Corrección Tarea 2: Gestión de Vistas y bloqueos tablas

Ejercicio

1. Crear la vista ***Clases2*** con los campos codigo, grupo, nombre de tutor de la tabla Clases para las clases con puntuación mayor de 4.

```
CREATE VIEW Clases2 AS SELECT codigo, grupo, nombre_tutor FROM Clases
WHERE puntuacion>4;
```

2. Crear la vista ***jugadores_puestos*** que tome las columnas codalumno, nombre, apellidos de la tabla Jugadores y nombre de la tabla Puestos para los que son "BASE".

```
CREATE OR REPLACE VIEW jugadores_puestos AS SELECT jugadores.codalumno,
jugadores.nombre, jugadores.apellido, puestos.nombre as puesto_nombre
 FROM jugadores , puestos
WHERE jugadores.puesto=puestos.codigo AND puestos.nombre="BASE";
```

3. Crear la vista ***clases_puestos*** que devuelva para cada clase el código y el nombre de puestos, suma de tantos_marcados por cada clase y puesto.

```
CREATE VIEW clases_puestos AS
SELECT J.clase, P.nombre, sum(tantos_marcados) as tantos
FROM Jugadores J, Puestos P
WHERE  J.puesto=P.codigo
GROUP BY J.clase, P.nombre;
```

4. Crear la vista ***vista3*** que para cada clase y para cada puesto devuelva los campos código, grupo y puntuación de clases, nombre de puestos y que cuente el número de jugadores en cada puesto. ¿podrías insertar un registro en esta vista? Justifica la respuesta.

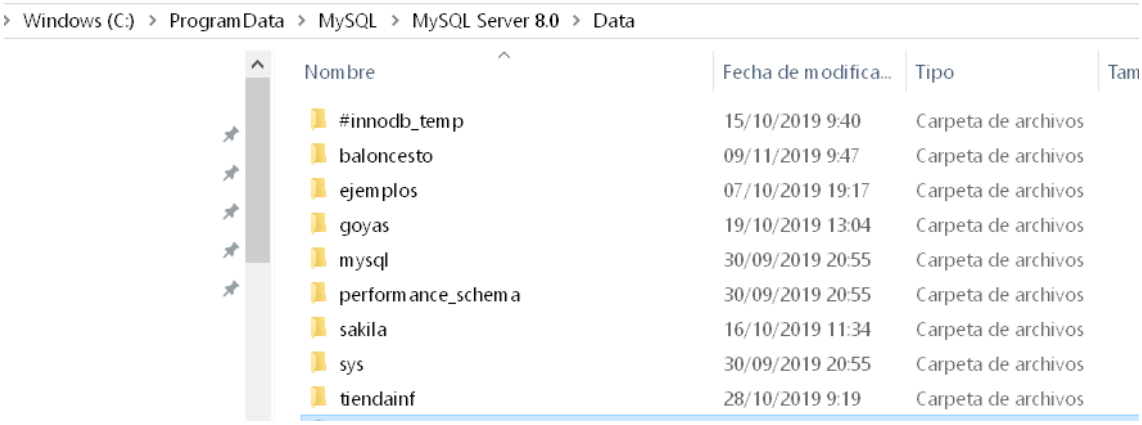
```
CREATE OR REPLACE VIEW vista3 AS
SELECT C.codigo, C.grupo, C.puntuacion, P.nombre, count(codalumno) as numalum
FROM Jugadores J, Puestos P, Clases C
WHERE J.puesto=P.codigo
AND J.clase = C.codigo
GROUP BY C.codigo, P.nombre;
```

5. Mostrar los directorios de datos para las bases de datos de MySQL creadas hasta el momento en esta BD.

```
1 •  SHOW VARIABLES LIKE '%dir%';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Variable_name	Value
basedir	C:\Program Files\MySQL\MySQL Server 8.0\
binlog_direct_non_transactional_updates	OFF
character_sets_dir	C:\Program Files\MySQL\MySQL Server 8.0\share\charsets\
datadir	C:\ProgramData\MySQL\MySQL Server 8.0\Data\



6. Crea y permite al usuario de nombre ***abd*** conectarse al servidor de MySQL desde el host local con la contraseña *ruby* con todos los privilegios sobre las tablas de la base de datos de nombre baloncesto.

```
CREATE USER abd@localhost IDENTIFIED BY 'rubi';

GRANT ALL ON baloncesto.* TO abd@localhost;
```

Para comprobar, que han sido aplicados podemos mirar la tabla del sistema mysql.db o el comando

```
SHOW GRANTS FOR abd@localhost;
```

7. Crear el superusuario de nombre ***Esther*** que pueda hacer todas las operaciones sobre todas las tablas de todas las bases de datos desde el host local con la contraseña *café* y que además sea capaz de trasferir privilegios a otros usuarios.

```
CREATE USER Esther@localhost IDENTIFIED BY 'café';
GRANT ALL PRIVILEGES ON *.* TO Esther@localhost WITH GRANT OPTION;

Para comprobar, que han sido aplicados podemos mirar la tabla del sistema mysql.user  o el comando

SHOW GRANTS FOR Esther@localhost;
```

8. Crear el usuario de nombre ***asistente*** con privilegios de actualización desde el host local sobre las columnas *codalumno, nombre, apellido, tantos_marcados* de la tabla Jugadores de la base de datos Baloncesto.

```
CREATE USER asistente@localhost;
GRANT UPDATE (codalumno, nombre, apellido, tantos_marcados) ON Baloncesto.Jugadores TO asistente@localhost;

Para comprobar, que han sido aplicados podemos mirar la tabla del sistema mysql.columns_priv  o el comando

SHOW GRANTS FOR asistente@localhost;
```

Imaginemos ahora que el motor ***no soporta transacciones*** (tablas tipo MyISAM) por lo que debemos especificar cómo hacer los bloqueos:

9. Crear la tabla **Puntosclase** con 2 campos codigo_clase y puntos_totales.

```
CREATE TABLE Puntosclase(codigo_clase  char(3) PRIMARY KEY, puntos_totales int(10));
```

10. Inserta valores para las clases E1A y E1B.

```
INSERT INTO Puntosclase VALUES ('1EA',50);
INSERT INTO Puntosclase VALUES ('E1B',70);
```

11. Tras leer los ejercicios 12, 13 y 14 Utilizar los bloqueos necesarios para bloquear adecuadamente Jugadores y Puntosclase y realizar correctamente la transacción.

```
LOCK TABLES Jugadores READ,  Puntosclase  as pcr READ, Puntosclase WRITE;
```

La transacción es

```
SET autocommit=0;
LOCK TABLES jugadores READ, puntosclase as pcr READ, puntosclase WRITE;
UPDATE puntosclase
SET puntos_totales=(SELECT sum(tantos_marcados) FROM jugadores WHERE clase='E1A')
WHERE codigo_clase='E1A';
COMMIT;
UNLOCK TABLES;
```

12. Obtener la suma de puntos de todos los jugadores pertenecientes a la clase 'E1A';

```
SELECT SUM(tantos_marcados) FROM Jugadores where clase='E1A';
```

13. Modificar en Puntosclase el campo puntos con el valor obtenido en la consulta anterior para la clase E1A.

```
UPDATE Puntosclase
SET puntos_totales=(SELECT SUM(tantos_marcados) FROM Jugadores where clase='E1A')
WHERE codigo_clase='E1A';
```

14. Desbloquear todas las tablas anteriores

```
UNLOCK TABLES;
```