

Módulo:	Bases de Datos
UT05:	Tratamiento de datos
Actividad de desarrollo:	Transacciones

1. Objetivo general

Insertar, modificar y eliminar registros en tablas MySQL mediante el uso de transacciones.

2. Metodología y material.

Máquina virtual con el sistema operativo Ubuntu y MySQL, Workbench y/o phpMyAdmin instalado. Consultar apuntes de la asignatura en el campus.

Para la realización de la práctica es necesario tener la base de datos NBA cargada con todos sus registros.

IMPORTANTE: Si se detectan dos prácticas iguales se anularán ambas sin posibilidad de cambiar su nota en esta práctica.

3. Descripción:

Usando MySQL Workbench crea un script que realice los siguientes puntos.

(Pega debajo de cada punto el código utilizado)

1. Comprueba que tienes desactivada la confirmación automática

```
set autocommit=0;
```

- Introduce en la tabla equipos, un nuevo equipo con nombre The Torpedors, ciudad Las Vegas conferencia oeste y division Pacific. Comprueba que la fila se ha insertado correctamente en la tabla, si es así, confirma la operación.

```
insert into equipos (Nombre,Ciudad,Conferencia,Division) Values ("The torpedors",  
"Las Vegas","West","Pacific");
```

```
commit;
```

- Introduce en la tabla jugadores un nuevo jugador con el nombre Luka Doncic, el peso 224, la altura 6-7, la posición G, el código 699, procedencia Serbia y equipo Bulls. No confirmes los cambios

```
insert into jugadores (codigo,Nombre,Altura,Peso,Posicion,Procedencia,Nombre_equipo)  
values (699,"Luka Doncic","6-7",224,"G","Serbia","Bulls");
```

- Aborta la operación anterior y comprueba que la fila ha desaparecido de la tabla.

```
rollback;
```

- Introduce en la tabla jugadores un nuevo jugador con tu nombreXX, 160 libras de peso(libras), tu altura(pies) y la posición en la que quieras jugar, el código 700, procedencia el barrio en el que vivas y equipo The Torpedors. Comprueba que la fila se ha insertado correctamente en la tabla, si es así, confirma la operación.

```
insert into jugadores (codigo,Nombre,Altura,Peso,Posicion,Procedencia,Nombre_equipo)  
values (700,"Azael05","5-6",160,"G","Vecindario","The torpedors");  
commit;
```

432	010	Brandon Wright	Vecindario	5-5	200	1	Warriors
433	700	Azael05	Vecindario	5-6	160	G	The torpedors
*							

6. Introduce en la tabla jugadores un nuevo jugador con tu apellido de nombre, tu peso , tu altura y la posición en la que quieras jugar, el código 700, procedencia el barrio en el que vivas y equipo Ducati. ¿Qué respuesta obtienes? ¿Por qué?

Que no se puede porque el código es el mismo y no se puede insertar en un mismo pk

```
create table prueba(
  id INT(6) PRIMARY KEY,
  nombre VARCHAR(30),
  equipo VARCHAR(30),
  peso INT(11)
);
```

7. Crea una nueva tabla llamada prueba según la siguiente sintaxis

```
11 • create table prueba(
12   id INT(6) Primary key,
13   nombre VARCHAR(30),
14   equipo VARCHAR(30),
15   peso INT(11)
16 );
```

8. Introduce en la tabla prueba los datos correspondientes de todos los jugadores de los LAKERS

```
insert into prueba select codigo,Nombre,Nombre_equipo,Peso from jugadores
where Nombre_equipo="Lakers"
```

9. Introduce en la tabla prueba un nuevo jugador con identificador 710, en nombre poner tu nombreXX y el resto de datos son los mismos que los del jugador con identificador 700.

```
insert into prueba values (710,"Azael05","The torpedors",160);
```

- 10. Introduce en la tabla prueba sólo el id, nombre y peso correspondientes de todos los jugadores de los Bulls con un peso inferior a 200.**

```
insert into prueba (id,nombre,peso) select codigo,Nombre,Peso from jugadores
where nombre_equipo="Bulls" and Peso<200;
```

11. Borrado de filas

- a. Borra todos los jugadores de los Lakers de la tabla prueba

```
delete from prueba where equipo="Lakers";
```

- b. Realiza una consulta para que se muestren todos los registros de la tabla prueba. Realiza un recorte de la captura y pégalo a continuación.

Result Grid					Filter Rows:					Edit:					Export/Import:					Wrapping				
#	id	nombre	equipo	peso																				
1	221	JamesOn Curry	NULL	190																				
2	223	Chris Duhon	NULL	185																				
3	227	Kirk Hinrich	NULL	190																				
4	228	Larry Hughes	NULL	185																				
5	710	Azael05	The torpedors	160																				
*	NULL	NULL	NULL	NULL																				

c. Borra todas las filas de la tabla prueba. Confirma la operación

```
delete from prueba;
```

12. Introduce en la tabla prueba los datos de los 10 jugadores con el menor peso de la nba

```
insert into prueba select codigo,Nombre,Nombre_equipo,Peso from jugadores  
order by Peso asc limit 10;
```

13. Modifica el peso del jugador 542 de la tabla prueba a 160

```
update prueba set peso=160 where id=542;
```

14. Modifica el peso de la tabla prueba para que el peso se muestre en Kilogramos en vez de en libras.

```
update prueba set peso=peso*0.45;
```

15. El jugador Juan Dixon ha fichado por The Torpedors

```
update prueba set equipo="The torpedors" where nombre="Dixon";
```

16. El jugador con el código 342 de la tabla prueba cambia su peso y equipo por el peso y equipo del jugador 392 de la tabla jugadores. Recuerda que en la tabla jugadores el peso sigue estando en libras y nosotros lo queremos en Kg

```
update prueba set peso=(select peso from jugadores where id=392)*0.45,  
equipo=(select equipo from jugadores where id=392) where id=342;
```

17. El jugador 53 ha cambiado al equipo de Gabe Pruitt y su peso ha cambiado al peso máximo de su equipo. Realizar los cambios en una única sentencia y sólo con los datos facilitados. (No vale hacer una consulta independiente para saber el equipo).

```
> update prueba set equipo=(select Nombre_equipo from jugadores where  
> Nombre="Gabe Pruitt"),peso=(select peso from jugadores where equipo=(  
- select Nombre_equipo from jugadores where Nombre="Gabe Pruitt")  
- order by equipo desc limit 1 )*0.45
```

18. Confirma la transacción anterior, abre una terminal, conecta con la bases de datos y realiza una consulta para que se muestren todos los registros de la tabla prueba. Realiza un recorte de la captura y pégalo en el siguiente recuadro

id	nombre	equipo	peso
9	Rashard McCants	Celtics	83
53	Juan Carlos Navarro	Celtics	83
87	T.J. Ford	Celtics	83
182	Earl Boykins	Celtics	83
252	Gabe Pruitt	Celtics	83
263	Speedy Claxton	Celtics	83
342	Aaron Brooks	Celtics	83
392	Sergio Rodriguez	Celtics	83
542	Juan Dixon	Celtics	83
700	Azael05	Celtics	83

19. Crea la tabla prueba2 con la siguiente sintaxis.

Create table prueba2 as select * from prueba;

```
create table prueba2 as select * from prueba;
```

a. Comprueba su estructura

la misma que prueba

b. Comprueba ahora qué datos tiene la tabla prueba2

la misma que prueba

20. Elimina todos los registros de la tabla prueba2

```
delete from prueba2;
```

21. Introduce en la tabla todos los jugadores que empiecen por A y acaben por S y a todos los jugadores con un código mayor de 650.

```
insert into prueba2 select codigo,Nombre,Nombre_equipo,Peso from jugadores  
where Nombre like 'A%s' or codigo>650 ;
```

22. Añadir la columna *media_puntos*(float) a la tabla prueba2

```
alter table prueba2 add column(media_puntos Float);
```

23. Introducir en la columna media_puntos del jugador 403 la media de puntos de toda su carrera

```
insert into prueba2 (id,media_puntos) select jugador, avg(Puntos_Por_Partido) from estadisticas where jugador = 403;
```

24. Introducir en la columna media_puntos la media de puntos en toda su carrera del jugador 342. Confirmar los cambios

```
insert into prueba2 (id,media_puntos) select jugador, avg(Puntos_Por_Partido) from estadisticas where jugador = 342;
```

25.Activa el modo Autocommit

```
set autocommit=1;
```

26. Inicia una transacción que realice las siguientes acciones en la tabla prueba2

- a. Actualiza la media de puntos de cada jugador que tenga ese dato vacío a 4.

```
update prueba2 set media_puntos= 4 where media_puntos is null;
```

- b. Actualiza la media de puntos de modo que sea igual a la media de puntos de cada jugador multiplicado por el menor de los pesos de todos los jugadores de la tabla jugadores.

```
update prueba2 set media_puntos= media_puntos*(select peso from jugadores order by peso asc limit 1);
```

- c. Actualiza la media de puntos del jugador que tenga tu nombre (usando nombre como restricción) a NULL

```
update prueba2 set media_puntos=null where nombre="Azael05";
```

- d. Eliminar los jugadores con peso entre 180 y 200 (ambos incluidos) y media de puntos por debajo de 100

```
delete from prueba2 where peso>180 and peso<200 and media_puntos<100;
```

- e. Revisa todos los valores actualizados y confirma la transacción.

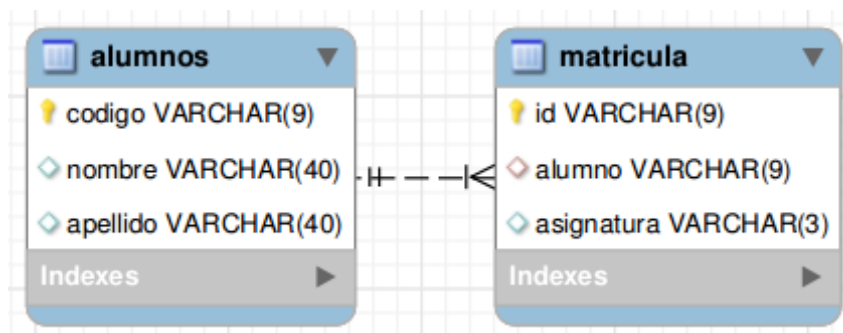
```
commit;
```


27. Abre una terminal, conecta con la bases de datos y realiza una consulta para que se muestren todos los registros de la tabla prueba2. Realiza un recorte de la captura y pégalo en el siguiente recuadro

id	nombre	equipo	peso	media_puntos
531	Andre Owens	Pacers	200	84
549	Antonio McDyess	Pistons	245	84
603	Andris Biedrns	Warriors	230	84
700	Azael05	The torpedors	160	NULL
403	NULL	NULL	NULL	163.227
342	NULL	NULL	NULL	105

APARECE NULL PORQUE SOLO TE PIDE LA MEDIA DE PUNTOS

28. Para los siguientes ejercicios deberás utilizar la siguiente base de datos, créala:



matricula.alumno es FK y hace referencia a alumnos.código

```

• create database colegioX;
• use colegioX;
• create table alumnos(codigo VARCHAR(9) primary key,nombre VARCHAR(40),apellido VARCHAR(40));
• create table matricula (id VARCHAR(9) primary key, alumnos varchar(9), asignatura varchar(3),
foreign key (alumnos) references alumnos(codigo)) ;
  
```

29.Insertar dos alumnos. Uno al que llamaremos X con tu nombre apellidos y otro para la destacada alumna Cristina Cifuentes.

```
insert into alumnos values (1,"Azael","Carballo"),(2,"cristina","Cifuentes");
```

30.Inserta a los dos alumnos las asignaturas de LND y BAE.

```
insert into alumnos values (1,"Azael","Carballo"),(2,"cristina",  
insert into matricula values(1,"1","LND"),(2,"1","Bae"),  
(3,"2","LND"),(4,"2","Bae");
```

31.Modificar la opción de referencia de clave foránea para que tanto al actualizar como al eliminar este en RESTRICT.

Al no escoger ninguna opción se queda como restrict.

32.Inicia una transacción

```
set autocommit=0;
```

33.Elimina el alumno X y verifica que sucede en la tabla matrícula

No se puede ya que está en restrict y no te deja.

```
delete from alumnos where codigo=1;
```

Yo le puse 1 en vez de x, perdón

34.¿Qué ha sucedido?

Que no se puede eliminar porque tiene clave referencial en modo restrict.

35.Cancela la transacción

```
rollback;  
set autocommit=1;
```

36.Inicia una transacción

```
set autocommit=0;
```

37.Modifica el código del alumno X y verifica que sucede en las dos tablas.

```
update alumnos set codigo=5 where codigo=1;
```

38. ¿Qué ha sucedido?

No se puede porque también tiene on update restrict es decir que no se puede cambiar

39. Cancela la transacción

```
rollback;  
set autocommit=1;
```

40. Modificar la opción de referencia de clave foránea para que tanto al actualizar como al eliminar este en CASCADE.

```
alter table matricula drop foreign key matricula_ibfk_1;  
alter table matricula add foreign key (alumnos) references alumnos(codigo)  
on update cascade on delete cascade;
```

41. Inicia una transacción

```
set autocommit=0;
```

42. Elimina el alumno X y verifica que sucede en la tabla matrícula

```
delete from alumnos where codigo=1;
```

43. ¿Qué ha sucedido?

Que se borra en las dos tablas porque está el delete en cascada

44. Cancela la transacción

```
rollback;|  
set autocommit=1;
```

45. Inicia una transacción

```
set autocommit=0;
```

46. Modifica el código del alumno X y verifica que sucede en las dos tablas.

```
update alumnos set codigo=3 where codigo=1;
```

47. ¿Qué ha sucedido?

Que se cambia en ambas tablas ya que tenemos el update en cascada

48.Cancela la transacción

```
rollback;  
set autocommit=1;
```

49.Modificar la opción de referencia de clave foránea para que tanto al actualizar como al eliminar este en SET NULL.

```
alter table matricula drop foreign key matricula_ibfk_1;  
alter table matricula add foreign key (alumnos) references alumnos(codigo)  
on update set null on delete set null;
```

50.Inicia una transacción

```
set autocommit=0;  
|
```

51.Elimina el alumno X y verifica que sucede en la tabla matrícula

```
delete from alumnos where codigo=1;
```

52.¿Qué ha sucedido?

Que en la tabla alumnos se ha borrado el campo y en la tabla matricula se ha quedado en nulo porque tenemos el delete en set null

53.Cancela la transacción

```
rollback;  
set autocommit=1;
```

54. Inicia una transacción

```
set autocommit=0;
```

55. Modifica el código del alumno X y verifica que sucede en las dos tablas.

```
update alumnos set codigo=3 where codigo=1;
```

				alumnos	matricula
				2	3
				Azael	Carballo
1	1	NULL	LND		
2	2	NULL	Bae		

56. ¿Qué ha sucedido?

Que en la tabla alumno se cambia el código y en la tabla matricula se queda nulo porque está en set null.

57. Cancela la transacción.

```
rollback;  
set autocommit=1;
```