contiene y que se relacionan, en el cual la información cumple las necesidades o expectativas de quien la requiera.
- *I* Aislamiento: Ejecutar transacciones independientes.
- *D* urabilidad: Reflejar cambios de la transacción realizada.

#### Administrador de almacenamiento

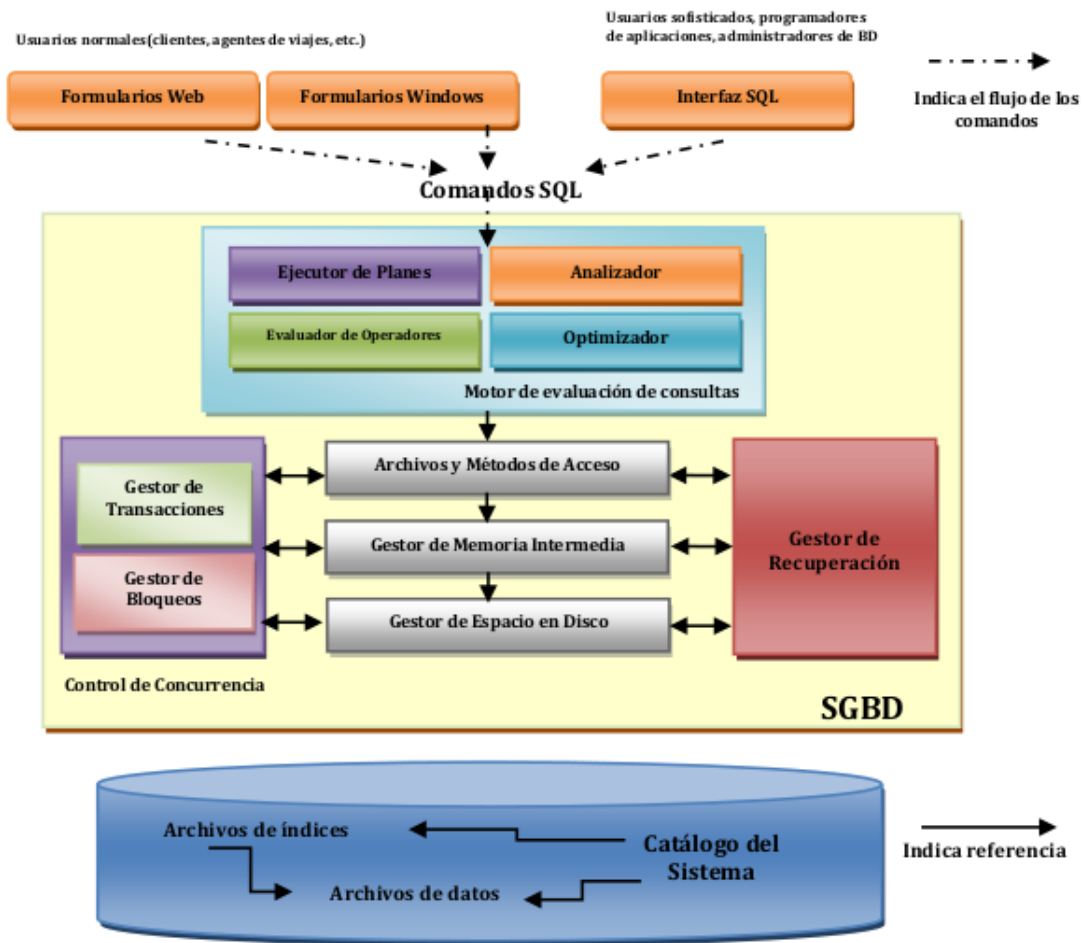
Garantiza que las transacciones soliciten y liberen los bloqueos de acuerdo con el correspondiente protocolo de bloqueo y programa la ejecución de las transacciones.

#### Administrador de bloqueos

Realiza un seguimiento de las solicitudes de bloqueo y concede los bloqueos sobre los objetos de la base de datos cuando quedan disponibles.

### 1.2.4. Administración de tolerancia a fallos

Responsable del mantenimiento de un registro y de la restauración del sistema a un estado consistente tras los fallos.



## 1.3. Clasificación de los SGBD

### 1.3.1. Modelo de datos

**Modelo:** Es una representación de la realidad que contiene características generales de algo que se va a realizar.

**Modelo de Datos:** Conjunto de conceptos que pueden servir para describir la estructura de la Base de Datos. La estructura está formada por: Datos, Tipo de datos, Vinculos, Restricciones, y puede incluir un conjunto de operaciones para especificar lecturas y actualizaciones.

Por ejemplo: Jerárquico, Relacional, Red, Orientado a Objetos, Entidad-Relación.



### 1.3.2. Número de Usuarios

- Monousuario
- Multiusuario

### 1.3.3. Proposito

- Específico
- General

### 1.3.4. Costos

- Gratis
- Licencia

### 1.3.5. Número de sitios

- Centralizado
- Distribuido: En varios servidores para evitar fallos.
  - Homogeneo: Los servicios son del mismo control
  - Heterogeneo: Interfaces de comunicación

## 1.4. Clasificación de Modelo de Datos

- Alto nivel/Conceptual Tiene conceptos muy cercano al modo en que el usuario percibe los datos.
- Representación/Implementación Sus conceptos pueden ser entendidos por los usuarios finales aunque no estan alejados de la forma en que estan organizados en la Base de Datos.
- Bajo nivel/Fisico Describe los detalles de como se almacenan los datos.

## 1.5. Elementos de un Modelo de Datos

- Entidad: Representa un objeto del mundo Real
- Atributo: Característica que permite descubrir una entidad.
- Vinculo: Describe una interacción entre dos entidades.
- Esquema de la Base de Datos: Descripción de la estructura de la Base de Datos.
- Diagrama del esquema de la Base de Datos: Representación gráfica del esquema de la base de datos.
- Estado de la Base de Datos: Datos en determinado tiempo.

Administrador de transacciones(estrategia de ejecución)

Administrador de bloqueos (Granularidad)

Bitácora.

### 1.5.1. Arquitectura de un SGBD

#### Tipos de usuarios

- Usuarios no sofisticados:
- Usuarios sofisticados:

#### Formas Web

#### Aplicaciones front-end

#### Interfaces SQL

### 1.5.2. SQL Lenguaje Estructurado de Consultas

#### Bloque 1

- Administrador de transacciones
- Administrador de bloques

**Bloque 2**

- Ejecutor del Plan
- PARSER
- Evaluador de Operaciones
- OPTIMIZADOR

**Bloque 3**

- Archivos y métodos de acceso
- Administrador de páginas
- Administrador de Espacio en disco

**Bloque 4**

- Administrador de Recuperación

**Bloque 5 Base**

Catalogo de Sistema

- Archivo de datos.
- Archivos de indices.

**1.5.3. Personas involucradas en un SBD**

**Administrador de Base de Datos**

**Diseñador de BD**

**Analistas de Sistemas-Programadores**

**Usuarios Finales**

- Usuarios Esporádicos
- Usuarios paramétricos/Simples

- Usuarios Autónomos
- Usuarios Avanzados

## 1.6. Normalización

### 1.6.1. Que es?

.Es el proceso de eliminación de redundancias en una tabla para que sea más fácil"

### 1.6.2. Descripcion

Tenemos que tener una relación bien estructurada, osea que contenga el minimo de redundancia y permita a los usuarios insertar, modificar y borrar registros en una tabla sin errores o inconsistencias.

Ademas debemos de evitar los 3 tipos de anomalías: Anomalías de inserción, anomalías en eliminación y anomalías en actualización.

Aqui es donde entra la normalización, el cual es un proceso formal ara decidir que atributos deberían ser agrupados en una relación, es el proceso de descomponer relaciones con anomalías para producir relaciones pequeñas y bien estructuradas.

### 1.6.3. Dependencias Funcionales

Es una restricción entre dos conjuntos de atributos de la base de datos. Una dependencia funcional, denotada por  $X \rightarrow Y$ , entre dos conjuntos de atributos  $X$  e  $Y$  que son subconjuntos de  $R$ , especifica una restricción sobre las posibles tuplas que podrán formar un estado de relación  $r$  de  $R$ .

La restricción dice que, para dos tuplas cualesquiera  $t_1$  y  $t_2$ , de  $r$  tales que  $t_1[X] = t_2[X]$ , debemos tener también  $t_1[Y] = t_2[Y]$ . Esto significa que los valores del componente  $Y$  de una tupla  $r$  dependen de los valores del componente  $X$ , o están determinados por ellos; o bien, que los valores del componente  $X$  de una tupla determinan la manera única (o funcionalmente) los valores del componente  $Y$ .

### 1.6.4. Forma Normal

Es un estado de una relación que resulta de aplicar simples reglas tomando en cuenta la dependencia funcional (o relaciones entre los atributos) de una relación.

- 1. Primera Forma Normal.(Redundancia y Atributos atómicos) Cualquier atributo multivalor (también llamado grupo repetitivo) tiene que ser eliminado.

Eliminar redundancia y convertir los atributos complejos en atributos atómicos, o no descomponibles.

- 2. Segunda Forma Normal(Los atributos solo depende de una clave). Cualquier dependendencia funcional parcial tienen que ser eliminadas, es decir, los atributos no claves son identificados por la llave primaria.

Un esquema de relación R está en 2FN si todo atributo no primo A en R depende funcionalmente de manera total de la clave primaria de R.

Se le puede “normalizar en 2FN” dando lugar a varias relaciones 2FN en las que los atributos no primos estén asociados sólo a la parte de la clave primaria de la que dependen funcionalmente de manera total.

- 3. Tercera Forma Normal(Atributos no llave no dependen de atributos no llave). Cualquier dependendencia funcional parcial tienen que ser eliminadas, es decir, los atributos no claves son identificados por la llave primaria.

Se elimina las dependencias transitivas; es decir atributos no clave no dependen de otros atributos no clave.

- 4. Boyce/Codd Forma Normal(Claves no dependan de atributos no clave). Cualquier anomalía resultante de dependencias funcionales tienen que ser eliminadas.

Un atributo clave es funcionalmente dependiente de un atributo no clave.

“Una relación R está en la FNBC si y sólo si cada determinante es una llave candidata”.

La relación es modificada de tal manera que el determinante de la relación que no es una llave candidata llega ser un componente de la llave primaria de la relación revisada. El atributo que es funcionalmente dependiente en el determinante llega a ser un atributo no clave. Esto es válido por la dependencia funcional

- 5. Cuarta Forma Normal. Cualquier dependendencia multivaluada tienen que ser eliminadas.
- 6. Quinta Forma Normal. Cualquier dependendencia de junta o de proyección tienen que ser eliminadas.

# Capítulo 2

## SQL

### 2.1. Introducción

Es un sublenguaje formado por:

- DDL Lenguaje de definición de datos.
- DML Lenguaje de manejo de datos
- CTL Lenguaje de control de transacciones.
- SDL Lenguaje de definición de almacenamiento.
- VDL Lenguaje de definición de vistas.

### 2.2. Aldunas cosas a tener en cuenta

Crea primero todas las restricciones de llaves foraneas y primarias.

### 2.3. Algunas sentencias

*Show Databases*; Muestra las bases que tenemos.

*drop database **NomBD***; Elimina una base de datos.

*create database **NomBD***; Crea una base de datos.

*use **NomBD***; Entra a la base de datos.

*Show tables* Muestra las tables en mi base de datos.

*Create table **Nomtable**(Atributos);* Crear una tabla donde atributo tiene nombre de variable, tipo y restricción.

*describe **Nomtable**;* Muestra la tabla.

*show create table **Nomtable**;* Describe la manera de creacion de la tabla.

*Alter table **Nomtable** add constraint PK Primary Key (idP);* Modificar una tabla y agrega restricción llamada Pk(unico) con su tipo Primary key Nombre del atributo.

*Alter table **Nomtable** add constraint FKP Foreign Key (**NomDato**) references **Nomtable**(**NomDato**);*Ligar tablas. Modificamos la table una restricción de llave foranea (cual) una referencia a tabla (Dato)

*Insert into **NomTable**(Atributos) values(Valores Atributos);* Inserta valores en una tabla respecto a su posición.

*Select \* From **NomTable**;* Muestra una tabla.

*alter table NomTable add column NomDato type null;* Agregar una columna

*alter table NomTable modify NomDato TypetoChange null;* Modificar el tipo de dato de una columna.

*alter table NomTable change NomDato NewName Type null;* Cambiar Nombre de dato;

*alter table NomTable drop column NomDato;* Eliminar una columna de una tabla.

*Delete from nombre;* Eliminar tabla;

*Delete from NomTable where IdP="P3";* Eliminar un dato.

*alter table NomTable drop Foreign Key nombreConstraint;*

*Delete from NomTable where IdP="P2"* Borra en cascada, si en la tabla original se desea eliminar entonces en la dependiente tambien se elimina

*Alter table **Nomtable** add constraint FKP Foreign Key (**NomDato**) references **Nomtable**(**NomDato**) on delete cascade on update cascade;* Ligar tablas. Modificamos la table una restricción de llave foranea (cual) una referencia a tabla (Dato) y decimos que elimine en cascada y actualizacion tambien.

*update NomTable set NomAtrib="P7"where NomAtrib="P3"* Modificarel valor de un dato ya contenido en la tabla en cascada en un tipo de atributo.

*Select \* from **NomTable** order by **NomAtrib** ASC;* Debe ir al final de la sentencia y ASC es orden ascendente aunque este esta por default

*Select \* from **NomTable** order by **NomAtrib** desc;* Debe ir al final de la sentencia y desc es orden descendente

*Select \* from **NomTable** order by **NomAtrib1** desc, **NomAtrib2** asc;* Ordena primero por Atrib1 y despues por Atrib2

*Select distinct **NomAtrib** from **NomTable**;* Elimina duplicados.

*Select \* from **NomTable**, (Select \* from **NomTable**) A;* Regresa un producto cruz y se crea una relación A que es la parte donde esta el parentesis, no se guarda físicamente..

*Select count(**NomAtrib**) from **NomTable**;* Cuenta tuplas sin contar nulls.

*Select sum(**NomAtrib**) from **NomTable**;* Suma.

*Select avg(**NomAtrib**) from **NomTable**;* Promedio.

*Select min(**NomAtrib**) from **NomTable**;* Valor minimo. En alfanuméricos es por orden de alfabeto.

*Select max(**NomAtrib**) from **NomTable**;* Valor maximo. En alfanuméricos es por orden de alfabeto.

Suma y promedio solo para datos numericos. max min y contador para alfanumericos

*Select (**NomAtrib**) from **NomTable** group by **NomAtrib2**;* Agrupar por atrib2

### 2.3.1. Crear un respaldo de la BD

*Mysqldump - -user=root - -password=root nombredb >Dir/archivo.sql* Donde dir es la direccion donde se guardará el archivo resultado.

### 2.3.2. Crear un repado por un puerto diferente

*Mysqldump - -user=root -P 3309 nombredb >Dir/archivo.sql* Donde -P es el puerto a escuchar.

### 2.3.3. Crear una bd a partir de un respaldo

*Mysql - -user=root - -password=root -P 3309 nombredb <Dir/archivo.sql*

*mysql -user=root -password=root Parejas2 <BD2CM1/archivo.sql* *mysqldump -user=root -password=root Parejas >'BD2CM1/archivo.sql'*

*mysqldump -user=Lalaandrea10 -password=root Parejas >BD2CM1/archivo.sql*

*select nombre as Name from profesor;* Alias de nombre es name;

*select (edad+20)x10 as operacion from profesor;* Crea una tabla llamada (edad+20)x10 y la renombra como operación



# Anexos