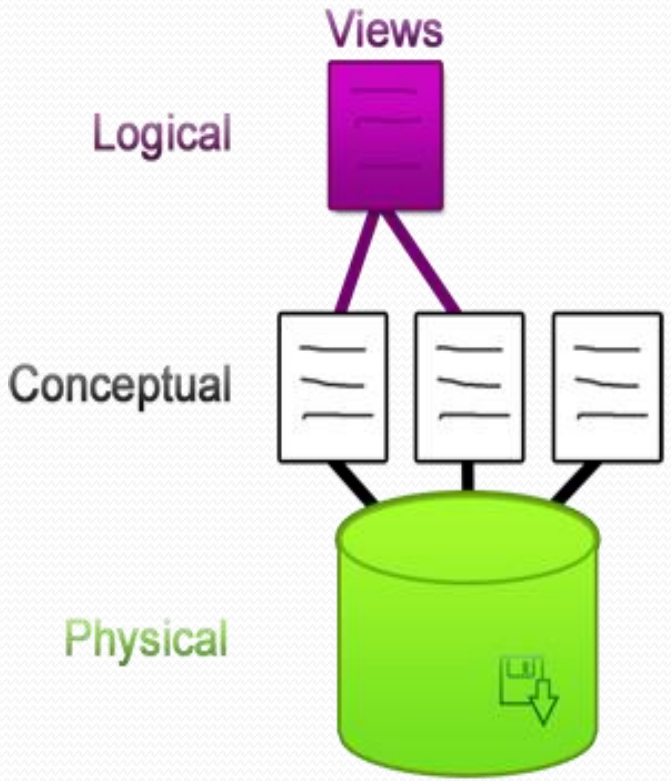
VISTAS

Niveles de abstracción



- **Nivel físico:** Es la implementación de las BD en dispositivos físicos. (Cómo se almacenan los datos)
- **Nivel conceptual (esquema lógico):** Es una abstracción del mundo real resultado del diseño. (Qué datos se almacenan y que relaciones existen)
- **Nivel lógico (vistas):** Permite a cada usuario percibir sólo las partes de la BD a las que tiene permiso.

VISTAS (VIEW)

- Una vista es una tabla virtual derivada de las tablas reales de una base de datos.
- Las vistas no se almacenan en la base de datos, sólo se almacena una definición de consulta, es decir, una vista contiene la instrucción SELECT necesaria para crearla.
- Resultado de la cual se produce una tabla cuyos datos proceden de la base de datos o de otras vistas.
- Si los datos de las tablas cambian, los de la vista que utiliza esos datos también cambia.
- Para definir una vista V, se especifica una consulta en SQL, a través de un conjunto de tablas existentes (R1, R2,…Rn).
- La vista V, se puede pensar como una tabla de los resultados de la consulta.

Usos de las vistas

- Las vistas se emplean para:
  - **Realizar consultas complejas más fácilmente:** Las vistas permiten dividir la consulta en varias partes.
  - **Proporcionar tablas con datos específicos:** Las vistas permiten ser utilizadas como tablas que resumen todos los datos.
  - **Modularidad de acceso a base de datos:** las vistas se pueden pensar en forma de módulos que da acceso a partes de la base de datos.

Comandos para vistas

CREATE VIEW : Define una tabla lógica a partir de una o más tablas físicas o de otras vistas.

CREATE OR REPLACE VIEW : Crea o reemplaza la definición de una vista en caso de que ya haya sido creada.

SHOW CREATE VIEW: Visualiza la forma en que se construyó la vista

DROP VIEW: Elimina una definición de vista (y cualquier vista definida a partir de ella). Ojo! no borra las tablas que se usan.

Sintaxis

CREATE VIEW nombre_vista **AS** consulta_SQL;

CREATE OR REPLACE VIEW nombre_vista **AS** consulta_SQL;

SHOW CREATE VIEW nombre_vista;

DROP VIEW nombre_vista;

Ejemplo 1: canciones con género e idioma

Habilitar la base de datos pixup

Mostrar las tablas de la base

Crear la vista:

```
CREATE OR REPLACE VIEW v_infoCanciones AS
SELECT c.id_cancion,c.nombre as cancion,duracion,i.nombre as
idioma,g.nombre as genero FROM disco_cancion JOIN (cancion c,idioma
i,genero g) USING(id_cancion,id_idioma,id_genero) order by 2;
```

Mostrar las tablas de la base

Consultar la tabla resultado

```
Select * from v_infoCanciones;
```

Ejemplo 2: discos con artistas

```
CREATE OR REPLACE VIEW v_infoDiscos AS
SELECT distinct id_disco,titulo AS disco,a.nombre AS
artista,id_disquera FROM disco_cancion JOIN (disco,artista a)
USING(id_disco,id_artista) order by 2;
```

Mostrar las tablas de la base

Consultar la tabla resultado

```
Select * from v_infoDiscos;
```

Ejemplo 3: discos con artistas y disqueras

Las vistas se pueden usar para consulta en JOIN como cualquier tabla

```
CREATE VIEW v_reporteDiscos AS
SELECT id_disco,disco,artista,nombre as disquera
FROM disquera
JOIN v_infoDiscos USING(id_disquera)
ORDER BY 2;
```

Mostrar las tablas de la base

Consultar la tabla resultado

```
Select * from v_reporteDiscos;
```

Más ejemplos

```
CREATE VIEW v_discosArtistas AS
select distinct id_disco,titulo as disco,id_artista,nombre as artista from disco
left join disco_cancion using(id_disco)
left join artista using(id_artista) order by 2;

CREATE VIEW v_artistasDiscos AS
select distinct id_disco,titulo as disco,id_artista,nombre as artista from disco
right join disco_cancion using(id_disco)
right join artista using(id_artista) order by 4;

CREATE VIEW v_infoCompletaFULL AS
SELECT * FROM v_artistasDiscos
UNION
SELECT * FROM v_discosArtistas;

SHOW tables;
Select * from v_infoCompletaFULL limit 20;
```

¿Preguntas?