

**UNIVERSIDAD TECNOLÓGICA DE PANAMÁ**  
**FACULTAD DE INGENIERÍA DE SISTEMAS**  
**COMPUTACIONALES**  
**DEPARTAMENTO DE SISTEMAS DE INFORMACIÓN,**  
**CONTROL Y EVALUACIÓN DE RECURSOS**  
**INFORMÁTICOS**

**LIC. EN ING. DE SISTEMAS DE INFORMACIÓN**

**SISTEMAS DE BASES DE DATOS**

**Laboratorio No. 3**

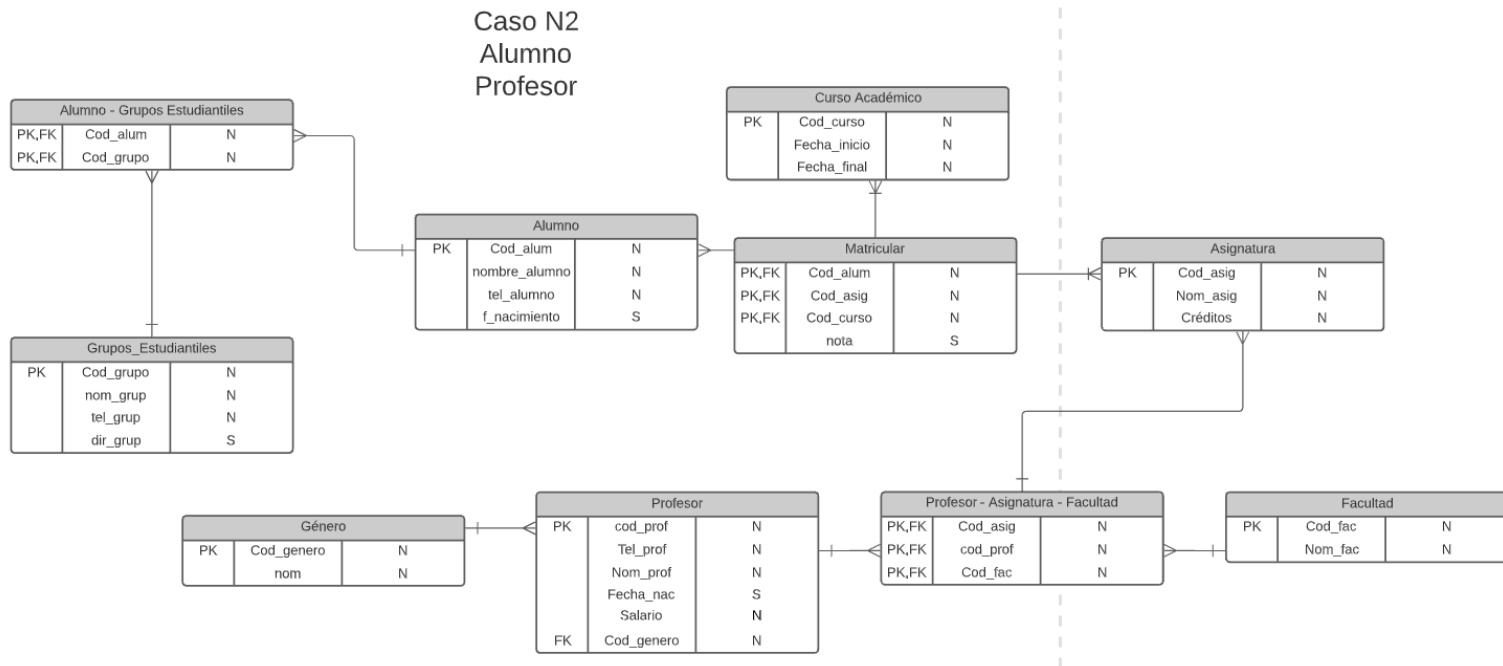
**Profesor:**  
**Ing. Henry Lezcano**

**Integrantes:**

**Cutire, Fernando (8-972-906)**  
**Díaz, Gabriel (20-53-5198)**  
**Escobar, Jorge (2-747-1772)**  
**Feng, William (8-977-446)**

## Caso No. 2 Alumno-Asignatura-Profesor

### Normalización



### Implementación física

Crear tabla CURSO\_ACAD

```
SQL> CREATE TABLE CURSO_ACAD (cod_curso number PRIMARY KEY not null,
2   f_inicio date not null,
3   f_final date not null
4 );
```

Crear tabla ALUMNO

```
SQL> CREATE TABLE ALUMNO (cod_alum number PRIMARY KEY not null,
2   nom_al varchar2(20) not null,
3   tel_al varchar2(20) not null,
4   f_nac date not null
5 );
```

Table created.

Crear tabla GENERO

```
SQL> CREATE TABLE GENERO (cod_genero number PRIMARY KEY not null,  
2     nom_genero varchar(20) not null  
3 );
```

Table created.

Crear tabla PROFESOR

```
SQL> CREATE TABLE PROFESOR (cod_prof number PRIMARY KEY not null,  
2     nom_prof varchar2(20) not null,  
3     tel_prof varchar2(20) not null,  
4     f_nac date not null,  
5     especialidad varchar2(20) not null,  
6     salario number(8,2) default 0,  
7     cod_genero number not null,  
8     CONSTRAINT fk_genero FOREIGN KEY (cod_genero) REFERENCES GENERO (cod_genero)  
9 );
```

Table created.

Crear tabla ASIGNATURA

```
SQL> CREATE TABLE ASIGNATURA (cod_asig number PRIMARY KEY not null,  
2     nom_asig varchar2(20) not null,  
3     creditos number not null  
4 );
```

Table created.

Crear tabla FACULTAD

```
SQL> CREATE TABLE FACULTAD (cod_fac number PRIMARY KEY not null,  
2     nom_fac varchar2(20) not null  
3 );
```

Table created.

### Crear tabla PROFESOR\_ASIGNATURA\_FACULTAD

```
SQL> CREATE TABLE PROFESOR_ASIGNATURA_FACULTAD (cod_asig number not null,  
2     cod_prof number not null,  
3     cod_fac number not null,  
4     CONSTRAINT fk_paf PRIMARY KEY (cod_asig, cod_prof, cod_fac),  
5     CONSTRAINT fk_prof FOREIGN KEY (cod_prof) REFERENCES PROFESOR (cod_prof),  
6     CONSTRAINT fk_asig FOREIGN KEY (cod_asig) REFERENCES ASIGNATURA (cod_asig),  
7     CONSTRAINT fk_fac FOREIGN KEY (cod_fac) REFERENCES FACULTAD (cod_fac)  
8 );
```

Table created.

### Crear tabla MATRICULA

```
SQL> CREATE TABLE MATRICULA (cod_asig number not null,  
2     cod_curso number not null,  
3     cod_alum number not null,  
4     nota number not null,  
5     CONSTRAINT fk_matr PRIMARY KEY (cod_alum, cod_asig, cod_curso),  
6     CONSTRAINT fk_asig_m FOREIGN KEY (cod_asig) REFERENCES ASIGNATURA (cod_asig),  
7     CONSTRAINT fk_curso_m FOREIGN KEY (cod_curso) REFERENCES CURSO_ACAD (cod_curso),  
8     CONSTRAINT fk_alum_m FOREIGN KEY (cod_alum) REFERENCES ALUMNO (cod_alum)  
9 );
```

Table created.

### Crear tabla GRUPOS\_ESTUDIANTILES

```
SQL> CREATE TABLE GRUPOS_ESTUDIANTILES (  
2     cod_grupo number PRIMARY KEY not null,  
3     nom_grupo varchar2(20) not null,  
4     tel_grupo varchar(20) not null,  
5     dir_grupo varchar(20) not null  
6 );
```

Table created.

## Crear tabla ALUMNOS\_GRUPOS\_ESTUDIANTILES

```
SQL> CREATE TABLE ALUMNO_GRUPOS_ESTUDIANTILES (cod_alum number not null,  
2     cod_grupo number not null,  
3     /* AGE = ALUMNO-GRUPOS_ESTUDIANTILES*/  
4     CONSTRAINT fk_agre PRIMARY KEY (cod_alum,cod_grupo),  
5     CONSTRAINT fk_asig_e FOREIGN KEY (cod_alum) REFERENCES ALUMNO (cod_alum),  
6     CONSTRAINT fk_grupo FOREIGN KEY (cod_grupo) REFERENCES GRUPOS_ESTUDIANTILES (cod_grupo)  
7 );  
  
Table created.
```

## Insertar datos a la tabla CURSO\_ACAD

```
SQL> INSERT INTO CURSO_ACAD VALUES(1, '10-mar-2021', '17-jul-2021');  
  
1 row created.
```

## Insertar datos a la tabla ALUMNO

```
SQL> INSERT INTO ALUMNO VALUES(1, 'Fernando', '6454-2367', '12-jul-2001');  
  
1 row created.
```

## Insertar datos a la tabla GENERO

```
SQL> INSERT INTO GENERO VALUES(0,'masculino');  
  
1 row created.  
  
SQL> INSERT INTO GENERO VALUES(1,'femenino');  
  
1 row created.
```

## Insertar datos a la tabla PROFESOR

```
SQL> INSERT INTO PROFESOR VALUES(1, 'Ariel Reluz', '8888-4444', '15-mar-2000', 'Ciberseguridad', 6000.00, 1);  
  
1 row created.
```

## Insertar datos a la tabla PROFESOR

```
SQL> INSERT INTO PROFESOR VALUES(2, 'Alexander Cañate', '8912-9876', '20-mar-1987', 'Cadena de Suministro', 5000.00, 0);  
  
1 row created.
```

Insertar datos a la tabla ASIGNATURA

```
SQL> INSERT INTO ASIGNATURA VALUES(1, 'Base de Datos', 4);  
  
1 row created.  
  
SQL> INSERT INTO ASIGNATURA VALUES(2, 'Base de Datos II', 4);  
  
1 row created.
```

Insertar datos a la tabla Facultad

```
SQL> INSERT INTO FACULTAD VALUES(1, 'FISC');  
  
1 row created.
```

Insertar datos a la tabla PROFESOR\_ASIGNATURA\_FACULTAD

```
SQL> INSERT INTO PROFESOR_ASIGNATURA_FACULTAD VALUES(1,1,1);  
  
1 row created.
```

Insertar datos a la tabla GRUPOS\_ESTUDIANTILES

```
SQL> INSERT INTO GRUPOS_ESTUDIANTILES VALUES (1, 'IEEE CS', '6879-2156', 'Prof. Nilda Yanguéz');  
  
1 row created.
```

Insertar datos a la tabla ALUMNO\_GRUPOS\_ESTUDIANTILES

```
SQL> INSERT INTO ALUMNO_GRUPOS_ESTUDIANTILES VALUES (1, 1);  
  
1 row created.
```

Crear vista VISTA\_FACULTAD

```
SQL> CREATE VIEW VISTA_FACULTAD AS
  2     SELECT f.nom_fac, p.nom_prof, p.salario, a.nom_asig
  3     FROM FACULTAD f, PROFESOR p, ASIGNATURA a;

View created.
```

Select para ver el contenido vista\_facultad

```
SQL> select * from vista_facultad;
```

NOM_FAC	NOM_PROF	SALARIO	NOM_ASIG
FISC	Ariel Reluz	6000	Base de Datos
FISC	Ariel Reluz	6000	Base de Datos II
FISC	Alexander Cañate	5000	Base de Datos
FISC	Alexander Cañate	5000	Base de Datos II

Crear tabla LABORATORIOS

```
SQL> CREATE TABLE LABORATORIOS (cod_lab number PRIMARY KEY not null,
  2     cod_asig number not null,
  3     cod_prof number not null,
  4     CONSTRAINT fk_prof_encargado FOREIGN KEY (cod_prof) REFERENCES PROFESOR (cod_prof),
  5     CONSTRAINT fk_asig_designada FOREIGN KEY (cod_asig) REFERENCES ASIGNATURA (cod_asig)
  6 );
```

Table created.

Eliminar tabla LABORATORIOS

```
SQL> drop table laboratorios;
```

Table dropped.

Update profesor

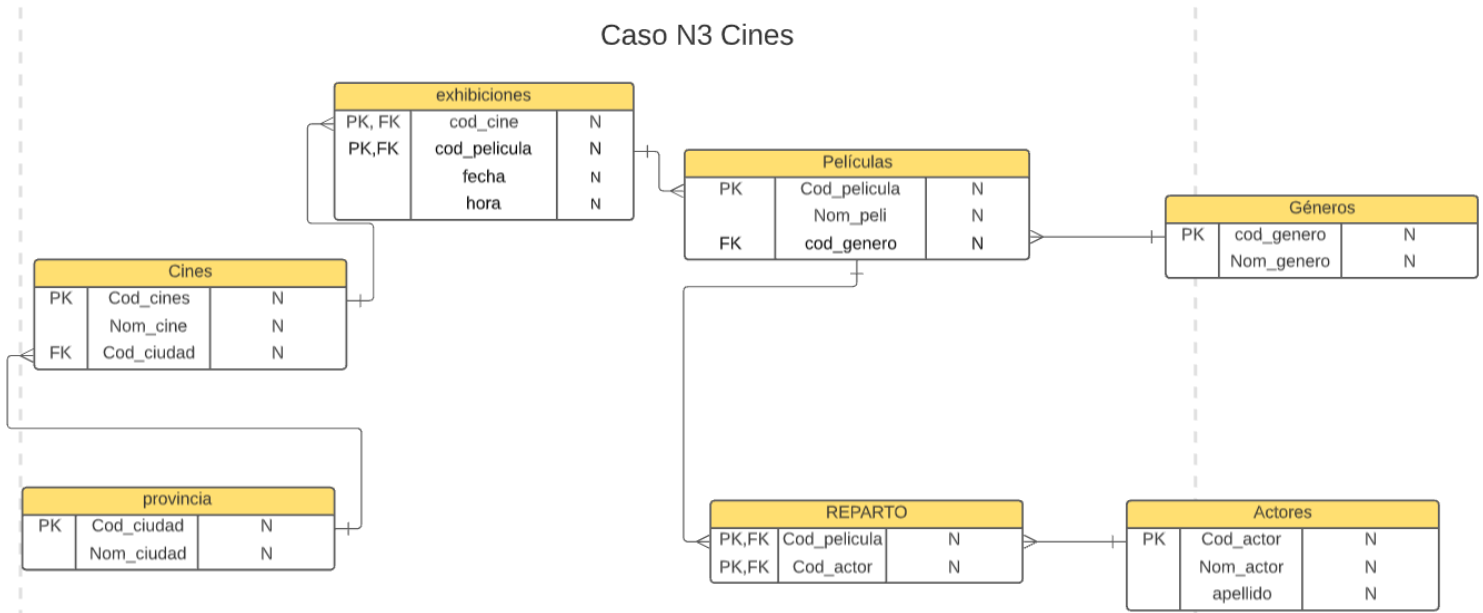
```
SQL> update profesor
  2  set salario = 7000
  3  where cod_prof = 2
  4  ;

1 row updated.
```



## Caso No.3 Cine-Película

### Normalización



### Implementación física

#### Crear tabla de actores

```
SQL> CREATE TABLE ACTORES (cod_actor number not null primary key,
2 actor_nombre varchar2(25) not null,
3 actor_apellido varchar2(25) not null
4 );
```

Table created.

#### Crear tabla de generos de peliculas

```
SQL> CREATE TABLE GENEROS (cod_genero number not null primary key,
2 genero_nombre varchar2(50) not null
3 );
```

Table created.

### Crear tabla de Peliculas

```
SQL> CREATE TABLE PELICULAS (cod_pelicula number not null primary key,  
2  pelicula_nombre varchar2(100) not null,  
3  cod_genero number,  
4  constraint fk_cod_genero foreign key (cod_genero) references GENEROS (cod_genero)  
5  );
```

Table created.

### Crear tabla del reparto

```
SQL> CREATE TABLE REPARTO (cod_pelicula number not null,  
2  cod_actor number not null,  
3  constraint cod_reparto_pk primary key (cod_pelicula, cod_actor),  
4  constraint fk_cod_pelicula foreign key (cod_pelicula) references PELICULAS (cod_pelicula),  
5  constraint fk_cod_actor foreign key (cod_actor) references ACTORES (cod_actor)  
6  );
```

Table created.

### Crear tabla de ciudades

```
SQL> CREATE TABLE CIUDADES (cod_ciudad number not null primary key,  
2  ciudad_nombre varchar2(100) not null  
3  );
```

Table created.

### Crear tabla de cines

```
SQL> CREATE TABLE CINES (cod_cine number not null primary key,  
2  cine_nombre varchar(100) not null,  
3  cod_ciudad number,  
4  constraint fk_cod_ciudad foreign key (cod_ciudad) references CIUDADES (cod_ciudad)  
5  );
```

Table created.

### Crear tabla de Exhibiciones

```
SQL> CREATE TABLE EXHIBICIONES (cod_cine number not null,
2  cod_pelicula number not null,
3  fecha date not null,
4  hora varchar2(5) not null,
5  constraint cod_exhibicion_pk primary key (cod_cine, cod_pelicula),
6  constraint fk_cod_cine foreign key (cod_cine) references CINES (cod_cine),
7  constraint fk_fk_pelicula_ex foreign key (cod_pelicula) references PELICULAS (cod_pelicula)
8  );
```

Table created.

## Inserciones

Insertar datos a la tabla de Géneros

```
SQL> INSERT INTO GENEROS(cod_genero, genero_nombre) VALUES(1, 'accion');
1 row created.

SQL> INSERT INTO GENEROS(cod_genero, genero_nombre) VALUES(2, 'romance');
1 row created.

SQL> INSERT INTO GENEROS(cod_genero, genero_nombre) VALUES(3, 'documental');
1 row created.

SQL> INSERT INTO GENEROS(cod_genero, genero_nombre) VALUES(4, 'comedia');
1 row created.

SQL> INSERT INTO GENEROS(cod_genero, genero_nombre) VALUES(5, 'drama');
1 row created.

SQL> select * from GENEROS;
```

COD_GENERO	GENERO_NOMBRE
1	accion
2	romance
3	documental
4	comedia
5	drama

## Insertar datos a la tabla de Actores

```
SQL> INSERT INTO ACTORES(Cod_actor, actor_nombre, actor_apellido) VALUES(1, 'fernando', 'cutire');
1 row created.

SQL> INSERT INTO ACTORES(Cod_actor, actor_nombre, actor_apellido) VALUES(2, 'gabriel', 'diaz');
1 row created.

SQL> INSERT INTO ACTORES(Cod_actor, actor_nombre, actor_apellido) VALUES(3, 'jorge', 'escobar');
1 row created.

SQL> INSERT INTO ACTORES(Cod_actor, actor_nombre, actor_apellido) VALUES(4, 'william', 'feng');
1 row created.

SQL> select * from actores;
```

COD_ACTOR	ACTOR_NOMBRE	ACTOR_APELLIDO
1	fernando	cutire
2	gabriel	diaz
3	jorge	escobar
4	william	feng

```
SQL> _
```

## Insertar datos a la tabla de Ciudades

```
SQL> INSERT INTO CIUDADES (cod_ciudad, ciudad_nombre) VALUES(1, 'ciudad de panama');
1 row created.

SQL> INSERT INTO CIUDADES (cod_ciudad, ciudad_nombre) VALUES(2, 'david');
1 row created.

SQL> INSERT INTO CIUDADES (cod_ciudad, ciudad_nombre) VALUES(3, 'colon');
1 row created.

SQL> INSERT INTO CIUDADES (cod_ciudad, ciudad_nombre) VALUES(4, 'santiago');
1 row created.

SQL> INSERT INTO CIUDADES (cod_ciudad, ciudad_nombre) VALUES(5, 'chitre');
1 row created.

SQL> INSERT INTO CIUDADES (cod_ciudad, ciudad_nombre) VALUES(6, 'PENONOME');
1 row created.
```

```
SQL> select * from ciudades;
```

```
COD_CIUADAD
```

```
-----
```

```
CIUDAD_NOMBRE
```

```
-----
```

```
          1  
ciudad de panama
```

```
          2  
david
```

```
          3  
colon
```

```
COD_CIUADAD
```

```
-----
```

```
CIUDAD_NOMBRE
```

```
-----
```

```
          4  
santiago
```

```
          5  
chitre
```

```
          6  
PENONOME
```

```
6 rows selected.
```

## Insertar datos a la tabla de Peliculas

```
SQL> INSERT INTO PELICULAS (cod_pelicula, pelicula_nombre, cod_genero) VALUES(1, 'cutire y las ardillas', 4);
```

```
1 row created.
```

```
SQL> INSERT INTO PELICULAS (cod_pelicula, pelicula_nombre, cod_genero) VALUES(2, 'el regreso de escobar', 1);
```

```
1 row created.
```

```
SQL> INSERT INTO PELICULAS (cod_pelicula, pelicula_nombre, cod_genero) VALUES(3, 'feng man' 5, 1);
```

```
INSERT INTO PELICULAS (cod_pelicula, pelicula_nombre, cod_genero) VALUES(3, 'feng man' 5, 1)
```

```
ERROR at line 1:  
ORA-00917: missing comma
```

```
SQL> INSERT INTO PELICULAS (cod_pelicula, pelicula_nombre, cod_genero) VALUES(3, 'feng man 5', 1);
```

```
1 row created.
```

```
SQL> INSERT INTO PELICULAS (cod_pelicula, pelicula_nombre, cod_genero) VALUES(4, 'gabox 3' , 3);
```

```
1 row created.
```

```

COD_PELICULA
-----
PELICULA_NOMBRE
-----
COD_GENERO
-----
          1
cutire y las ardillas
          4

          2
el regreso de escobar
          1

COD_PELICULA
-----
PELICULA_NOMBRE
-----
COD_GENERO
-----

          3
feng man 5
          1

          4
gabox 3

COD_PELICULA
-----
PELICULA_NOMBRE
-----
COD_GENERO
-----

          3

```

Insertar datos a la tabla de Cines

```
SQL> INSERT INTO CINES VALUES(1, 'CUTIROPOLIS 1', 1);
```

```
1 row created.
```

```
SQL> INSERT INTO CINES VALUES(2, 'CUTIROPOLIS 2', 3);
```

```
1 row created.
```

```
SQL> INSERT INTO CINES VALUES(3, 'ESCOMARK', 6);
```

```
1 row created.
```

```
SQL> select * from cines;
```

```
    COD_CINE
```

```
-----  
CINE_NOMBRE
```

```
-----  
COD_CIUADAD
```

```
-----  
          1  
CUTIROPOLIS 1  
          1
```

```
          2  
CUTIROPOLIS 2  
          3
```

```
    COD_CINE
```

```
-----  
CINE_NOMBRE
```

```
-----  
COD_CIUADAD
```

```
-----  
          3  
ESCOMARK  
          6
```

Insertar datos a la tabla de Exhibiciones



```
SQL> INSERT INTO EXHIBICIONES VALUES (1, 2, '01-jun-2021','08:00');
1 row created.

SQL> INSERT INTO EXHIBICIONES VALUES (2, 2, '01-jun-2021','08:00');
1 row created.

SQL> INSERT INTO EXHIBICIONES VALUES (2, 3, '01-jun-2021','09:00');
1 row created.

SQL> INSERT INTO EXHIBICIONES VALUES (3, 2, '04-jun-2021','20:00');
1 row created.
```

```
SQL> select * from EXHIBICIONES;
```

COD_CINE	COD_PELICULA	FECHA	HORA
1	2	01/06/21	08:00
2	2	01/06/21	08:00
2	3	01/06/21	09:00
3	2	04/06/21	20:00

Eliminar la la columna de categorías de la tabla de películas:

```
SQL> desc peliculas;
```

Name	Null?	Type
COD_PELICULA	NOT NULL	NUMBER
PELICULA_NOMBRE	NOT NULL	VARCHAR2(100)
COD_GENERO		NUMBER
CATEGORIA		VARCHAR2(45)

```
SQL> ALTER TABLE PELICULAS
2 DROP COLUMN categoria;
```

```
Table altered.
```

```
SQL>
```

Creando las vistas:

Seleccionando de VISTA\_CARTELERA\_CINE\_1

COD_CINE	CINE_NOMBRE	COD_CIU DAD	COD_CINE	COD_PELICULA	FECHA	HORA
1	CUTIROPOLIS	1	1	1	30-JUN-21	10:00 am
1	CUTIROPOLIS	1	1	1	30-JUN-21	01:00 pm
1	CUTIROPOLIS	1	1	3	01-JUL-21	07:00 pm
1	CUTIROPOLIS	1	1	2	01-JUL-21	09:00 pm

SQL > █

Ver las películas, disponibles en el cine con el id = 1, con su nombre, fecha y hora de función.