



POLITÉCNICA
"Ingeniamos el futuro"

CAMPUS
DE EXCELENCIA
INTERNACIONAL

Universidad Politécnica de Madrid
ETS de Ingenieros Informáticos



BASES DE DATOS

Ingeniería Informática

Matemáticas e Informática

BASES DE DATOS

Acceso a Bases de datos en Java

BASES DE DATOS

Contacto con Prof. Alejandro Rodríguez

Email: alejandro.rg@upm.es

Transacciones (I)

Una transacción es un conjunto de instrucciones a ejecutar de forma indivisible o atómica.

Un SGBD se dice que es transaccional (ACID compliant) si tiene las funcionalidades necesarias para que sus transacciones tengan características ACID:

- Atomicidad
- Consistencia
- Aislamiento
- Persistencia

Transacciones (II)

START TRANSACTION; //no necesario set autocommit=0;
toda transacción marcada con start transaction el
autocommit se desactiva – también se puede usar BEGIN;

```
INSERT INTO actor(first_name, last_name, last_update)
VALUES ('Leonardo', 'DiCaprio', STR_TO_DATE('2016-
08-03 00:00:00', '%Y-%d-%m %k:%i:%s'));
```

```
COMMIT;
```

Transacciones (II)



Ver ejemplo – EX1

Transacciones (III)

START TRANSACTION;

```
INSERT INTO actor(first_name, last_name, last_update)
VALUES ('Marisa', 'Tomei', STR_TO_DATE('2012-03-05
00:00:00', '%Y-%d-%m %k:%i:%s'));
```

ROLLBACK;

Transacciones (III)



Ver ejemplo – EX2

Operadores de asignación (I)

“:=“ Asigna un valor

“= “ Asigna un valor (como parte de una sentencia SET o como parte de la cláusula SET de una sentencia UPDATE)

Operadores de asignación (II)

```
SELECT @nombre:=first_name FROM actor where  
actor_id = 104; //nombre del actor
```

```
SELECT @film:=title FROM actor,film_actor,film where  
actor.actor_id = 104 and actor.actor_id =  
film_actor.actor_id and film_actor.film_id = film.film_id and  
film.length > 180; //peliculas donde ha participado que  
duran más de 3h
```

```
SELECT @nombre; //vemos el nombre
```

```
SELECT @film; //vemos la película (solo habrá una)
```

Operadores de asignación (II)



Ver ejemplo – EX3

Transacciones en Java (I)

```
conn.setAutoCommit(false);

String query = "INSERT INTO actor (first_name, last_name, last_update) VALUES (?, ?, ?)";

PreparedStatement pst1 = conn.prepareStatement(query);
pst1.setString(1, "Luis");
pst1.setString(2, "Tosar");
pst1.setDate(3, new Date(new java.util.Date().getTime()));

int result1 = pst1.executeUpdate();

Statement st2 = conn.createStatement();
int result2 = st2.executeUpdate("DELETE FROM actor where actor_id = '5'");

conn.commit();

System.out.println("Número de filas afectadas (R1): " + result1);
System.out.println("Número de filas afectadas (R2): " + result2);
System.out.println("Query ejecutada!");
```

El DELETE va a dar error. Explicar que ocurre.



Transacciones en Java (I)



Ver ejemplo – EJ2_9

Transacciones en Java (II)

```
Statement st2 = conn.createStatement();
int result2 = st2
    .executeUpdate("DELETE FROM actor where actor_id = '5'");

conn.commit();

System.out.println("Número de filas afectadas (R1): " + result1);
System.out.println("Número de filas afectadas (R2): " + result2);
System.out.println("Query ejecutada!");

pst1.close();
st2.close();
conn.close();
} catch (SQLException esql) {
    System.err.println("Mensaje: " + esql.getMessage());
    System.err.println("Código: " + esql.getErrorCode());
    System.err.println("Estado SQL: " + esql.getSQLState());
} catch (ClassNotFoundException e) {
    System.err.println("Error al cargar el driver.");
}
```

Transacciones en Java (II)



Ver ejemplo – EJ2_10

Transacciones en Java (III)

```

conn.setAutoCommit(false);

// Insertamos un actor nuevo

String query = "INSERT INTO actor (first_name, last_name, last_update) VALUES (?, ?, ?)";

PreparedStatement pst1 = conn.prepareStatement(query);
pst1.setString(1, "Luis");
pst1.setString(2, "Tosar");
pst1.setDate(3, new Date(new java.util.Date().getTime()));

int result1 = pst1.executeUpdate();

Statement st2 = conn.createStatement();
ResultSet rs2 = st2.executeQuery("SELECT * from actor where last_name = 'Tosar'");
while (rs2.next()) {
    int id = rs2.getInt("actor_id");
    String firstName = rs2.getString("first_name");
    String lastName = rs2.getString("last_name");
    Date lastUpdate = rs2.getDate("last_update");

    System.out.println("Actor:");
    System.out.println("\tID: " + id);
    System.out.println("\tName: " + firstName);
    System.out.println("\tLast name: " + lastName);
    System.out.println("\tLast update: " + lastUpdate.toString());
}

```

Supongamos que el INSERT se ejecuta sin errores. ¿Devolverá el resultado el SELECT?



Transacciones en Java (III)



Ver ejemplo – EJ2_11

Transacciones en Java (IV)

```

conn.setAutoCommit (false);

// Insertamos un actor nuevo

String query = "INSERT INTO actor (first_name, last_name, last_update) VALUES (?, ?, ?)";

PreparedStatement pst1 = conn.prepareStatement (query);
pst1.setString (1, "Luis");
pst1.setString (2, "Tosar");
pst1.setDate (3, new Date (new java.util.Date ().getTime ()));

int result1 = pst1.executeUpdate ();

Statement st2 = conn.createStatement ();
ResultSet rs2 = st2.executeQuery ("SELECT * from actor where last_name = 'Tosar'");
while (rs2.next ()) {
    int id = rs2.getInt ("actor_id");
    String firstName = rs2.getString ("first_name");
    String lastName = rs2.getString ("last_name");
    Date lastUpdate = rs2.getDate ("last_update");

    System.out.println ("Actor:");
    System.out.println ("\tID: " + id);
    System.out.println ("\tName: " + firstName);
    System.out.println ("\tLast name: " + lastName);
    System.out.println ("\tLast update: " + lastUpdate.toString ());
}

```

Supongamos que el INSERT se ejecuta sin errores. ¿Devolverá el resultado el SELECT?



Transacciones en Java (IV)

```

conn.commit(); // hacemos un commit antes del delete. Esto hará que la
// se ejecute y el delete forme parte de otra diferente

Statement st3 = conn.createStatement();
int result3 = st3
    .executeUpdate("DELETE FROM actor where actor_id = '5'");

System.out.println("Número de filas afectadas (R1): " + result1);
System.out.println("Número de filas afectadas (R3): " + result3);
System.out.println("Query ejecutada!");

pst1.close();
st2.close();
st3.close();
conn.close();

```

Y ahora, el DELETE falla. ¿Estará el dato del INSERT en la BD? ¿Por qué?



Transacciones en Java (IV)



Ver ejemplo – EJ2_12