

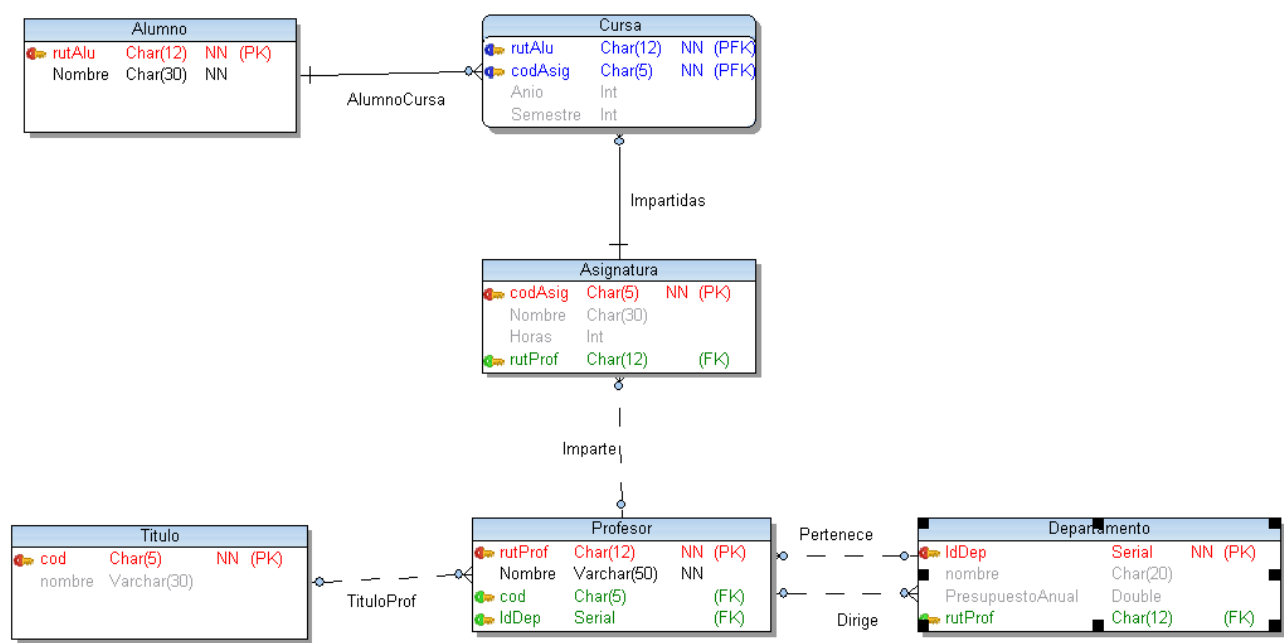
# LABORATORIO N°10

## VISTAS E ÍNDICES

OBJETIVO: DISEÑAR VISTAS E ÍNDICES PARA OPTIMIZA LA BASE DE DATOS

Actividades

- 1. Inicie MySQL y MySQL-Front.
- 2. Ejecute el siguiente Script Sql para crear la BD Universidad que se muestra en la figura;



-- Eliminar la BD Universida -----

drop database universidad;

-- Crear BD Universidad -----

create database universidad;

use universidad;

-- Crear Tablas -----

```
CREATE TABLE Alumno ( rutAlu Char(12) NOT NULL,  Nombre Char(30) NOT NULL );

ALTER TABLE Alumno ADD PRIMARY KEY (rutAlu);

CREATE TABLE Asignatura( codAsig Char(5) NOT NULL,  Nombre Char(30),  Horas Int,  rutProf Char(12));

ALTER TABLE Asignatura ADD PRIMARY KEY (codAsig);

CREATE TABLE Profesor( rutProf Char(12) NOT NULL,  Nombre Varchar(50) NOT NULL,  cod Char(5),  IdDep int);

ALTER TABLE Profesor ADD PRIMARY KEY (rutProf);

CREATE TABLE Cursa( rutAlu Char(12) NOT NULL,  codAsig Char(5) NOT NULL,  Anio Int,  Semestre Int);

ALTER TABLE Cursa ADD PRIMARY KEY (rutAlu,codAsig);

CREATE TABLE Titulo( cod Char(5) NOT NULL,  nombre Varchar(30));

ALTER TABLE Titulo ADD PRIMARY KEY (cod);

CREATE TABLE Departamento( IdDep int NOT NULL,  nombre Char(20),  PresupuestoAnual Double,  rutProf Char(12));
```

```
ALTER TABLE Departamento ADD PRIMARY KEY (IdDep);
```

```
ALTER TABLE Cursa ADD CONSTRAINT AlumnoCursa FOREIGN KEY (rutAlu) REFERENCES Alumno (rutAlu) ON DELETE NO ACTION ON UPDATE NO ACTION;
```

```
ALTER TABLE Cursa ADD CONSTRAINT Impartidas FOREIGN KEY (codAsig) REFERENCES Asignatura (codAsig) ON DELETE NO ACTION ON UPDATE NO ACTION;
```

```
ALTER TABLE Asignatura ADD CONSTRAINT Imparte FOREIGN KEY (rutProf) REFERENCES Profesor (rutProf) ON DELETE NO ACTION ON UPDATE NO ACTION;
```

```
ALTER TABLE Profesor ADD CONSTRAINT TituloProf FOREIGN KEY (cod) REFERENCES Titulo (cod) ON DELETE NO ACTION ON UPDATE NO ACTION;
```

```
ALTER TABLE Profesor ADD CONSTRAINT Pertenece FOREIGN KEY (IdDep) REFERENCES Departamento (IdDep) ON DELETE NO ACTION ON UPDATE NO ACTION;
```

### 3. Ejecute los siguientes SQL para insertar datos en la BD Universidad

-- Insertamos en la tabla Departamento -----

```
insert into departamento (idDep, nombre, presupuestoAnual, rutProf) values
```

```
(1, 'ingenieria', 100000000, '11111111-1');
```

```
insert into departamento (idDep, nombre, presupuestoAnual, rutProf) values
```

```
(2, 'sistemas', 50000000, '22222222-2');
```

```
insert into departamento (idDep, nombre, presupuestoAnual, rutProf) values
```

```
(3, 'matematica', 150000000, '33333333-3');
```

-- Insertamos en la tabla Título -----

```
insert into titulo (cod, nombre) values ('12345', 'ingeniero informatico');
```

```
insert into titulo (cod, nombre) values ('67890', 'mecanico');
```

-- Insertamos en la tabla Profesor -----

```
insert into profesor (rutProf, nombre, cod, IdDep) values
```

```
('11111111-1', 'juan perez', '12345', 1);
```

```
insert into profesor (rutProf, nombre, cod, IdDep) values
```

```
('121914856-5', 'jose carrasco', '12345', 1);
```

```
insert into profesor (rutProf, nombre, cod, IdDep) values
```

```
('10354869-7', 'manuel osess', '67890', 1);
```

```
insert into profesor (rutProf, nombre, cod, IdDep) values
```

```
('15487956-8', 'marco lopez', '67890', 2);
```

```
insert into profesor (rutProf, nombre, cod, IdDep) values
```

```
('167458852-9', 'anita bustamante', '12345', 2);
```

-- Insertamos en la tabla Alumno -----

```
insert into alumno (rutAlu, nombre) values ('15126954-8', 'margarita pinilla');
```

```
insert into alumno (rutAlu, nombre) values ('14852654-7', 'monica soto');
```

```
insert into alumno (rutAlu, nombre) values ('15785365-8', 'marco ferj');
```

```
insert into alumno (rutAlu, nombre) values ('14652985-4', 'manuel resto');
```

```
-- Insertamos en la tabla Asignatura -----
```

```
insert into asignatura (codAsig, nombre, horas, rutProf) values
```

```
( '11111', 'logistica', 6, '11111111-1');
```

```
insert into asignatura (codAsig, nombre, horas, rutProf) values
```

```
( '22222', 'calculo I', 6, '121914856-5');
```

```
insert into asignatura (codAsig, nombre, horas, rutProf) values
```

```
( '33333', 'fisica I', 4, '10354869-7');
```

```
-- Insertamos en la tabla Cursa -----
```

```
insert into cursa (rutAlu, codAsig, Anio, Semestre) values
```

```
('15126954-8', '11111', 2010, 1);
```

```
insert into cursa (rutAlu, codAsig, Anio, Semestre) values
```

```
('15126954-8', '22222', 2010, 1);
```

```
insert into cursa (rutAlu, codAsig, Anio, Semestre) values
```

```
('15126954-8', '33333', 2010, 2);
```

```
insert into cursa (rutAlu, codAsig, Anio, Semestre) values
```

```
('14852654-7', '11111', 2010, 1);
```

```
insert into cursa (rutAlu, codAsig, Anio, Semestre) values
```

```
('14852654-7', '22222', 2010, 2);
```

```
insert into cursa (rutAlu, codAsig, Anio, Semestre) values
```

```
('14852654-7', '33333', 2010, 2);
```

```
insert into cursa (rutAlu, codAsig, Anio, Semestre) values
```

```
('14652985-4', '11111', 2010, 1);
```

## VISTAS

Una **vista** de base de datos es un resultado de una consulta SQL de una o varias tablas; también se le puede considerar una tabla virtual.

Las vistas tienen la misma estructura que una tabla: filas y columnas. La única diferencia es que sólo se almacena de ellas la definición, no los datos. Los datos que se recuperan mediante una consulta a una vista se presentarán igual que los de una tabla. De hecho, si no se sabe que se está trabajando con una vista, nada hace suponer que es así. Al igual que sucede con una tabla, se pueden insertar, actualizar, borrar y seleccionar datos en una vista. Aunque siempre es posible seleccionar datos de una vista, en algunas condiciones existen restricciones para realizar el resto de las operaciones sobre vistas.

Una vista se especifica a través de una expresión de consulta (una sentencia SELECT) que la calcula y que puede realizarse sobre una o más tablas. Sobre un conjunto de tablas relacionales se puede trabajar con un número cualquiera de vistas.

La mayoría de los SGBD soportan la creación y manipulación de vistas.

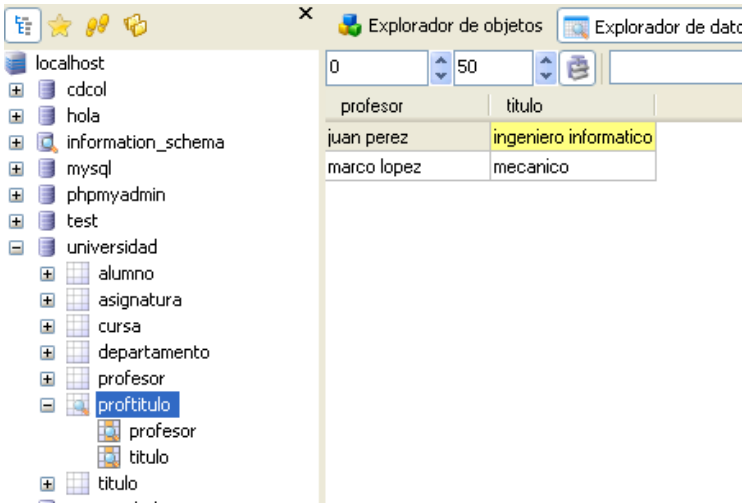
La sintaxis general para crear una vista es la siguiente:

**CREATE VIEW *nombre\_vista* [(*columnas*)] AS *sentencia\_select***

La **sentencia\_select** es una sentencia SELECT que proporciona la definición de la vista. Puede estar dirigida a tablas de la base o a otras vistas.

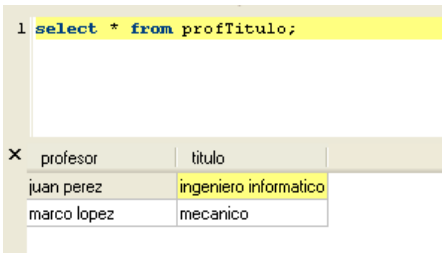
- ▶ Ejemplo: La siguiente consulta SQL crea una vista llamada `profTitulo` con el nombre del profesor y el nombre de su título
  - `create view profTitulo as select profesor.nombre as profesor, titulo.nombre as titulo from titulo inner join profesor on titulo.cod =profesor.cod;`

Se creará una nueva tabla en la BD Universidad con el nombre `profTitulo` y con dos atributos, `profesor` y `titulo`



A pesar de que `profTitulo` pareciera ser una tabla, realmente no contiene datos. Los datos que muestra están allí gracias a la consulta SQL creada. Es como crear una consulta SQL y guardar los datos para no estar ejecutando la consulta SQL cada vez que se requiere.

Para acceder a los datos en una vista, basta con ejecutar un SELECT “como si fuera una tabla más de la BD”.



4. Cree las siguientes vistas para la BD Universidad
- a) Crear una vista que contenga los nombres de los profesores, el nombre del depto en el que trabajan y el nombre del título profesional.
  - b) Crear una vista que almacene los datos de los alumnos, las asignaturas que cursa y el nombre del profesor que las imparten.
  - c) Crear una vista con los datos de las asignaturas y los nombres de los profesores que las imparten.

INDICES

- ▶ Un índice (o KEY, o INDEX) es un grupo de datos que asocia con una o varias columnas de la tabla. En este grupo de datos aparece la relación entre el contenido y el número de fila donde está ubicado.

1. Ejecutemos una consulta SQL utilizando “Explain” para revisar su ejecución.

**EXPLAIN SELECT \* FROM profesor WHERE nombre = ‘juan perez’;**

La consulta anterior mostrará la siguiente información:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	profesor	ALL					2	Using where

Observamos que no utiliza ningún índice para realizar la búsqueda.

2. Creemos un índice para el campo Nombre de la tabla Profesor, ya que se realizarán muchas búsquedas por este campo.

**create index buscaNombre on profesor (nombre);**

Nombre	Tipo	NULL	Por defecto	Extras	Coment:
Índice principal	rutProf			unique	
IdDep	IdDep			unique	
TituloProf	cod				
buscaNombre	Nombre				
rutProf	char(12)	No			
Nombre	varchar(50)	No			
cod	char(5)	Sí	<NULL>		
IdDep	bigint(20) unsigned	No	<auto_increme...		

Debe aparecer un ícono azul que muestra el nuevo índice.

3. Ahora ejecutemos nuevamente la consulta SQL anterior

**SELECT \* FROM profesor WHERE nombre =‘juan perez’;**

1 EXPLAIN SELECT * FROM profesor WHERE nombre = 'juan perez';									
id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	profesor	ref	buscaNombre	buscaNombre	52	const	1	Using where

Observamos que la consulta SQL busca el nombre del profesor utilizando el índice creado.

4. Cree los siguientes índices para la BS Universidad
- a. Un índice para el nombre del alumno
- b. Un índice para el nombre de la asignatura

Suba un archivo de Word con los pantallazos de la resolución de los ejercicios planteados