

## UNIDAD 6: INSERT, UPDATE, DELETE

0.- Como paso previo se hará una copia de las tablas que intervengan en las operaciones de manipulación de datos que se realizarán a continuación.

```
CREATE TABLE alumnos1 like alumnos;  
INSERT INTO alumnos1 SELECT * FROM alumnos;  
CREATE TABLE alumnomodulo1 like alumnomodulo;  
INSERT INTO alumnomodulo1 SELECT * FROM alumnomodulo;
```

1.- Crea la tabla alumnogrupos con los campos: id\_alumno, id\_ciclo, turno (varchar(1)). Inserta a partir de una consulta que obtenga el id\_alumno y el id\_ciclo desde alumnomodulos sin que las filas se repitan. El turno lo ponemos a null en la propia select.

```
CREATE TABLE `alumnogrupos` (  
  `id_alumno` INT NOT NULL,  
  `id_ciclo` INT NOT NULL,  
  `turno` VARCHAR(1)  
ENGINE=InnoDB;  
  
INSERT INTO alumnogrupos SELECT distinct id_alumno, id_ciclo, null from alumnomodulo;  
mysql> INSERT INTO alumnogrupos (id_alumno,id_ciclo,turno) SELECT distinct  
id_alumno, id_ciclo, null from alumnomodulo;  
Records: 127 Duplicates: 0 Warnings: 0
```

Otra forma es:

```
CREATE TABLE alumnogrupos2 as (SELECT distinct id_alumno, id_ciclo from  
alumnomodulo);  
  
alter table alumnogrupos2 add turno varchar(1);
```

```
CREATE TABLE `alumnogrupos3` as (SELECT distinct id_alumno, id_ciclo, null  
"turno" from alumnomodulo);  
  
ALTER TABLE `alumnogrupos3` MODIFY COLUMN `turno` VARCHAR(1) NULL  
DEFAULT NULL AFTER `id_ciclo`;
```

!!!RECORDAD QUE EN MYSQL SE PUEDE USAR CHANGE AUNQUE EN SQL ESTANDAR SEA MODIFY !!!

```
ALTER TABLE alumnogrupos3 CHANGE COLUMN turno turno VARCHAR(1) NULL  
DEFAULT NULL AFTER id_ciclo;
```

2.- Añadir a la tabla anterior la clave primaria: id\_alumno e id\_ciclo :

```
ALTER TABLE alumnogrupos ADD CONSTRAINT `PK_alumnogrupos` PRIMARY KEY  
(id_alumno,id_ciclo);
```

3.- Añadir a la tabla anterior que id\_alumno es foránea de alumno (se borra en cascada) y que id\_ciclo es foránea de ciclo

```
ALTER TABLE alumnogrupos ADD CONSTRAINT FK1_alumnogrupos_ciclo FOREIGN KEY (id_ciclo) REFERENCES ciclos (id_ciclo) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE alumnogrupos ADD CONSTRAINT FK2_alumnogrupos_alumno FOREIGN KEY (id_alumno) REFERENCES alumnos (id_alumno) ON DELETE CASCADE ON UPDATE CASCADE;
```

\*NOTA: Si hubiera que borrar la restricción:

```
alter table alumnogrupos drop foreign key FK1_alumnogrupos_ciclo;
```

4.- Actualizar los turnos de los alumnos del ciclo 1 a "m". Otra instrucción para los del ciclo 2 a la tarde "t". Y otra más que actualice los alumnos del ciclo 3 a mañana si id\_alumno es impar. Los que quedan que vayan a la tarde (son del ciclo 3 con id\_alumno par)

```
UPDATE alumnogrupos SET TURNO="M" WHERE IDCICLO=1;
```

```
UPDATE alumnogrupos SET TURNO="T" WHERE IDCICLO=2;
```

```
UPDATE alumnogrupos SET TURNO="M" WHERE IDCICLO=3 AND MOD(IDALUMNO,2)=1;
```

```
UPDATE alumnogrupos SET TURNO="T" WHERE IDCICLO=3 AND IDALUMNO%2=0;
```

5.- Visualizar cuántos alumnos hay por ciclo:

```
select idciclo, count(*) from alumnogrupos group by idciclo;
```

	1		32	
	2		33	
	3		62	

5b.- Visualizar el id\_alumno y en cuántos ciclos está matriculado.

```
select idalumno, count(idciclo) from alumnogrupos group by idalumno;
```

6.- Buscar los alumnos que tienen más de un ciclo en alumnogrupos e ir borrándolos de alumno. Deberían borrarse de alumnomodulo y alumnogrupos automáticamente. Si no está en on delete cascade en alumnomodulo actualizar la foreign key para que así sea.

```
mysql> alter table alumnomodulo drop foreign key  
fk_alumno_alumnomodulo;
```

Query OK, 0 rows affected (0,06 sec)

Records: 0 Duplicates: 0 Warnings: 0

```
mysql> alter table alumnomodulo add constraint  
fk_alumno_alumnomodulo FOREIGN KEY(idalumno) REFERENCES  
alumnos(id_alumno) ON DELETE CASCADE ON UPDATE CASCADE;
```

Query OK, 537 rows affected (0,52 sec)

Records: 537 Duplicates: 0 Warnings: 0

Estos serían los alumnos con más de un ciclo:

```
SELECT IDALUMNO, COUNT(IDCICLO) FROM alumnogrupo GROUP BY IDALUMNO  
HAVING COUNT(IDCICLO)>1;
```

Estos serían los alumnos con más de un ciclo y sus nombres:

```
SELECT IDALUMNO,NOMBRE,count(idciclo) FROM alumnogrupo ag, alumnos a  
where ag.idalumno=a.id_alumno GROUP BY IDALUMNO HAVING COUNT(IDCICLO)>1;
```

Ahora para borrarlos con la select anterior, se usa un DELETE para borrar sólo a esos alumnos:

```
delete from alumnos where id_alumno IN(SELECT IDALUMNO FROM alumnogrupo  
GROUP BY IDALUMNO HAVING COUNT(IDCICLO)>1);
```

NOTA: Revisad que la foreign key idalumno en alumnomodulo tenga la restricción on delete cascade, de no ser no se permitirá el borrado de dichos alumnos en la tabla padre 'alumnos'. Modificar la restricción para que se permita borrar en cascada todos los registros que tengan ese idalumno en la tabla alumnomodulo y en la alumnogrupo.

Atención ¡! La subselect SÓLO PUEDE DEVOLVER UN CAMPO para poder hacer la comparación. No hemos puesto =, hemos puesto IN porque nos devuelve más de un id\_alumno.

7.0.- Calcular cuántos módulos da cada profesor:

```
select id_profesor, count(id_modulo) from profesoresmodulos group by id_profesor;
```

7.- Crear la tabla tutores con id\_profesor, ciclo, turno (estos dos últimos a null), con los profesores que dan 2 ó 3 módulos.

1/ Crear la tabla tutores solo con los id\_profesor que impartan dos o tres módulos:

```
create table tutores as select id_profesor from profesoresmodulos group by  
id_profesor having count(id_modulo)=2 or count(id_modulo)=3;
```

ojo a lo de subbusqueda no se puede editar la misma tabla en la que se busca

2/ Añadir los campos ciclo y turno para que sean entero y varchar(1) respectivamente.

```
Alter table tutores add id_ciclo int, add turno varchar(1);
```

También, sustituyendo a lo anterior :

```
create table tutores as select id_profesor, null 'ciclo',null 'turno' from profesoresmodulos group by id_profesor having count(id_modulo) =2 or count(id_modulo)=3;
```

```
alter table tutores modify ciclo int;
```

```
alter table tutores modify turno char;
```

3/ Actualizar el ciclo y el turno de forma que queden todos cubierto (DAW:m; DAM:t, ASIR: t)

```
update tutores set id_ciclo=1,turno='m' where id_profesor=1;
```

```
update tutores set id_ciclo=2,turno='t' where id_profesor=3;
```

```
update tutores set id_ciclo=3,turno='t' where id_profesor=14;
```

5/ Añadir la clave primaria:

```
Alter table tutores add constraint PK_TUTORES PRIMARY KEY (id_profesor);
```

Otra forma:

Se puede utilizar la select para visualizar la tabla:

```
SELECT id_profesor, null "ciclo", null "turno" FROM profesoresmodulos  
GROUP BY id_profesor  
HAVING COUNT(*) IN (2,3);
```

Después crear la tabla y modificar los campos:

```
CREATE TABLE TUTORES  
AS  
SELECT id_profesor, null "ciclo", null "turno"  
FROM profesoresmodulos  
GROUP BY id_profesor  
HAVING COUNT(*) IN (2,3);
```

```
ALTER TABLE TUTORES MODIFY ciclo INT;
```

```
ALTER TABLE TUTORES MODIFY turno VARCHAR(1);
```

Las actualizaciones Se pueden hacer como queráis, ir dando los valores registro a registro:

```
UPDATE TUTORES SET CICLO=1, TURNO="M" WHERE id_profesor =1;
```

```
UPDATE TUTORES SET CICLO=2 , TURNO="T" WHERE ID_PROFESOR =3;
```

```
UPDATE TUTORES SET CICLO=3 , TURNO="T" WHERE ID_PROFESOR =14;
```