

**MMSQL**

**Nasrin Khodapanah**



# AGENDA

- Security
  - Stored Procedures
  - Stored Functions
  - Triggers



# Triggers



# TRIGGERS - PRINCIPALS

- Triggers are objects of a database.
- Triggers are attached to a table, they will trigger the execution of a statement or a block of instructions, when, multiple rows are inserted, deleted, or modified in the table to which they are attached.
- An SQL trigger allows you to specify SQL actions that should be executed automatically when a specific event occurs in the database.
- For example, you can use a trigger to automatically update a record in one table whenever a record is inserted into another table.



# TRIGGERS-PRINCIPALS

- Triggers are like Stored Procedures; we can use them to execute one or many instructions.
- The only difference is that you cannot call a trigger, it has to be triggered by an event.
- The event could be:
  - INSERT
  - UPDATE
  - DELETE



# TRIGGERS - PRINCIPALS

- Once the trigger is triggered, the instruction can be executed either before the event or after the event.
- A trigger can insert or update any table but the one that triggered the trigger (attached table).
- A trigger can only modify (insert, update, delete) the line being processed.



# TRIGGERS - CREATE

```
CREATE TRIGGER trigger_name  
when event  
ON table_name FOR EACH ROW  
BEGIN  
    -- body  
END;
```



# TRIGGERS - CREATE

- trigger\_name – the trigger name
- when – before or after
- event – insert, update or delete
- body – where line or bloc of code will go





# TRIGGERS - CREATE

- **CREATE TRIGGER** after\_insert\_animal  
**AFTER INSERT ON** Animal **FOR** EACH  
ROW

**BEGIN**

– body

**END;**



# TRIGGERS - CREATE

- Conventions
  - Only one trigger per event combination
  - BEFORE UPDATE / AFTER DELETE, etc
  - So maximum of six



# TRIGGERS - CREATE

- Conventions
  - Only one trigger per event combination
  - BEFORE UPDATE / AFTER DELETE, etc
  - So maximum of six



# TRIGGERS - CREATE

- OLD and NEW
  - Default MySQL variables in order to retrieve the column value before or after a trigger.
- OLD: values can be read but not modified
- NEW: values can be both read and modified.



# TRIGGERS-DELETE

- In the case of an UPDATE the OLD and NEW values coexist.
- In the case of an INSERT OLD value doesn't exist.
- In the case of a DELETE the NEW value doesn't exist.
- Examples
  - REFER to workspace-triggers.sql

# EXO

- Create a function that retrieves the animal's id and returns its price. Price is determined by the race's price if not exist by species' price.

Then execute the following query :

```
SELECT name, dob, sex, get_species_name(species_id)
species, get_mother_name(id) mom, get_price(id) price
FROM animal
```

- Add a new column to the "species" table to show the count of each species number found in the "animal" table by utilizing the "count\_species" function and inserting the appropriate values.

Then create a trigger that updates this column each time a record is inserted into the "animal" table. To test it insert a new record in animal table.

