

# Índice

# 1. Introduccion a las bases de datos

## 2. Conceptos de sistemas y arquitectura de bases de datos

### 2.1. Concepto de Base de datos

Una base de datos se puede definir como un conjunto de datos "ordenados", definidos por un modelo de datos, los cuales se almacenan en dispositivos de almacenamiento.

Su finalidad es: Manipular, controlar, consultar, preservar los datos de la misma.

### 2.2. Sistemas de Base de Datos

Un sistema de base de datos es aquel Sistema de software que involucre el almacenamiento y consulta de datos cuando sea requerido.

### 2.3. Aplicaciones de los Sistemas de BD

### 2.4. Proposito de los sistemas de BD

Los sistemas de bases de datos tienen como intención solucionar los problemas que se derivan de utilizar un sistema meramente de archivos. Algunos de estos problemas son:

- Los datos aparecen de manera redundante.
- Los datos son inconsistentes.
- Los datos se encuentran totalmente aislados.
- No es posible el acceso concurrente

En cambio, los Sistemas de Bases de Datos cumplen con las siguientes características:

- Minimizan la redundancia en los datos.
- Eliminan inconsistencias en los datos.
- Permiten la comunicación con distintos repositorios de datos.
- Mantiene un control de concurrencia en las transacciones.

[Nota]

Una transacción es una operación que afecta al Repositorio de datos, pueden ser de diferentes tipos como:

- Inserción
- Actualización
- Eliminación
- Consulta

### 2.5. Usuarios de la BD

#### 2.5.1. Administrador de BD

Encargado de:

- Monitorear el performance.
- Define tiempos de respaldo.

- Proceso de tuning afinacion / reorganizacion fisica.
- Llevar a cabo las tecnicas de recuperacion.
- Administra herramientas [Objetos accedidos, Matriz de autorizacion: creacion de cuentas de usuarios]

### 2.5.2. Diseñadores de la BD

- Genera propios modelos de datos [conceptuales (E/R), logicos (Relacionales), fisicos (Indices)]
- Genera esquema de BD: [Medicas, Administrativas, Puntos de esquema]

—————Inserte aqui diagramas de ejemplo para modelos conceptuales y logicos.—————

### 2.5.3. Programadores de aplicaciones

- Interfaces de los usuarios finales (Dashboards) (facilita el acceso a ciertos.ºbjtos de la BD)
- Interfaces para la gestion de aplicaciones.
  - IDE's (java / .NET)
  - Lenguajes scripts
  - Conectividad con el servidor de datos (APIs)
  - Programacion Orientada a Objetos

### 2.5.4. Usuarios Finales

- Usuarios casuales (nivel gerencia)
  - Inteligencia (Big Data)
  - Negocios
  - Reportes (Toma de decisiones)
- Principales o parametricos
  - Puntos de ventas
- Sofisticados
  - Desarrolladores de apps
  - DBAs (Admins de Bases de Datos)
  - Investigadores
- Independientes (Stand Alone)
  - Proyectos Escolares
  - No hay conectividad con otros "nodos.<sup>a</sup> la internet

## 2.6. Ciclo de vida de una BD

### 1. Planificacion

- Viabilidad de la implementacion
- Supervisores, dueños, gerentes
- Contexto, ¿ Para que ?, citas, control, etc.
- Tiempos de entrega

### 2. Analisis y Formulacion de Requerimientos (Acciones que debe hacer nuestro sistema)

- Contexto
- Requerimientos [Basicos, Funcionales, No Funcionales]
- Reglas del negocio (Proceso) restricciones (IDE's)

### 3. Diseño

- Generar los modelos de datos [logicos, conceptuales y fisicos] orientados a objetos relacionales, multidimensionales.
- Aplicaciones de usuarios finales (Diseñar GUI) HCI (Human Computer Interaction)

### 4. Implementacion

- Construir el repositorio de datos
- Conectar al servidor

### 5. Operacion y Mantenimiento

- Crecimiento de usuarios y datos
- Posibles cambios que se deban realizar
- Errores y Correcciones
- Monitoreo

## 2.7. Modelos de de datos

### 2.7.1. Conceptual

- Permiten definir el ambito / context de la BD
- Lenguaje natural (idioma nativo)
- Comprendido por cualquier persona
- Proceso de analisis de requerimientos
- Diagrama Entidad / Relacion o Entidad / Asociacion

—————Inserte ejemplo de Esquema Conceptual—————

### 2.7.2. Logico

- Definen el esquema de base de datos en terminos de relacion
- Tablas:
  - Filas (tuplas)
  - Columnas (nombre atributo)
  - claves relacionales [Primary Key, Foreign Key, Candidate key]
  - Tipos de datos
  - Constraints (restricciones)
- Puedes construir la BD fisicamente
  - Manual
  - Automatica (Herramientas CASE)

—————Inserte ejemplo de Esquema Logico—————

### 2.7.3. Fisico

Depende del Sistema Operativo

- Define las estructuras de almacenamiento de datos.
- Define los indices(index) para acceder a los archivos fisicos del repositorio de datos.

—————Inserte ejemplo de Esquema Fisico—————

## 2.8. Arquitectura de 3 niveles

### 2.8.1. Nivel externo

Describe una parte de la BD que es relevante a un usuario en particular, excluye datos irrelevantes así como información a la que el usuario no se le tiene permitido el acceso.

Los usuarios finales son quienes visualizan este nivel por medio de GUIs.

### 2.8.2. Nivel conceptual

Describe:

- Que datos se almacenan.
- Como se relacionan los datos entre ellos.

Algunas de sus características son:

- El DBA trabaja en este nivel.
- Describe la estructura de todos los usuarios.
- Solo el DBA define este nivel.
- Da una vista global de la BD.
- Es independiente del software y hardware.

### 2.8.3. Nivel interno

Describe como la información es almacenada en la BD.

- Define las estructuras de almacenamiento (depende del S.O.)
- Define la construcción

## 2.9. Independencia de datos

- **Independencia Lógica:** Capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación.
- **Independencia Física:** Capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos).

## 2.10. Arquitectura de los SGBD

El gestor de datos es un sistema de software invisible para el usuario final, compuesto por un lenguaje de definición de datos, un lenguaje de manipulación y de consulta, que puede trabajar a distintos niveles.

1. Aplicaciones de usuarios finales
2. Comandos SQL
3. SGBD

a) Motor de evaluación de consultas

- 1) **Analizador o Compilador DDL/DML:** Realiza el análisis léxico (Gramática SQL) y sintáctico (formulación correcta), además de validar los objetos descritos en la consulta con los objetos definidos en los metadatos.
- 2) **Evaluador de Operadores:** Verifica los operadores de álgebra relacional que incluye consultas.
- 3) **Optimizador:** Representa la consulta en un "Árbol canónico" y un "Árbol de consulta". Reduce la cardinalidad (de tuplas) y el grado (de atributos).
- 4) **Ejecutor de planes:** Elige el mejor orden (estrategia) para ejecutar las consultas.

b) **Control de concurrencia:**

- Manejo de transacciones
- Propiedades ACID [Atomicity, Consistency, Isolation, Durability]

- Estados de una  $t_i$  [Activa, Parcialmente confirmado, Confirmada (Commit al repositorio de datos)]
- **Bloqueos**
  - Binario
  - Bloqueo de multiples niveles. [Bloqueo de lectura, Bloqueo de escritura, Desbloqueo].
- **Marcas de tiempo**
  - Semaforo
  - Tiempo de ejecucion
- **Gestor de transacciones**
  - Gestiona las  $t_i$
  - **Bitacoras de las  $t_i$** 
    - 1) inicia
    - 2) read, x
    - 3) write, x, valor previo, nuevo valor
    - 4) abort, x
    - 5) commit, x

c) Archivos

- Catalogo del sistema
- Archivo de indices
- Archivos de datos

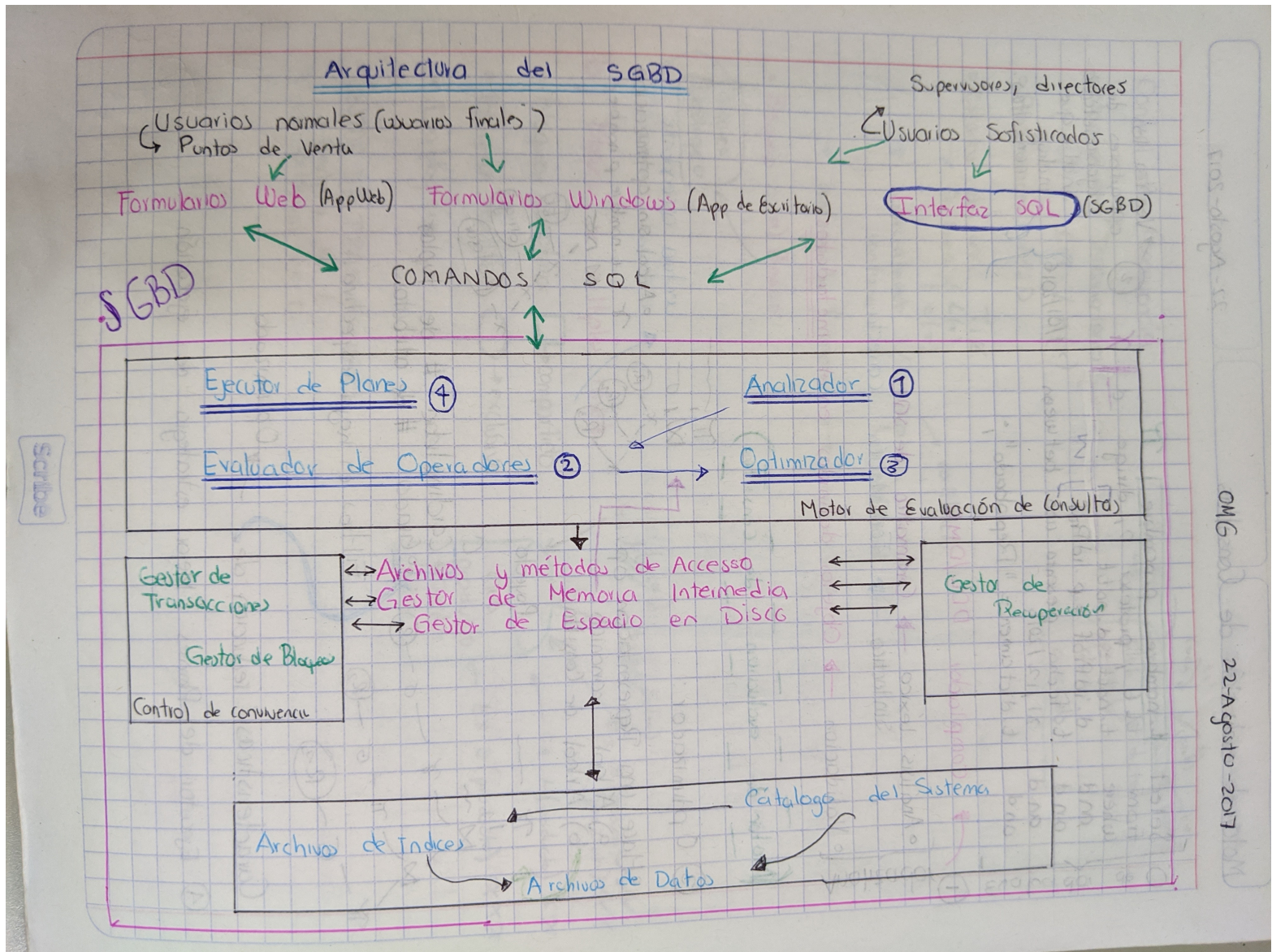


Figura 1: Arquitectura de un SGBD

### 2.11. Arquitectura Cliente/Servidor para los SGBD

### 2.12. Arquitectura Centralizada para los SGBD

### 2.13. DDL Data Definition Language

Involves:

- create table
- drop table

### 2.14. DML Data Manipulation Language

Involves:

- insert into
- delete
- select
- update

## 3. Algebra relacional y SQL standar

**DEF.** Una expresion en el algebra relacional se construye a partir de subexpresiones.

Sean  $E_1$  &  $E_2$  expresiones del algebra relacional. Las siguientes son todas expresiones **binarias** del algebra relaional.

- **Union:**  $E_1 \cup E_2$  (Se requiere que el numero de columnas sea el mismo entre tablas asi como los nombres de columnas.)
- **Diferencia:**  $E_1 - E_2$
- **Producto Cartesiano:**  $E_1 \times E_2$

Y las siguientes son operaciones **unarias**:

- **Seleccion:**  $\sigma_p(E_1)$  donde  $p$  es un predicado de atributos de  $E_1$ . (Trabaja con renglones completos)
- **Proyeccion:**  $\Pi_s(E_1)$  donde  $s$  es un conjunto de atributos de  $E_1$ . (Trabaja con columnas completas)
- **Renombramiento:**  $P_x(E_1)$  donde  $E_x$  es un nuevo nombre de la relacion  $E_1$ .



Sean  $R$ ,  $S$  y  $T$  entidades ejemplificaremos las operaciones mencionadas anteriormente:

R			S			T		
A	B	C	A	B	C	B	C	D
a	1	a	a	1	a	a	1	a
b	1	b	a	3	f	3	b	1
a	1	d				3	c	2
b	2	f				1	d	4
						2	a	3

Seleccion

$SL_{A=a}(R) = \sigma_{A=a}(R) \Rightarrow$

Resultado		
A	B	C
a	1	a
a	1	d

Proyeccion

$PJ_{A,B}(R) = \Pi_{A,B}(R) \Rightarrow$

Resultado	
A	B
a	1
b	1
a	1
b	2

Union

$Union(R,S) = R \cup S \Rightarrow$

Resultado		
A	B	C
a	1	a
b	1	b
a	1	d
b	2	f
a	3	f

Diferencia

$Dif(R,S) = R - S \Rightarrow$

Resultado		
A	B	C
b	1	b
a	1	d
b	2	f

Producto Cartesiano

$PC(R, S) = R \times S \Rightarrow$

Resultado					
R.A	R.B	R.C	S.A	S.B	S.C
a	1	a	a	1	a
b	1	b	a	1	a
a	1	d	a	1	a
b	2	f	a	1	a
a	1	a	a	3	f
b	1	b	a	3	f
a	1	d	a	3	f
b	2	f	a	3	f

Producto Cartesiano

$PC(R, T) = R \times T \Rightarrow$

Resultado					
R.A	R.B	T.B	R.C	T.C	T.D
a	1	1	a	a	1
b	1	1	b	a	1
a	1	1	d	a	1
b	2	1	f	a	1
a	1	3	a	b	1
b	1	3	b	b	1
a	1	3	d	b	1
b	2	3	f	b	1
a	1	3	a	c	2
b	1	3	b	c	2
a	1	3	d	c	2
b	2	3	f	c	2
a	1	1	a	d	4
b	1	1	b	d	4
a	1	1	d	d	4
b	2	1	f	d	4
a	1	2	a	a	3
b	1	2	b	a	3
a	1	2	d	a	3
b	2	2	f	a	3

## JOIN

$$JOIN(R, T)_{R.C=T.C} \Rightarrow$$

Resultado					
R.A	R.B	R.C	T.B	T.C	T.D
a	1	a	1	a	1
b	1	b	3	b	1
a	1	d	1	d	4
a	1	a	2	a	3

## 4. Analisis de una base de datos

## 5. Diseno de una base de datos

## 6. Oracle DB Commands

### 6.1. Start Oracle DB

```
[root@localhost ~]# . ./setupXEvars
```

The contents of this script are the following:

```
export ORACLE_SID=XE
export ORAENV_ASK=NO
. /opt/oracle/product/18c/dbhomeXE/bin/oraenv
/sbin/service oracle-xe-18c start
```

### 6.2. Login as sysdba (root)

```
[root@localhost ~]# sqlplus sys as sysdba
```

### 6.3. Alter Oracle Date Language

```
alter session set nls_session_parameters
alter session set nls_date_language='english'
select * from nls_session_parameters;
```

### 6.4. Create User

Our username will be "sergio" and our password will be also "sergio";  
Login as sysdba and enter the following commands.

```
alter session set "_ORACLE_SCRIPT"=true;
create user sergio identified by sergio default tablespace users temporary tablespace temp;
grant connect, resource to sergio;
alter user sergio quota unlimited on users;
alter user sergio quota unlimited on temp;
```

## 6.5. Create Student Database from script

```
SQL> @dbs/011/createStudent.sql
```

## 6.6. SPOOL Command

Saves **all** the **output** of the inserted commands **into** the given file (mylog.txt)

```
SQL> spool mylog.txt
```

## 6.7. TO\_DATE() Function

```
to_date('05-apr-2003 20:14:33', 'dd-mon-yyyy hh24:mm:ss');
```

## 6.8. ROWNUM Column

**ROWNUM:** Is a column appended to every table which corresponds to the index of every row in the table.

## 6.9. Install Sample Database

Login as sysdba and run the following commands.

```
alter session set "_ORACLE_SCRIPT"=true;  
@?/demo/schema/human_resources/hr_main.sql
```

# 7. Bibliografia

SQL by Example

# 8. Tareas

### 8.1. Investigar que es SQL [Done]

SQL stands for "Structure Query Language." and is a language used to perform queries in databases managers.

### 8.2. Investigar que es "SPOOL"[Done]

Is a command used to save the output of the commands we insert into the sqlplus console.

# 9. Student Database Schema

# APPENDIX D

# STUDENT DATABASE

# SCHEMA

