



UNIVERSIDAD
COOPERATIVA
DE COLOMBIA

Gestión de la Información

Gustavo Adolfo Gómez Gómez

**MSc. Gestión, aplicación y desarrollo de software
2024**

www.ucc.edu.co

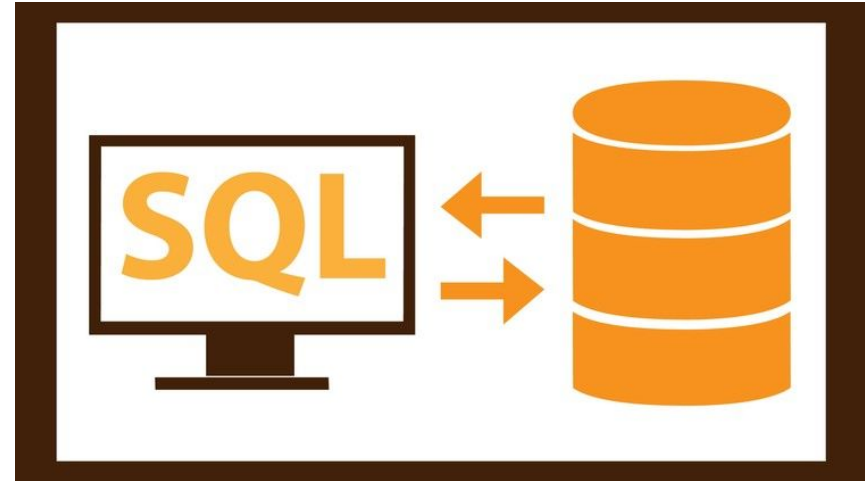
Gestión de la Información

Unidad 1: Asociar diseños, plataformas y soportes informáticos a los modelos de gestión de información de la organización.

Tema: Lenguaje Estructurado de Consultas: SQL

Lenguaje Estructurado de Consultas: SQL

- Basado en el álgebra relacional
- Lenguaje estructurado y formal
- No Solo “consultas”
 - Crear y definir estructuras de datos
 - Restricciones
 - Modificar datos



A propósito de sistemas Ó programas de gestión

- Sistema de gestión de bases de datos

SGBD ó DBMS (Database Management System)

- Permite la gestión de las funcionalidades de una base de datos
- No Solo guardar los datos
 - Ayuda a mantener la seguridad
 - Restricciones de acceso
 - Modificar datos

Existen los SGBD relacionales (SGBDR ó RDBMS)

Todo RDBMS implementan SQL



SQL - Componentes

- **Lenguaje de definición de datos**
 - **LDD - Definición**
 - Esquemas de relaciones (tablas)
 - Modificación relaciones
- **Lenguaje de manipulación de datos**
 - **LMD - Definición**
 - Consultas de datos
 - Basado en álgebra relacional
 - Insertar, borrar y modificar tuplas
- **Lenguaje de Control de datos**
 - **LCD - Definición**
 - Especifica Restricciones de integridad
 - Garantiza la seguridad de los datos



SQL - Componentes

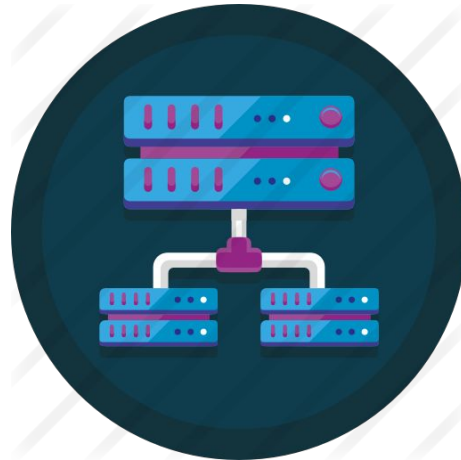
- **Definición de vistas**
 - Conjuntos de datos
- **Control de transacciones**
 - Procedimientos almacenados
 - Lógica de datos
- **Autorización**
 - Usuarios y roles
 - Opciones de acceso
 - Relaciones
 - Vistas
 - Operaciones



LDD: Definición de datos

Pero antes....

Sistemas de Bases de Datos



Sistemas de Bases de Datos

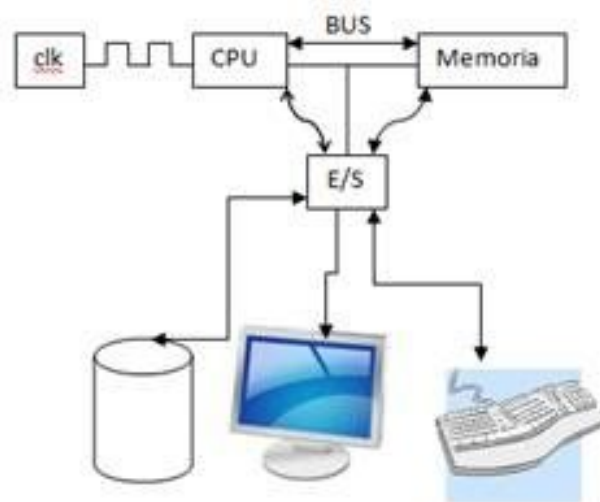
Colección de diferentes componentes de software de base de datos:

1. **Programas de aplicación de bases de datos**
2. **Componentes cliente**
3. **Servidor(es) de bases de datos**
4. **Bases de datos**

Sistemas de Bases de Datos

Arquitectura de los sistemas de bases de datos

Sistema operativo subyacente: Arquitectura Computacional donde se ejecuta



Sistemas de Bases de Datos

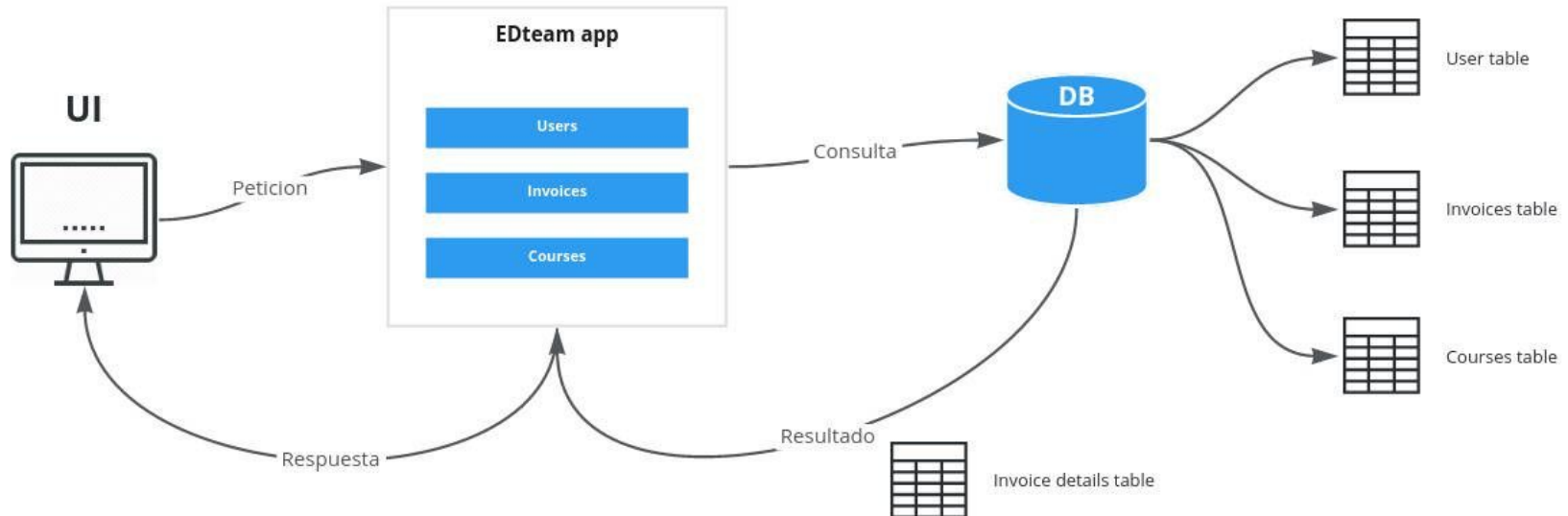
Tipos de arquitectura de los sistemas de bases de datos

1. **Sistema Centralizado**
2. **Arquitectura en RED (cliente - Servidor)**
3. **Procesamiento Paralelo**
4. **Almacenamiento Distribuido**

* Las mismas de cualquier otro sistema de información

Arquitectura Sistemas de Bases de Datos

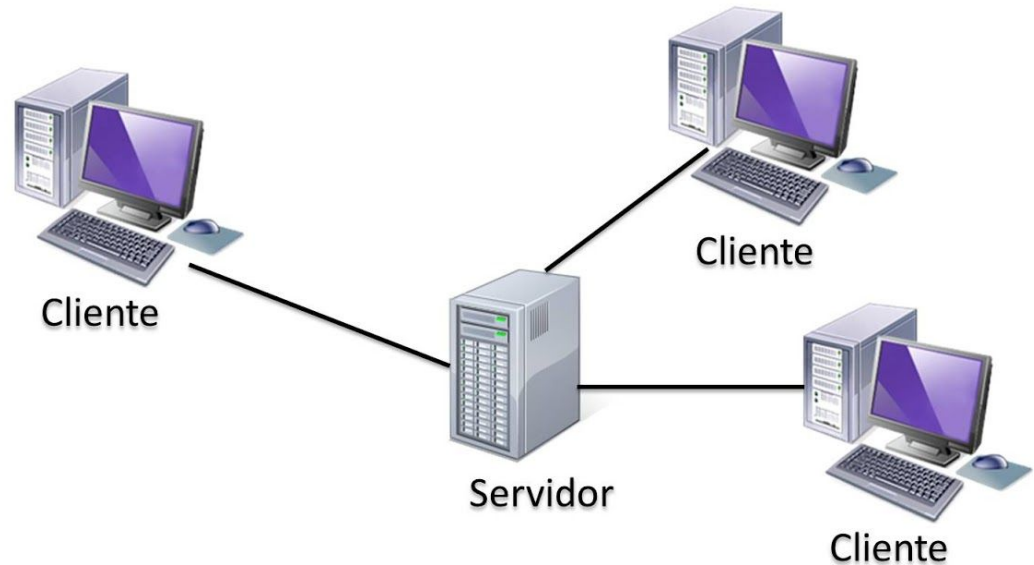
Sistema Centralizado: Monousuarios ó multiusuarios



Arquitectura Sistemas de Bases de Datos

Sistema Cliente-Servidor

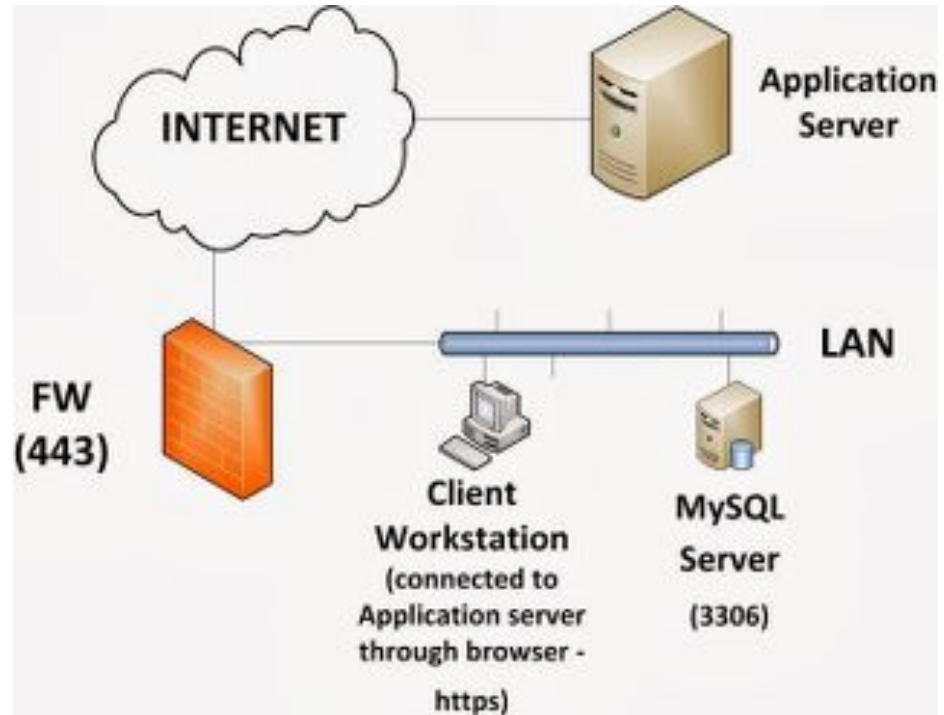
- Terminales (clientes) conectados a un sistema central que “sirve” la funcionalidad del SGBD
- Basados en el protocolo de red IP



Arquitectura Sistemas de Bases de Datos

Sistema Cliente-Servidor

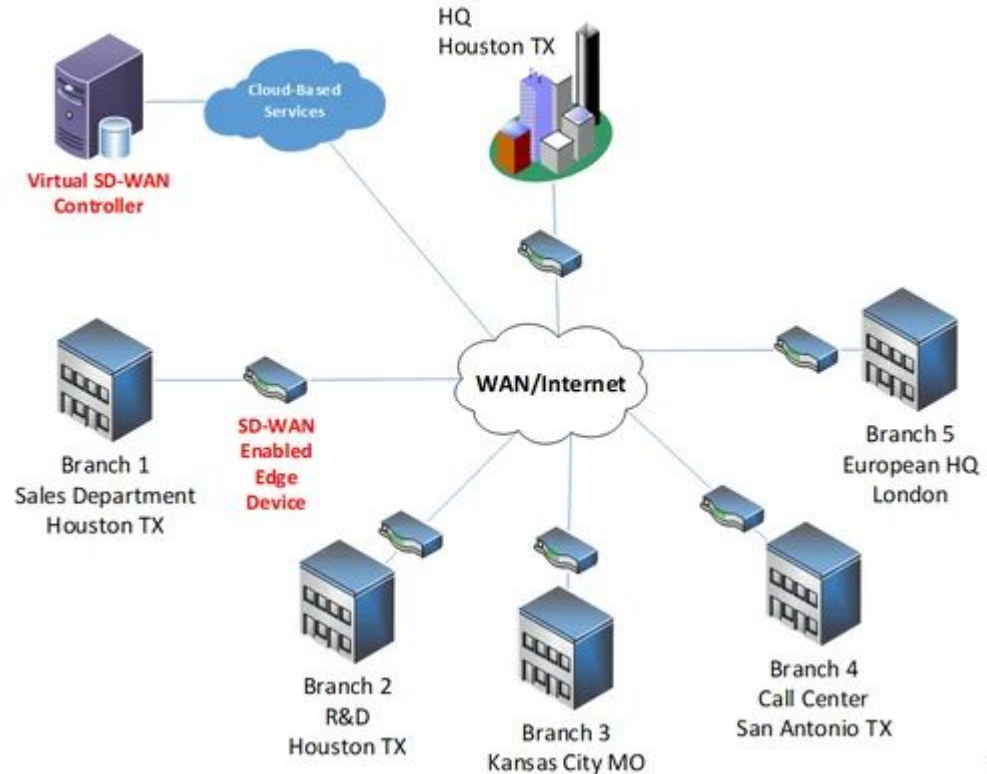
- Basados en el protocolo de red IP
 - Red de área Local (LAN)



Arquitectura Sistemas de Bases de Datos

Sistema Cliente-Servidor

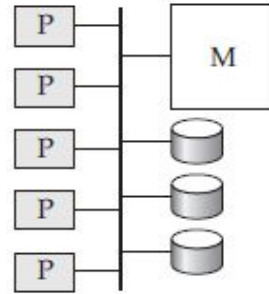
- Basados en el protocolo de red IP
 - Red de área Amplia (WAN)



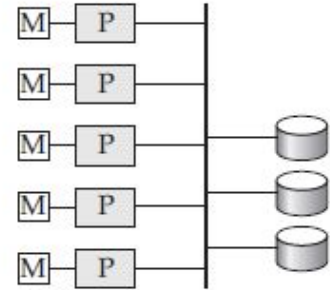
Arquitectura Sistemas de Bases de Datos

Sistema paralelo

- Hilos de ejecución (multiprocesamiento)
- Mayor velocidad (Sistemas actuales)
- Memoria compartida
- Disco compartido



(a) memoria compartida

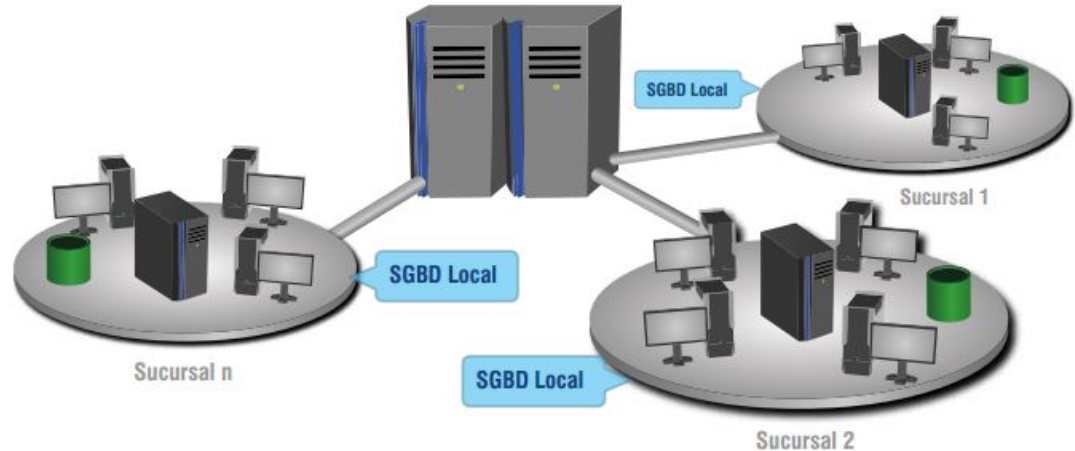


(b) disco compartido

Arquitectura Sistemas de Bases de Datos

Sistema Distribuido

- Redes de alta velocidad
- Nodos de distribución
- Datos autónomos y compartidos
- Alta disponibilidad



Componentes de los SGBD

- A. Lenguajes de los SGBD
- B. Diccionario de datos
- C. Seguridad e integridad de los datos
- D. Administrador de BD



Lenguajes de los SGBD

- **Lenguaje de definición de datos**
 - **LDD - Definición**
 - Esquemas de relaciones (tablas)
 - Modificación relaciones
- **Lenguaje de manipulación de datos**
 - **LMD - Definición**
 - Consultas de datos
 - Basado en álgebra relacional
 - Insertar, borrar y modificar tuplas
- **Lenguaje de Control de datos**
 - **LCD - Definición**
 - Especifica Restricciones de integridad
 - Garantiza la seguridad de los datos



Componentes de los SGBD

B. El Diccionario de Datos

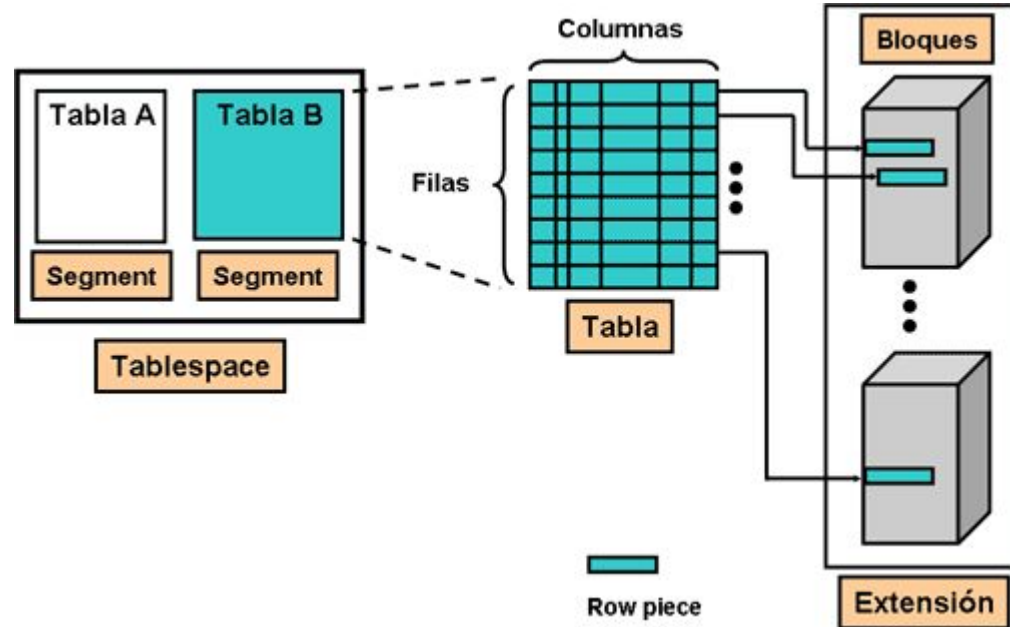
Descripción de la BD y los objetos que la conforman



Componentes de los SGBD

B. El Diccionario de Datos

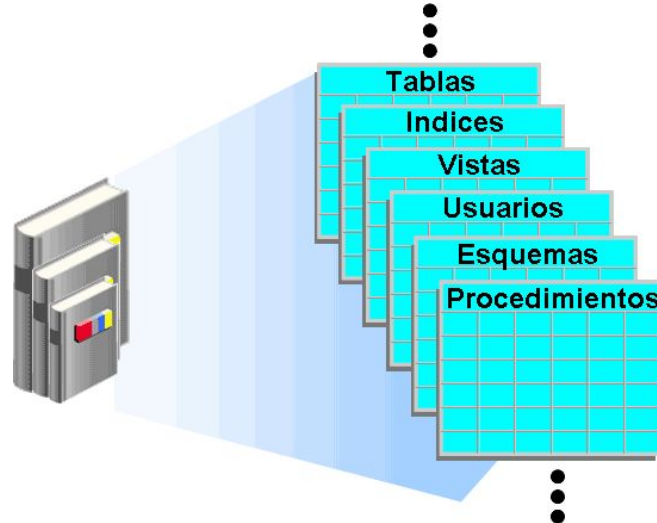
- Estructura lógica y física



Componentes de los SGBD

B. El Diccionario de Datos

- Definición de todos los objetos : tablas, vistas, índices, disparadores, etc



Componentes de los SGBD

B. El Diccionario de Datos

- Valores por defectos de columnas

Productos

Campo 1	Campo 2	Campo 3	Campo 4

Sucursales

Campo 1	Campo 2	Campo 3	Campo 4

Clientes

Campo 1	Campo 2	Campo 3	Campo 4

Componentes de los SGBD

B. El Diccionario de Datos

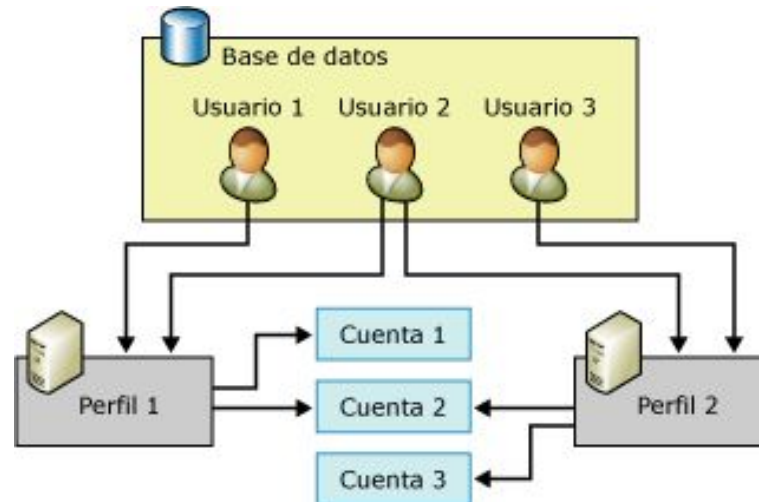
- Restricciones de integridad



Componentes de los SGBD

B. El Diccionario de Datos

- Privilegios y roles otorgados



Componentes de los SGBD

C. Seguridad e integridad de los datos

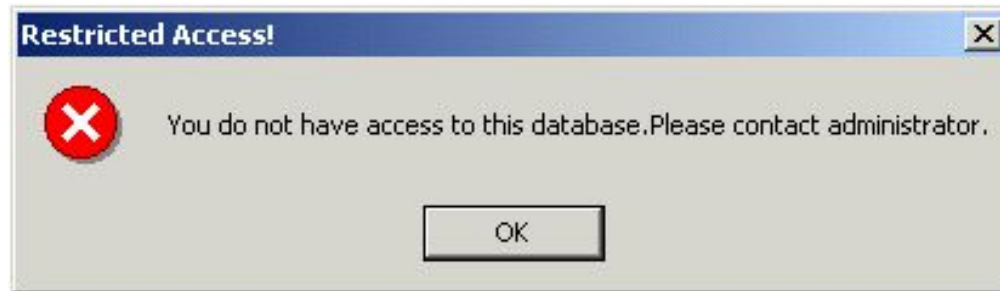
Un SGBD proporciona diferentes mecanismos para garantizar la seguridad e integridad



Componentes de los SGBD

C. Seguridad e integridad de los datos

- Garantiza la protección contra accesos no autorizados



Componentes de los SGBD

- C. Seguridad e integridad de los datos
- Implementa restricciones de integridad



Componentes de los SGBD

C. Seguridad e integridad de los datos

- Herramientas para la planificación y realización copias de seguridad



Componentes de los SGBD

C. Seguridad e integridad de los datos

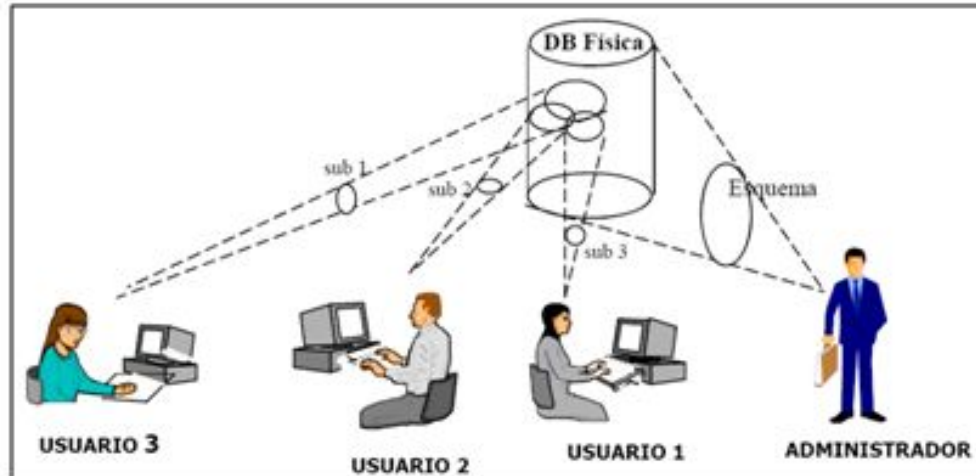
- Asegurar la recuperación de la BD a un estado consistente



Componentes de los SGBD

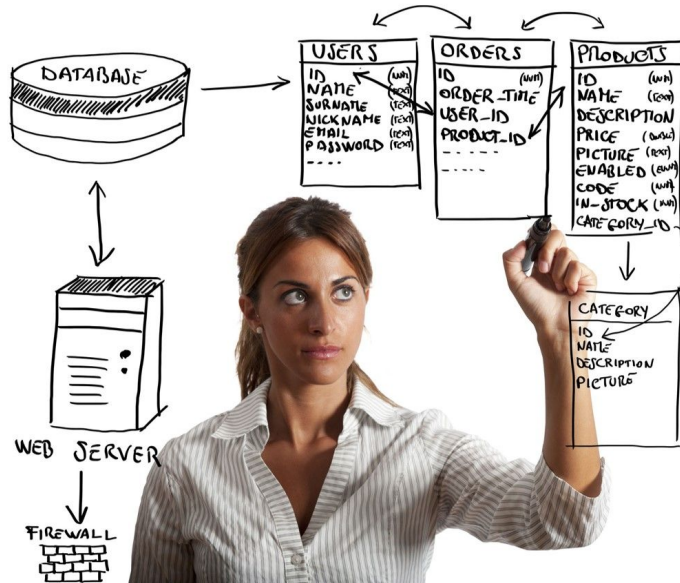
C. Seguridad e integridad de los datos

- Asegurar acceso concurrente de usuarios



Componentes de los SGBD

D. Administrador de la Base de Datos



Componentes de los SGBD

D. Administrador de la Base de Datos (Database Administrator ó DBA)

Máximo nivel de privilegios

Objetivo: “Garantizar que la BD cumple los fines previstos por la Organización”

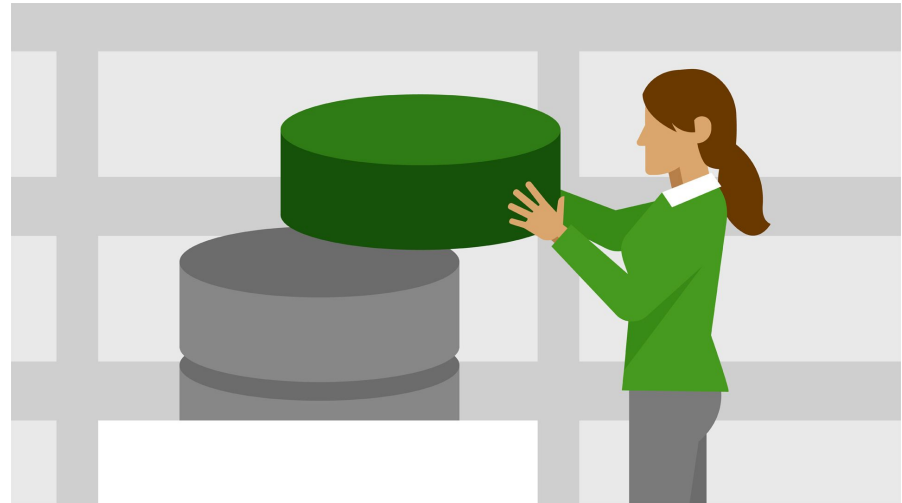
TAREAS del DBA



Componentes de los SGBD

D. Administrador de la Base de Datos

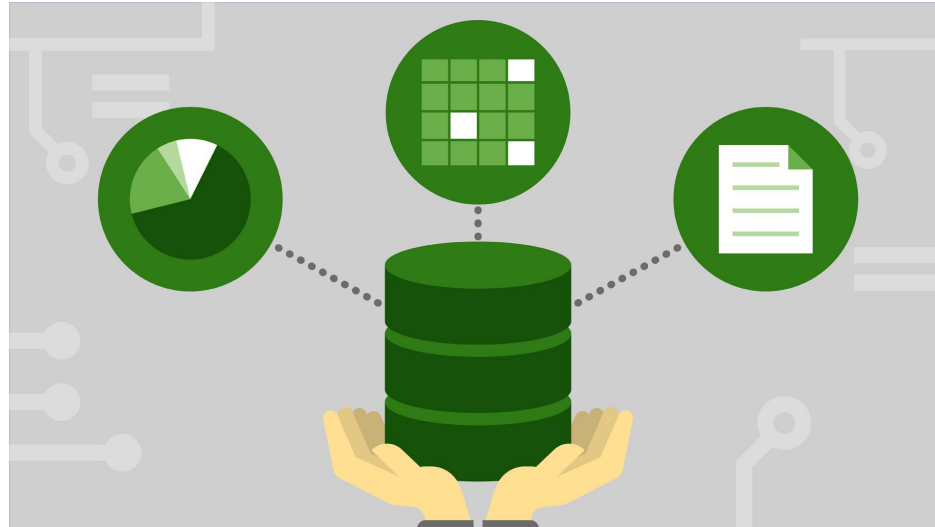
- Instalar SGBD en el sistema informático y crear las Bases de datos que se vayan a gestionar



Componentes de los SGBD

D. Administrador de la Base de Datos

- Crear y mantener el esquema de la Bd



Componentes de los SGBD

D. Administrador de la Base de Datos

- Crear y mantener las cuentas de usuarios



Componentes de los SGBD

D. Administrador de la Base de Datos

- Gestionar disponibilidad de servidores de BD



Componentes de los SGBD

D. Administrador de la Base de Datos

- Establecer estándares de uso, políticas de acceso y protocolos para usuarios



Componentes de los SGBD

D. Administrador de la Base de Datos

- Suministrar información necesarios a equipos de análisis y programación



SGBD:

Y Ahora: Con qué SGBD podemos trabajar?



MySQL

- Libre * (Opensource)
 - Versión Comunitaria bajo licencia GPL
- Multiplataforma
- Ligero
- Cliente - Servidor
- Desarrollado en C++
- API
 - Librerías de conexión Lenguajes de Programación



* Desde la versión 6 es adquirida por Oracle. La versión libre se llama ahora MariaDB

MySQL

Conexión BD

- Protocolo: TCP/IP
 - Sockets (puerto)
 - Pipes
 - Ficheros locales

```
$ mysql -h servidor.misitio.org -u <usuario> -p
```

MySQL

Conexión BD

- -h servidor de base datos (localhost, 127.0.0.1)
- -u usuario
- -p (clave)

```
$ mysql -h servidor.misitio.org -u <usuario> -p
```

MySQL

Conexión BD

- -h servidor de base datos (localhost, 127.0.0.1)
- -u usuario
- -p (clave)

```
C:\Users\Acer>mysql -h localhost -u root -p
Enter password: *****
```

MySQL

Conexión BD

```
C:\Users\Acer>mysql -h localhost -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.7.23-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> _
```

Conexión BD

Verificación conexión:

select user(), connection_id(), version(), database();

```
mysql> select user(), connection_id(), version(), database();
+-----+-----+-----+-----+
| user()          | connection_id() | version() | database() |
+-----+-----+-----+-----+
| root@localhost | 10              | 5.7.23-log | NULL       |
+-----+-----+-----+-----+
1 row in set (0.49 sec)
```


Conexión BD

Indicadores Consola de comandos:

Indicadores de *mysql*

Indicador	Significado
mysql>	Espera una nueva sentencia
->	La sentencia aún no se ha terminado con ;
">	Una cadena en comillas dobles no se ha cerrado
'>	Una cadena en comillas simples no se ha cerrado

Conexión BD

Variables y expresiones

select @x := 1;

```
mysql> select @x := 1;
+-----+
| @x := 1 |
+-----+
|      1 |
+-----+
1 row in set (0.13 sec)
```

MySQL

Conexión BD

Variables y expresiones

select @x, sqrt(@x), sin(@x), @x + 10, @x > 10;

```
mysql> select @x, sqrt(@x), sin(@x), @x + 10, @x > 10;
+-----+-----+-----+-----+-----+
| @x    | sqrt(@x) | sin(@x) | @x + 10 | @x > 10 |
+-----+-----+-----+-----+-----+
| 1     | 1        | 0.8414709848078965 | 11      | 0       |
+-----+-----+-----+-----+-----+
1 row in set (0.07 sec)
```

MySQL

Comandos

- Show databases;
- Use [database];
- Show tables;
- Describe [table];
- Create / drop [database];



MySQL

Tipos de datos

- Numéricos
- Cadenas de caracteres
- Fechas
- Horas



Tipos de datos

- Numéricos
 - Unsigned

Tipos

Tipo	Espacio de almacenamiento	Significado
<code>tinyint[(M)]</code>	1 byte	Entero muy pequeño
<code>smallint[(M)]</code>	2 bytes	Entero pequeño
<code>mediumint[(M)]</code>	3 bytes	Entero mediano
<code>int[(M)]</code>	4 bytes	Entero
<code>bigint[(M)]</code>	8 bytes	Entero grande

Tipos de datos

- Numéricos
 - Unsigned

Tipos

Tipo	Espacio de almacenamiento	Significado
float	4 bytes	Simple precisión
double	8 bytes	Doble precisión
decimal	M + 2 bytes	Cadena de caracteres representando un número flotante

Tipos de datos

- Cadenas de caracteres

Tipos

Tipo	Equivalente	Tamaño máximo	Espacio de almacenamiento
char[(M)]		M bytes	M bytes
varchar[(M)]		M bytes	L+1 bytes
tinytext	tinyblob	2^8-1 bytes	L+1 bytes
text	blob	$2^{16}-1$ bytes	L+2 bytes
mediumtext	mediumblob	$2^{24}-1$ bytes	L+3 bytes
longtext	longblob	$2^{32}-1$ bytes	L+4 bytes
enum('v1','v2',...)		65535 valores	1 o 2 bytes
set('v1','v2',...)		64 valores	1 a 8 bytes

MySQL

Tipos de datos

- Fechas y Horas

Tipos

Tipo	Espacio de almacenamiento	Rango
date	3 bytes	'1000-01-01' al '9999-12-31'
time	3 bytes	'-838:59:59' a '838:59:59'
datetime	8 bytes	'1000-01-01 00:00:00' a '9999-12-31 23:59:59'
timestamp[(M)]	4 bytes	19700101000000 al año 2037
year[(M)]	1 byte	1901 a 2155

LDD: Definición de datos

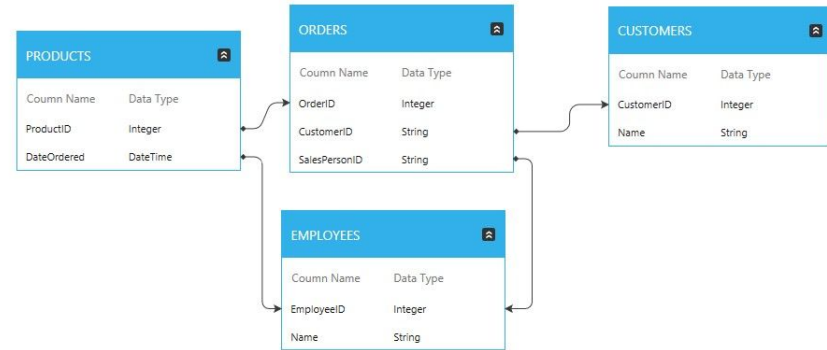
Ahora sí, devuelta a LDD ó lenguaje de definición de datos.

Lenguaje de definición de datos - DDL

LDD: Definición de datos

Para cada relación (tabla) se especifica:

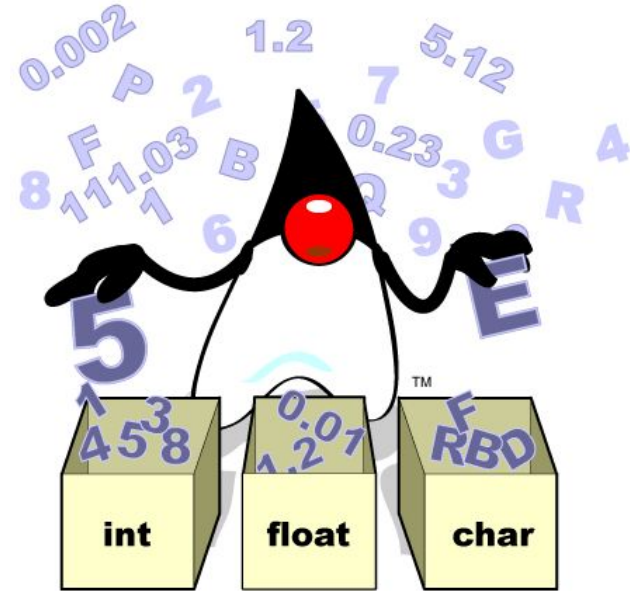
- Esquema(Estructura/campos/atributos)
- Dominio de valores de cada campo
- Restricciones de integridad
- Índices
- Seguridad y autorización
- Estructura almacenamiento físico



LDD: Definición de datos

Tipos básicos de dominios de datos

- **Char(n)**: cadena caracteres de longitud fija
- **varchar(n)**: cadena caracteres variable con long máxima=n
- **Int**: Entero (longitud depende del RDBMS)
- **numeric(d,p)**: números decimales de precisión d=dígitos, p=precisión después de la coma
- **Date**: tipo de dato de fecha (yyyy-mm-dd)



Definición básica de esquemas

Crear Relación

```
create table R (  
    Attr1 Dom1, Attr2 Dom2, ..... Attrn Domn,  
    (restricción-integridad1),  
    .....,  
    (restricción-integridadn)  
    )
```



Definición básica de esquemas

R=Nombre de la relación

Atributos y su tipo de dato (dominio)

Restricciones de integridad sobre atributos

- No nulos
- Únicos
- Defecto

Ejp: nombres varchar(100) **NOT NULL**

Ejp: correo_electronico varchar(200) **UNIQUE**

Ejp: activo char(2) **DEFAULT 'si'**

```
create table R (  
    Attr1 Dom1,  
    Attr2 Dom2, .....  
    Attrn Domn,  
)
```

Definición básica de esquemas

R=Nombre de la relación

Restricciones de integridad:

- Llave primaria (NO Nulos, Únicos)
- Restricción de datos
- Llaves foráneas

```
create table R (  
    Attr1 Dom1, Attr2 Dom2, ..... Attrn Domn,  
    (restricción-integridad1),  
    .....,  
    (restricción-integridadn)  
)
```

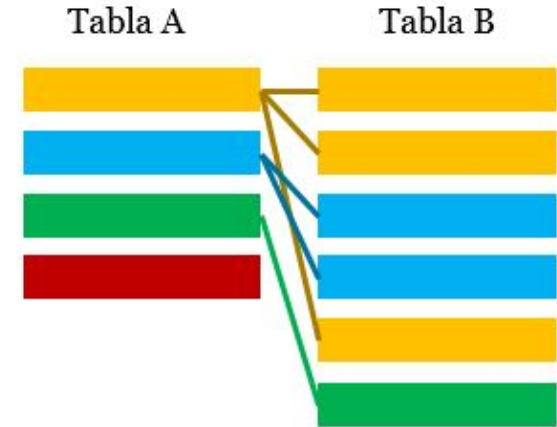
Ejp: **PRIMARY KEY** (cedula)

Ejp: **CHECK**(activo in (val1,...valX))

Ejp: **FOREIGN KEY**(ciudad) **REFERENCES** Ciudad (id)

Integridad Referencial

- Garantizar la integridad de las relaciones
- Claves internas y externas
 - Primarias y foráneas
- Coherencia de los datos
- Modificación específica no altera



Integridad Referencial

Se utilizan al crear o modificar relaciones

```
Create table cliente (  
    Id_cliente int,  
    Nombre_cliente varchar(100),  
    Calle varchar(200),  
    Primary Key(id_cliente)  
)
```

```
Create table cuenta (  
    Id_cuenta int,  
    Id_cliente int,  
    saldo double(5,1),  
    Fecha Date(),  
    Primary Key(id_cuenta),  
    Foreign Key(Id_cliente) references cliente  
)
```

Integridad Referencial

Pueden tener condiciones

- Al actualizar dato
- Al eliminar dato

On Delete / On Update

- Prohibir (Restrict)
- Modificar Dependencia (Cascade)
- Poner nulo (Set Null)
- No hacer nada (No Action)



Integridad Referencial

Pueden tener condiciones

Ejp: no permitir modificar nómina



Create table empleado_telefono (

....

Foreign Key(nomina) references empleado on update restrict

)

Integridad Referencial

Pueden tener condiciones

Ejp: Si se actualiza la clave del depto
Modificar la dependencia en empleado



Create table empleado (

....

Foreign Key(clave_depto) **references** departamento **on update cascade**

)

Definición básica de esquemas

Ejemplo:

Relación de estudiantes

```
create table estudiante(  
    codigo_estudiante varchar(20),  
    numero_identificacion int(10),  
    nombre varchar(50) NOT NULL,  
    apellidos varchar(50),  
    direccion varchar(100) NULL,  
    genero char(1),  
    primary key (codigo_estudiante),  
    check( genero in('F','M') )  
)
```

Definición básica de esquemas

Borrar Relación

No puede tener datos (depende de RDBMS)

drop table R

R=Nombre de la relación

Ejp:

drop table estudiante



SQL

Definición básica de esquemas

Editar Relación

Agregar atributos (campos)

alter table R add AttrX DomX

R=Nombre de la relación

Atributo

Dominio de datos

Ejp: Agregar atributo de teléfono



SQL

alter table estudiante add telefono int(20)

Definición básica de esquemas

Eliminar atributos (campos)

alter table R drop AttrX

R=Nombre de la relación

Atributo

Ejp: Quitar atributo *nombres*



SQL

alter table estudiante drop nombres

Definición básica de esquemas

Eliminar/añadir restricciones

alter table R add restriccion

R=Nombre de la relación

restricción

Ejp: Agregar restricción de llave primaria en el atributo numero_identificacion

*alter table **estudiante** add primary key(numero_identificacion)*

Resumen Lenguaje de Definición de datos

- CREATE
- ALTER
- DROP

Conexión BD

Procesos por lotes

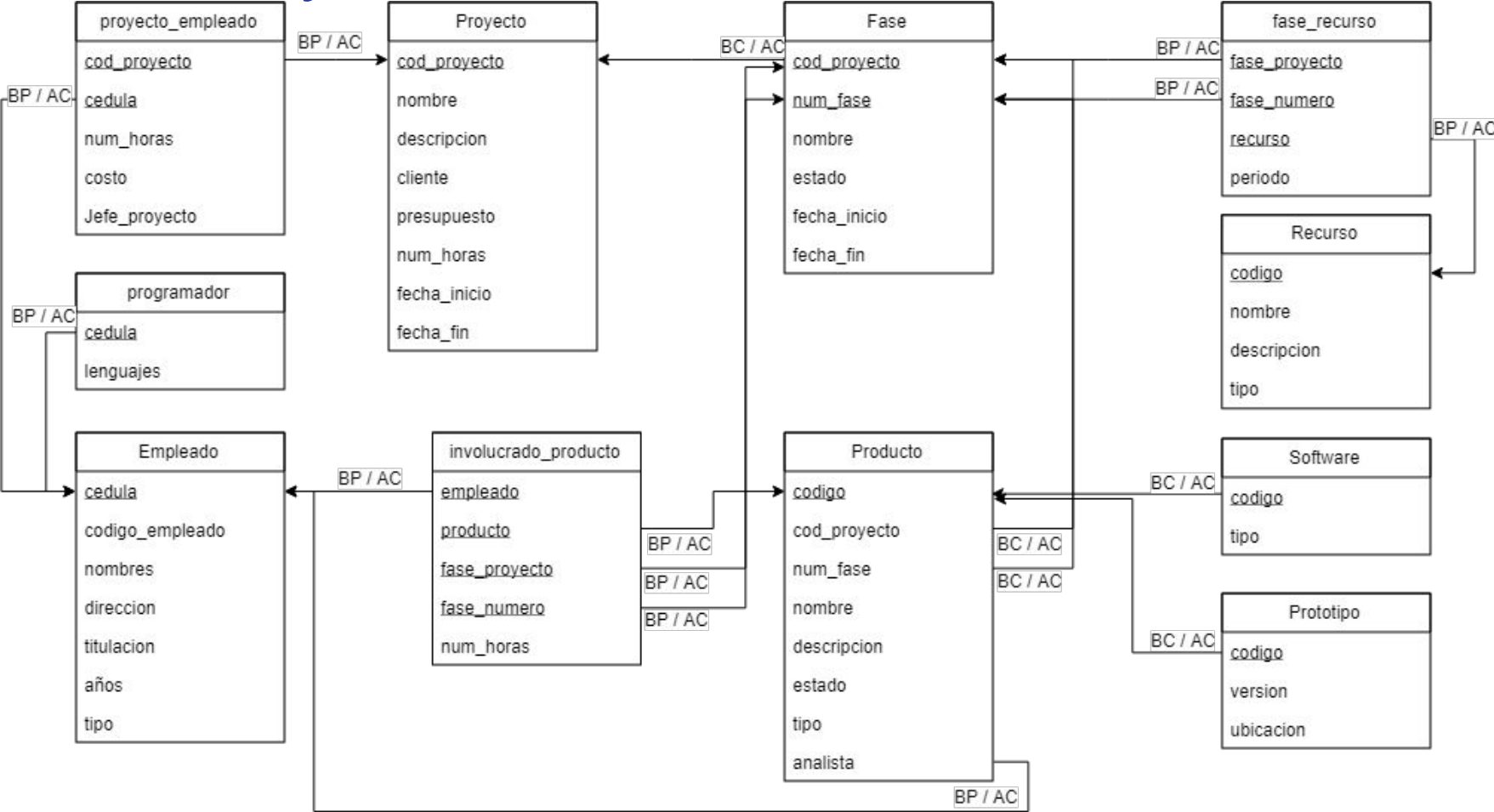
mysql -u root -h localhost -p < demo.sql

```
drop database demo;  
create database demo;  
use demo;  
create table productos (  
  parte varchar(20),  
  tipo varchar(20) ,  
  especificación varchar(20) ,  
  psugerido float(6,2),  
  clave int(3) zerofill not null auto_increment,  
  primary key (clave)  
);
```

Practicar

Gestión Proyectos de Software

Esquema Relacional



Lenguaje de Manipulación de Datos - DML

Modificación de base de datos

Inserción

- Especificar los datos de la tupla a insertar
- Atributos opcionales

```
insert into relacion  
values ('datoX', 'datY', datoZ)
```


Modificación de base de datos

Insertión (insert)

- Especificar los datos de la tupla a insertar
- Atributos opcionales
- Orden de los atributos

```
insert into relacion (atrX, atrY, atrZ)  
values ('datoX', 'datY', datoZ)
```

Modificación de base de datos

Inserción

Ejemplo insertar un nuevo cliente

```
insert into Cliente (cliente_id, nombres, telefono, ciudad, direccion)  
values (10, 'Ramiro Pereira', 3021025515, 'Piedecuesta', 'Calle 10')
```

Modificación de base de datos

Actualizaciones (update)

- Modificar uno más valores de una relación
- Opcional: Actualiza basado en condición

Update relacion

Set atrX='datoX', atrY='datoY'

Where condicion

Modificación de base de datos

Actualizaciones (update)

Ejemplo: Se requiere aplicar la ganancia anual del 5% a las cuentas

Update cuenta

Set saldo=saldo*0.05

Modificación de base de datos

Actualizaciones (update)

Ejemplo: Actualizar el teléfono y la dirección al cliente 1 así:

Nuevo teléfono=3002015412, nueva dirección= Carrera 25 #8-8

Update cliente

Set telefono=3002015412, direccion='Carrera 25 #8-8'

Where cliente_id=1

Modificación de base de datos

Actualizaciones (update)

Ejemplo: restar el capital en un 5% a los préstamos mayores al promedio general de préstamos.

```
Update prestamo
Set valor_prestado=valor_prestado-valor_prestado*0.05
Where valor_prestado > (
    Select avg(valor_prestado)
    Form prestamo
)
```

Modificación de base de datos

Borrado

- Borra tuplas completas
- Se expresan como consultas
- Si se borran todas las tuplas se omite el where

delete from r
where P

Modificación de base de datos

Borrado

Ejemplo: Borrar las cuentas con saldo igual a 0

```
delete  
from cuentas  
where saldo=0
```


Ejercicio

Se requiere modelar un sistema de información que gestione la información de una Aerolínea.



Ejercicio

Se quiere diseñar un sistema de gestión de información para almacenar datos sobre los casos llevados a cabo en un juzgado. Se debe tener en cuenta entidades como: Juez, Abogados (Defensor/Fiscal), acusado, veredicto, condena, tipo de condena (Intramural, domiciliaria, etc).



ejercicio

Una empresa de juguetes desea almacenar la información de su actividad. Cree un modelo entidad relación teniendo en cuenta que quiere almacenar datos de empleados como nombre, documento de identidad, dirección, teléfono, fecha de nacimiento, salario y cargo. Los empleados pueden:

- Arquitecto: de quien se requiere almacenar valor de las comisiones y cantidad de proyectos.
- Administrativo: de quienes se requiere saber el nivel jerárquico, la dependencia y número de subordinados.
- Ingeniero: quienes tienen una especialidad y un número de años de experiencia

Tenga en cuenta que un arquitecto puede desempeñar otro puesto: administrativo o ingeniero.

En Producción se requiere almacenar información de los juguetes como tipo (mecánico, digital, didáctico, etc), precio, peso. Al igual que su relación con sus partes, proveedores, público objetivo.