

Módulo:

UT05:

Actividad de desarrollo:

Bases de Datos

Tratamiento de datos

Concurrencia

1. Objetivo general

Reproducir los problemas típicos de concurrencia.

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.

3. Descripción:

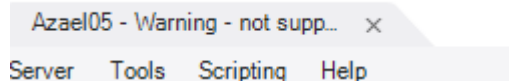
Usando la terminal para conectarse a la base de datos realizar los siguientes puntos:

La entrega de esta actividad consiste en responder a las preguntas planteadas en los ejercicios y 3 capturas de pantalla realizadas en los momentos que se indican y responder a las preguntas realizadas

1. Crear el usuario *NombreXX* con todos los privilegios. (Nombre es tu nombre y XX tu número para actividades).

```
create user 'Azael05'@'%' identified by '1234';  
Grant all privileges on *.* to 'Azael05'@'%';
```

2. Conecta con el SGBD usando el usuario creado en el punto anterior



3. Crear la base de datos UD5.

```
1 create database UD5;
```

4. Abrir la bases de datos UD5.

```
2 • USE UD5
```

5. Consulta el nivel de aislamiento de la sesión actual. ¿Cuál es el nivel por defecto?

```
• show variables like '%isolation%';
```

Para la sesión la variable es REPEATABLE-READ

6. Consulta el nivel de aislamiento global. ¿Cuál es el nivel por defecto?

```
show global variables like '%isolation%'
```

La variable por defecto global es la misma que para el usuario actual.

7. Establecer el tipo de aislamiento de la sesión a read uncommitted

```
SET SESSION TRANSACTION isolation level READ UNCOMMITTED;
```

8. ¿Qué problemas de concurrencia permite que se produzcan el nivel read uncommitted?

Se producen dirty, phantom, non repeatable y actualizaciones perdidas

9. Confirmar que el tipo de aislamiento se ha modificado

	Variable_name	Value
▶	transaction_isolation	READ-UNCOMMITTED

10. Crear la tabla coches con la columna nombre [varchar(20)] y con el motor innodb

```
create table coches(nombre VARCHAR(20)) engine =innodb;
```

11. Inserta los valores toyota, audi y Tu nombre y Apellido

```
create table coches(nombre VARCHAR(20)) engine =innodb;  
insert into coches values ("Toyota"),("Audi"),("Azael"),("Carballo");
```

12. Iniciar otra sesión (Sesión 2). Abrir otra sesión con la base de datos.

Daniel - Warning - not support... x

13. Ejecutar prompt [U:\U BD:\d]> en las dos sesiones para que se aprecie el nombre de usuario en el terminal.

No tengo terminal en casa así que lo tendré que hacer desde el workbench



PRIMER PROBLEMA

1. Sesión1: Iniciar transacción y consultar los valores de la tabla coches

```
set autocommit=0;  
select * from coches;
```

2. Sesión2: Modifica el nombre audi a volvo (Si utilizas transacciones confirma los cambios)

```
update coches set nombre="volvo" where nombre="audi";
```

3. Sesión1: consultar los valores de la tabla coches.

En la 1 sesión

	nombre
▶	Toyota
	volvo
	Azael
	Carballo

4. Fíjate que ha sucedido entre la primera y segunda consulta estando dentro de la misma transacción.¿Qué tipo de problema de concurrencia ha ocurrido?

Está fallando non repeatable read, porque hemos consultado el dato dos veces y se ha modificado en la segunda

5. Realizar una captura de la pantalla con las dos sesiones

Azael05 - Warning - not supp... x Daniel - Warning - not supported x

6. Cancelar la transacción

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

SEGUNDO PROBLEMA

1. Sesión1: Iniciar transacción y consultar los valores de la tabla coches

```
set autocommit=0;  
select * from coches;
```

2. Sesión2: Inserta un nuevo registro en la tabla (renault)

```
insert into coches values("renault");
```

3. Sesión1: consultar los valores de la tabla coches.

	nombre
▶	Toyota
	volvo
	Azael
	Carballo
	renault

4. Fíjate que ha sucedido entre la primera y segunda consulta estando dentro de la misma transacción. ¿Qué tipo de problema de concurrencia ha ocurrido?

Se ha añadido el Renault, ha habido un problema de phantom read, ya que se ha añadido una fila más.

5. Realizar una captura de la pantalla con las dos sesiones



TERCER PROBLEMA

1. Crear la tabla total con la columna suma (int)

```
create table normal(suma int);
```

2. Sesión1: Iniciar transacción y consultar los valores de la tabla coches

```
set autocommit=0;
select * from coches;
```

3. Sesión2: Iniciar transacción e Inserta un nuevo registro en la tabla (alfa)¿?

```
set autocommit=0;
insert into coches values("Mercedes");
```

4. Sesión1: consultar los valores de la tabla coches y realizar una consulta que cuente el total de filas.

```
select * from coches;
select count(*) from coches;
```

5. Sesión1: insertar en la tabla total el número total filas de la tabla coches usando la consulta anterior.

```
insert into normal select count(*) from coches;
```

6. Sesión2: cancelar transacción.


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

7. Sesión1: realizar una consulta que cuente el total de filas.

	count(*)
▶	5

8. Fíjate que ha sucedido con el dato que hemos guardado en la tabla total. ¿Ese dato es cierto? ¿Qué tipo de problema de concurrencia ha ocurrido?

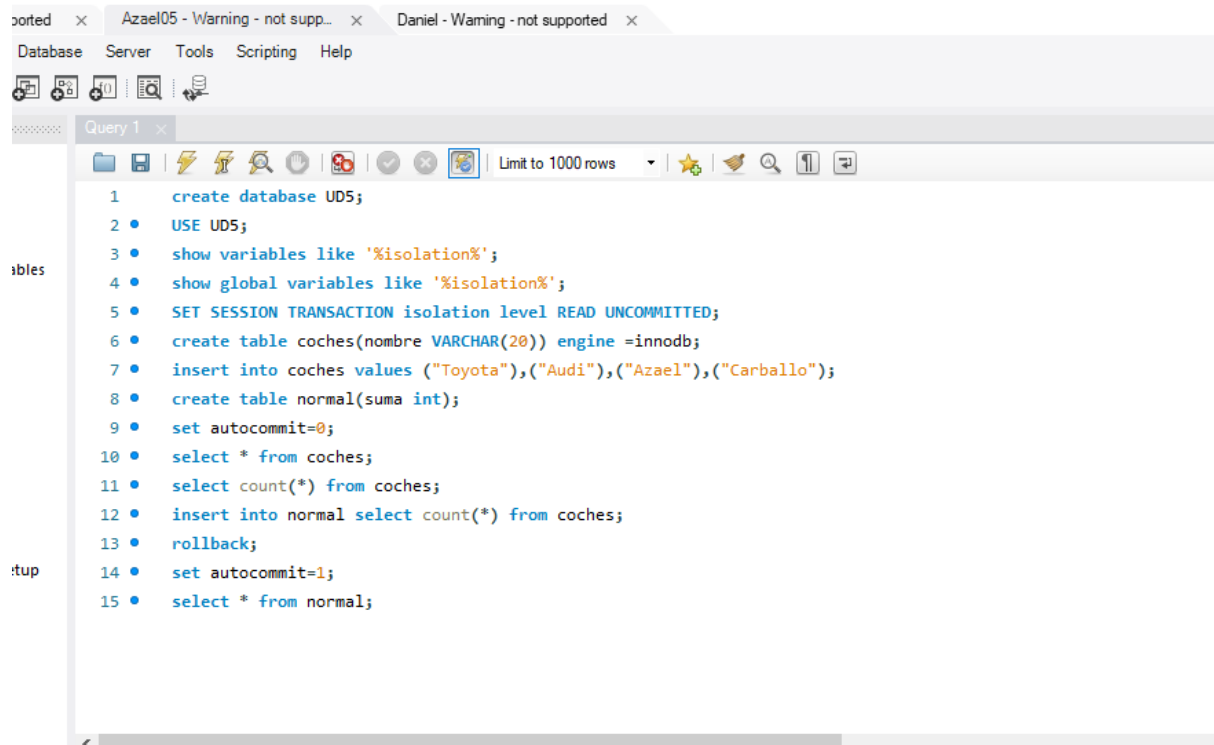
```
15 • select * from normal;
```

<	
Result Grid  Filter Rows: <input type="text"/>	
	suma
▶	6

No es cierto, el tipo de concurrencia es actualizaciones perdidas , ya que se han iniciado dos transacciones a la vez y una ha actualizado a la otra y después la he cancelado pero aún así se ha sobrescrito

9. Realizar una captura de la pantalla con las dos sesiones

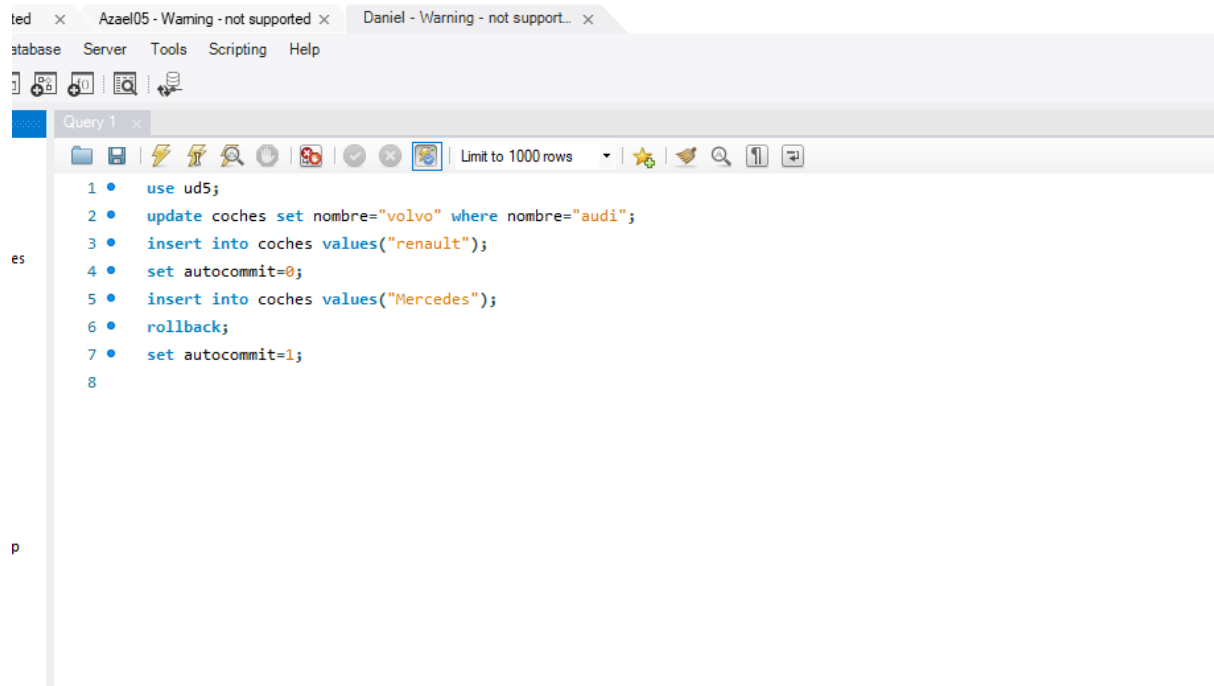
Sesión 1 con todas las consultas necesarias hechas



The screenshot shows a MySQL IDE window with a menu bar (Database, Server, Tools, Scripting, Help) and a toolbar. The main editor displays a SQL script for 'Query 1'. The script includes commands to create a database, use it, show variables, create tables, insert data, and perform a rollback. The script is as follows:

```
1 create database UD5;
2 USE UD5;
3 show variables like '%isolation%';
4 show global variables like '%isolation%';
5 SET SESSION TRANSACTION isolation level READ UNCOMMITTED;
6 create table coches(nombre VARCHAR(20)) engine =innodb;
7 insert into coches values ("Toyota"),("Audi"),("Azael"),("Carballo");
8 create table normal(suma int);
9 set autocommit=0;
10 select * from coches;
11 select count(*) from coches;
12 insert into normal select count(*) from coches;
13 rollback;
14 set autocommit=1;
15 select * from normal;
```

Sesión 2 con todas las consultas hechas



The screenshot shows a MySQL IDE window with a menu bar (Database, Server, Tools, Scripting, Help) and a toolbar. The main editor displays a SQL script for 'Query 1'. The script includes commands to use a database, update data, insert data, and perform a rollback. The script is as follows:

```
1 use ud5;
2 update coches set nombre="volvo" where nombre="audi";
3 insert into coches values("renault");
4 set autocommit=0;
5 insert into coches values("Mercedes");
6 rollback;
7 set autocommit=1;
8
```