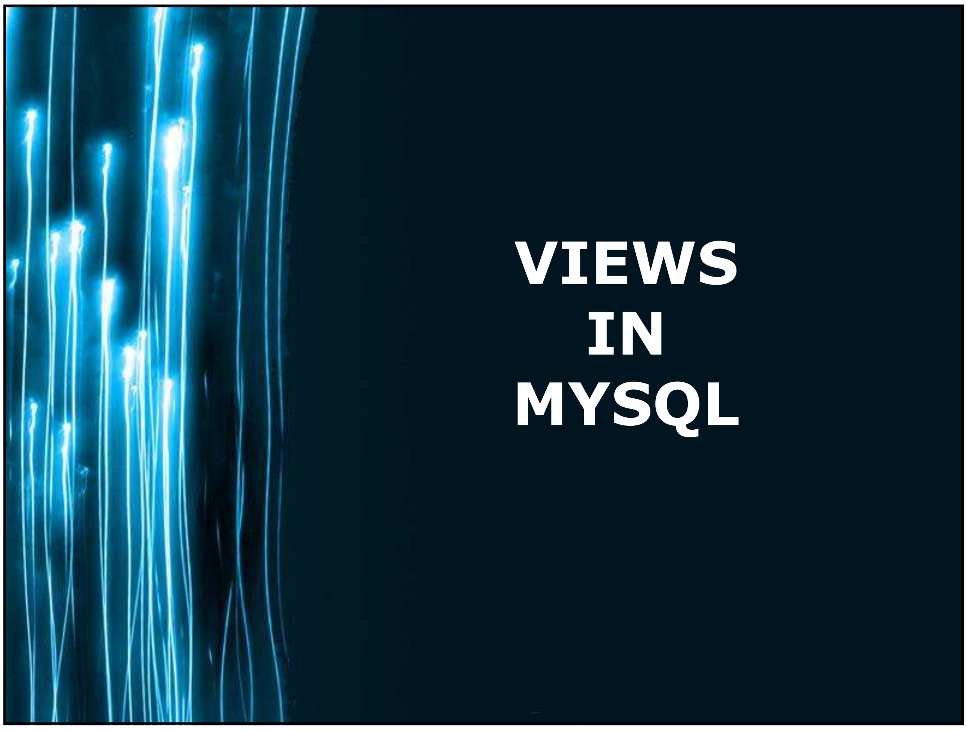
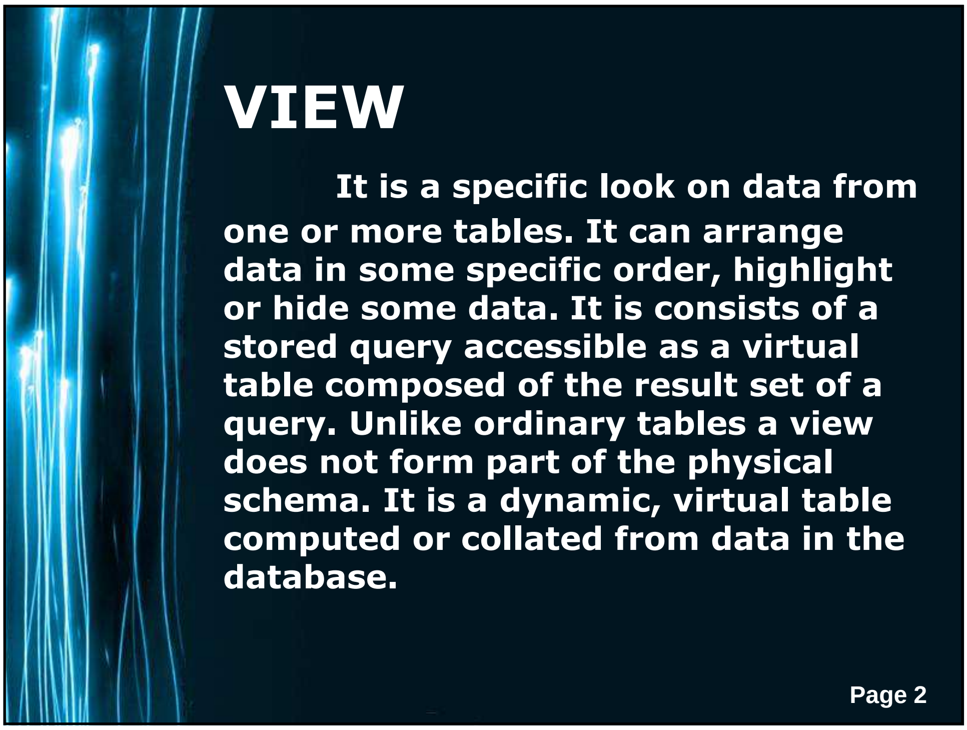
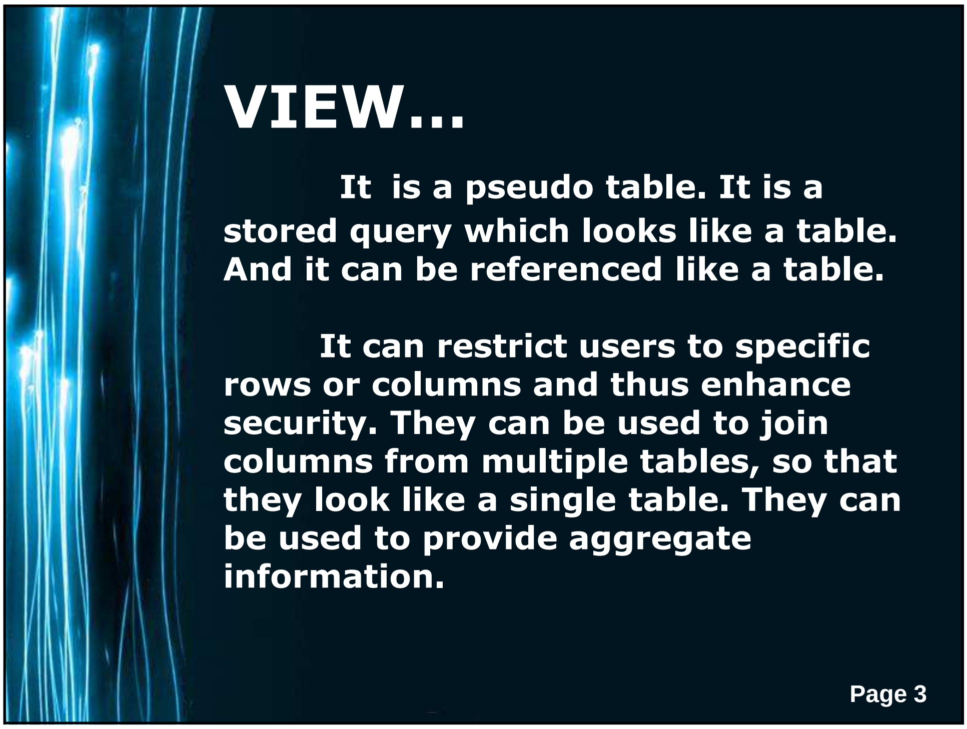


VIEWS IN MYSQL



VIEW

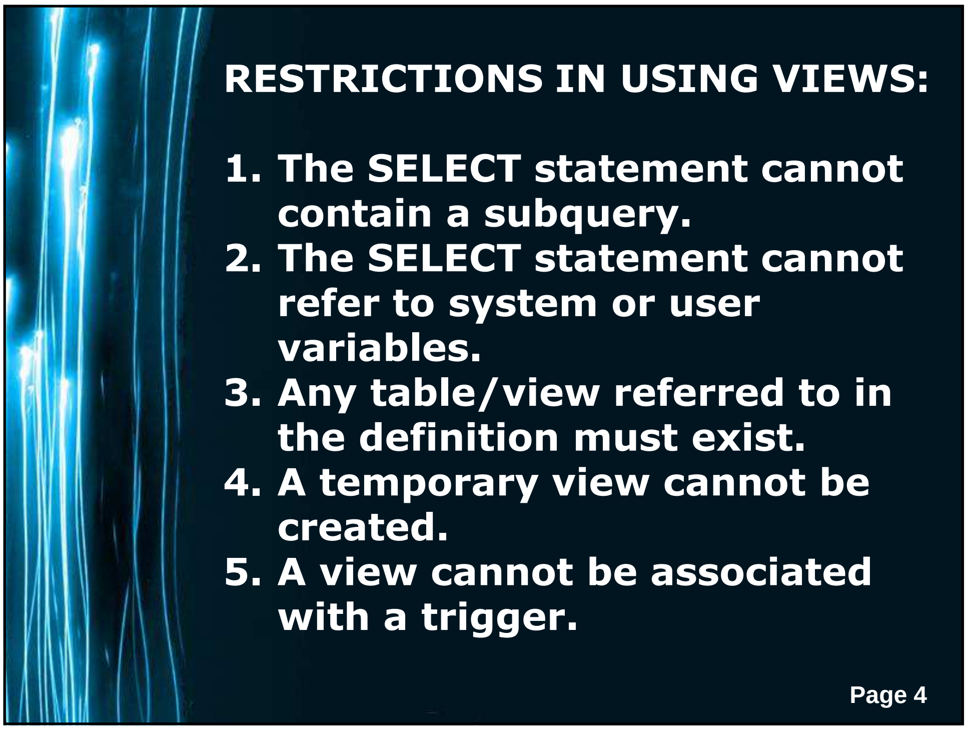
It is a specific look on data from one or more tables. It can arrange data in some specific order, highlight or hide some data. It is consists of a stored query accessible as a virtual table composed of the result set of a query. Unlike ordinary tables a view does not form part of the physical schema. It is a dynamic, virtual table computed or collated from data in the database.



VIEW…

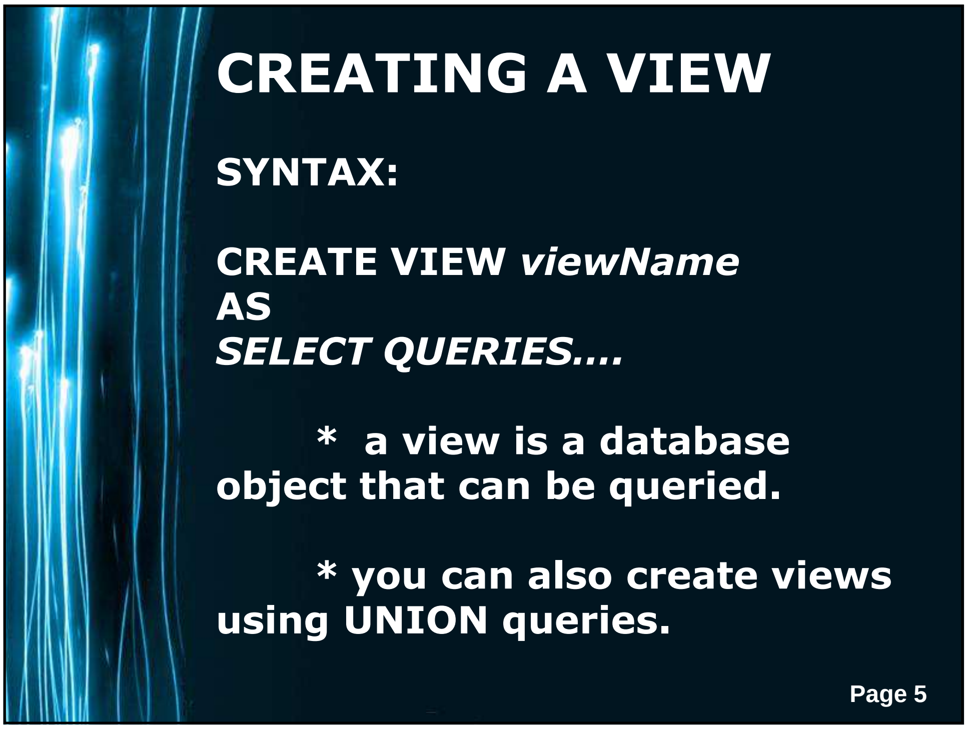
It is a pseudo table. It is a stored query which looks like a table. And it can be referenced like a table.

It can restrict users to specific rows or columns and thus enhance security. They can be used to join columns from multiple tables, so that they look like a single table. They can be used to provide aggregate information.



RESTRICTIONS IN USING VIEWS:

1. The SELECT statement cannot contain a subquery.
2. The SELECT statement cannot refer to system or user variables.
3. Any table/view referred to in the definition must exist.
4. A temporary view cannot be created.
5. A view cannot be associated with a trigger.



CREATING A VIEW

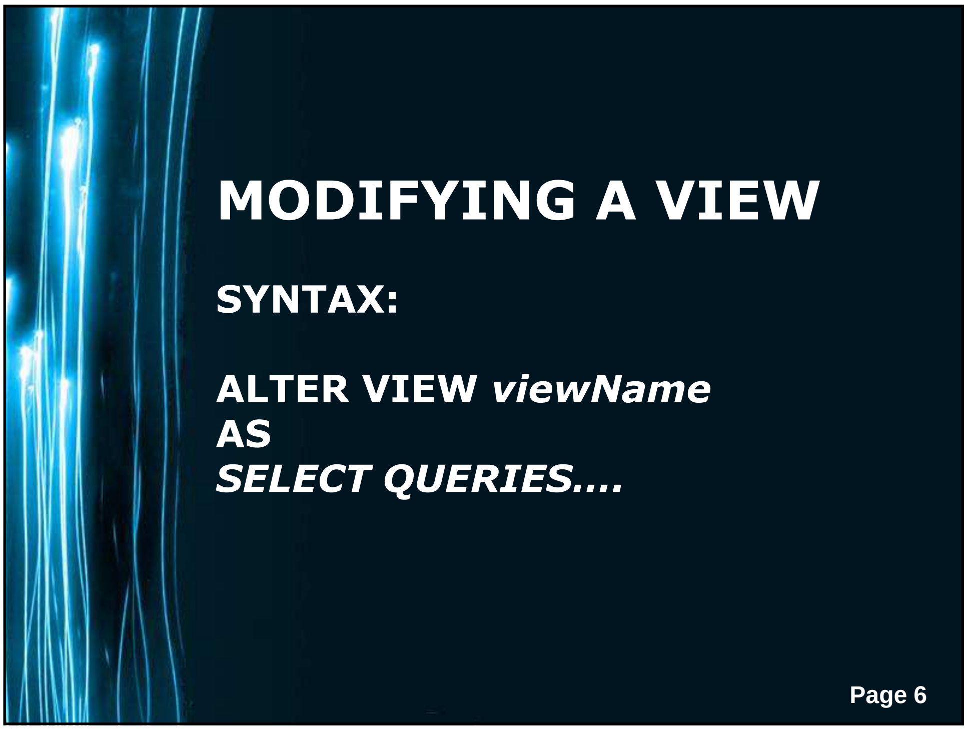
SYNTAX:

CREATE VIEW *viewName*
AS
SELECT QUERIES….

*** a view is a database object that can be queried.**

*** you can also create views using UNION queries.**

Page 5

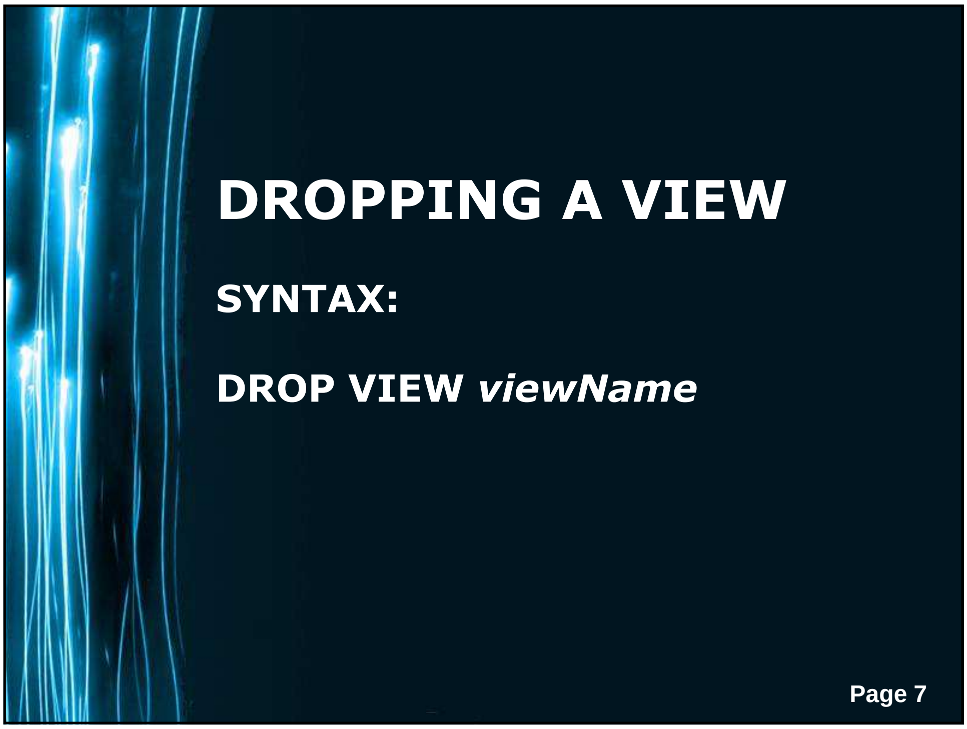


MODIFYING A VIEW

SYNTAX:

ALTER VIEW *viewName*
AS
SELECT QUERIES….

Page 6



DROPPING A VIEW

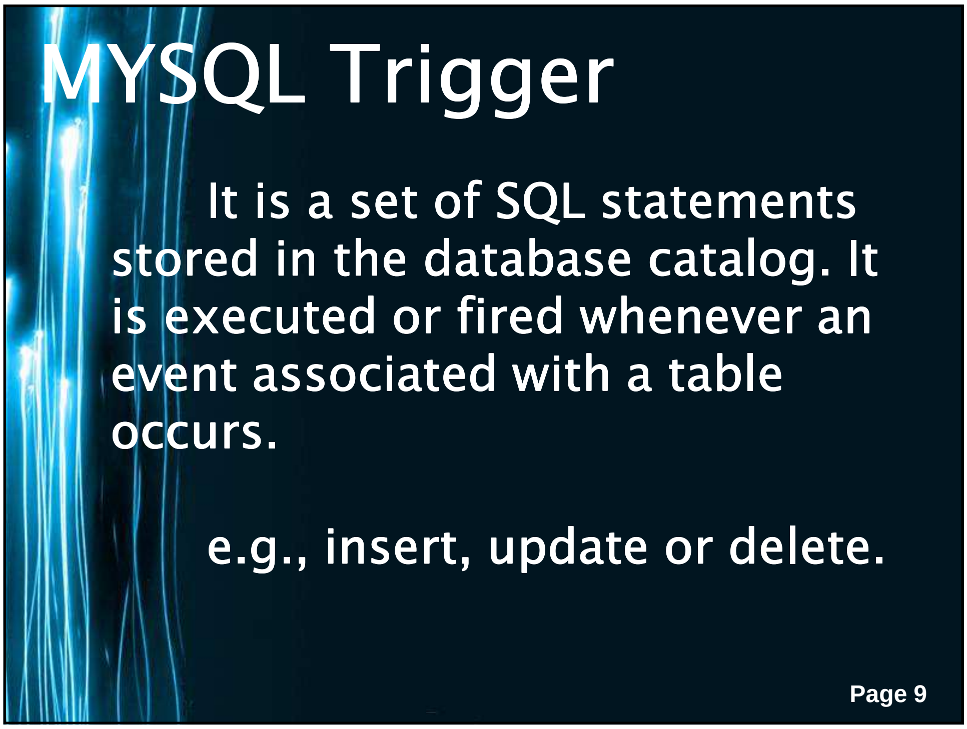
SYNTAX:

DROP VIEW *viewName*

Page 7

MySQL Trigger

Page 8

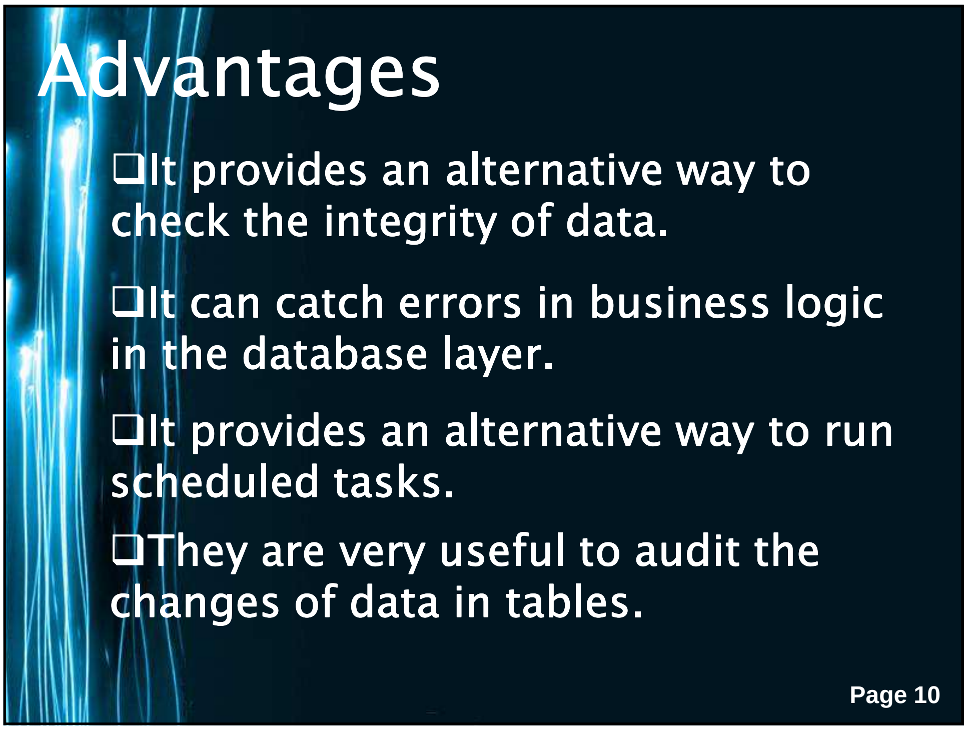


MYSQL Trigger

It is a set of SQL statements stored in the database catalog. It is executed or fired whenever an event associated with a table occurs.

e.g., insert, update or delete.

Page 9



Advantages

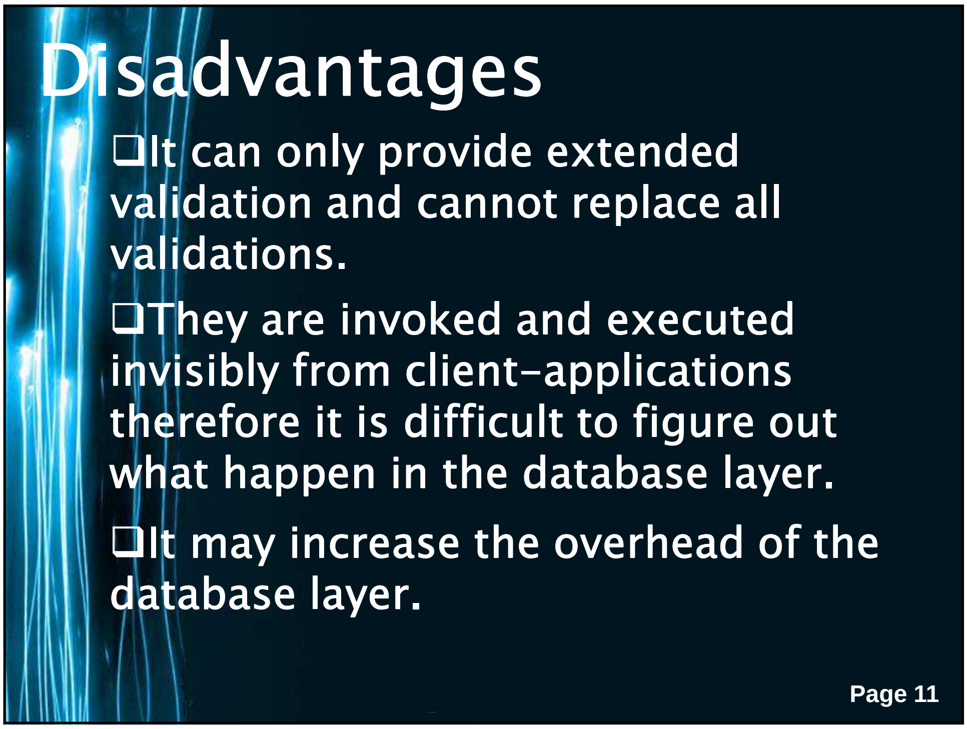
❑It provides an alternative way to check the integrity of data.

❑It can catch errors in business logic in the database layer.

❑It provides an alternative way to run scheduled tasks.

❑They are very useful to audit the changes of data in tables.

Page 10

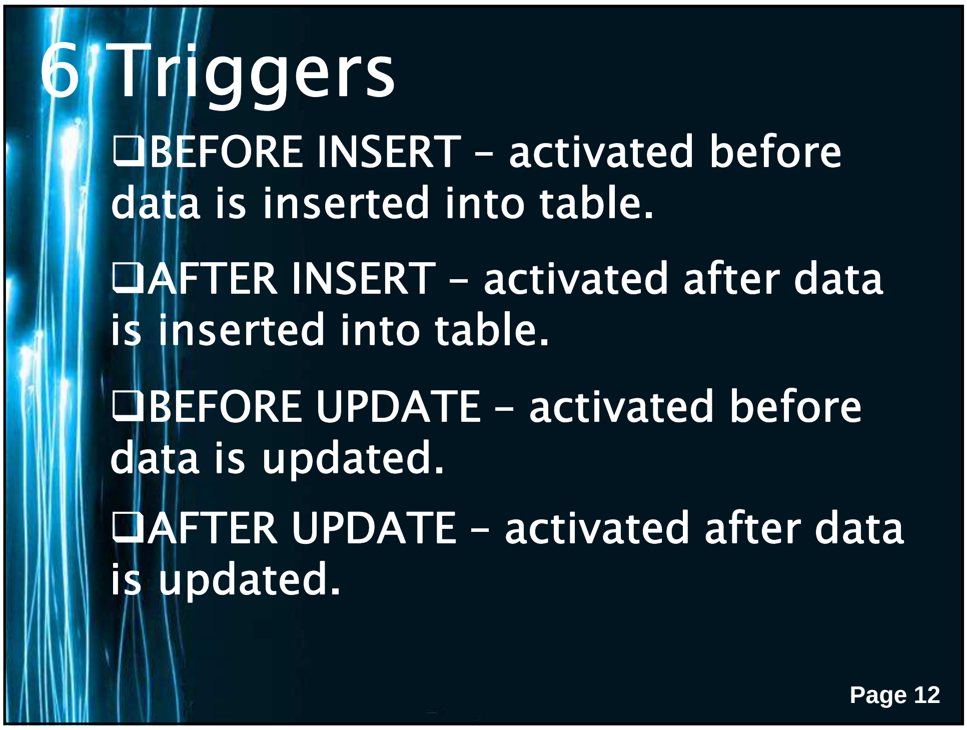


Disadvantages

❑It can only provide extended validation and cannot replace all validations.

❑They are invoked and executed invisibly from client-applications therefore it is difficult to figure out what happen in the database layer.

❑It may increase the overhead of the database layer.



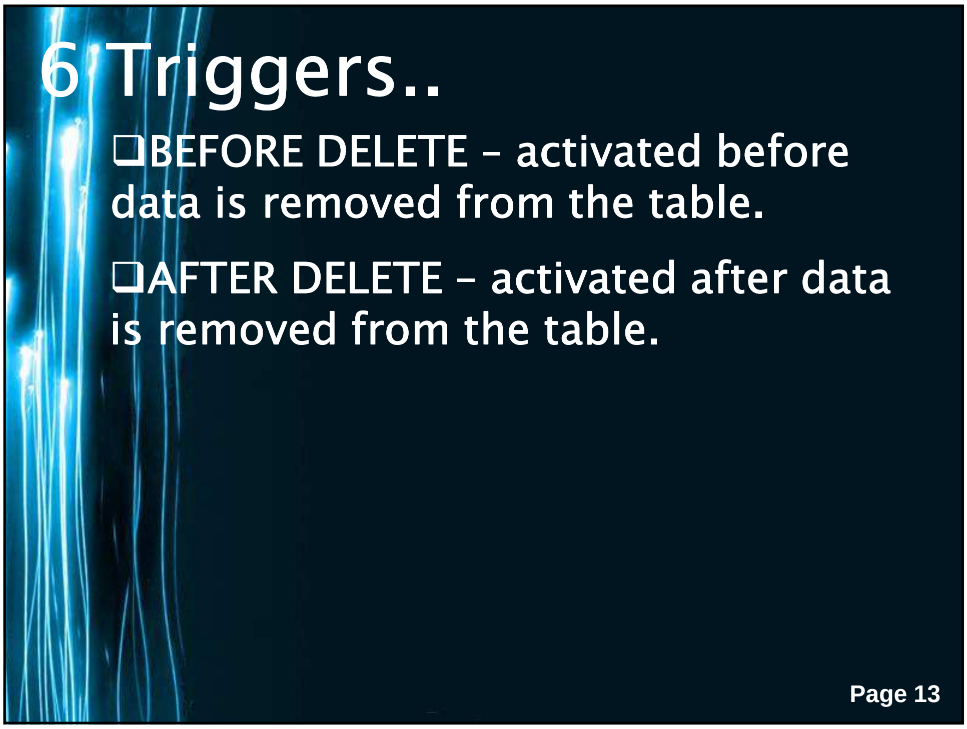
6 Triggers

❑BEFORE INSERT – activated before data is inserted into table.

❑AFTER INSERT – activated after data is inserted into table.

❑BEFORE UPDATE – activated before data is updated.

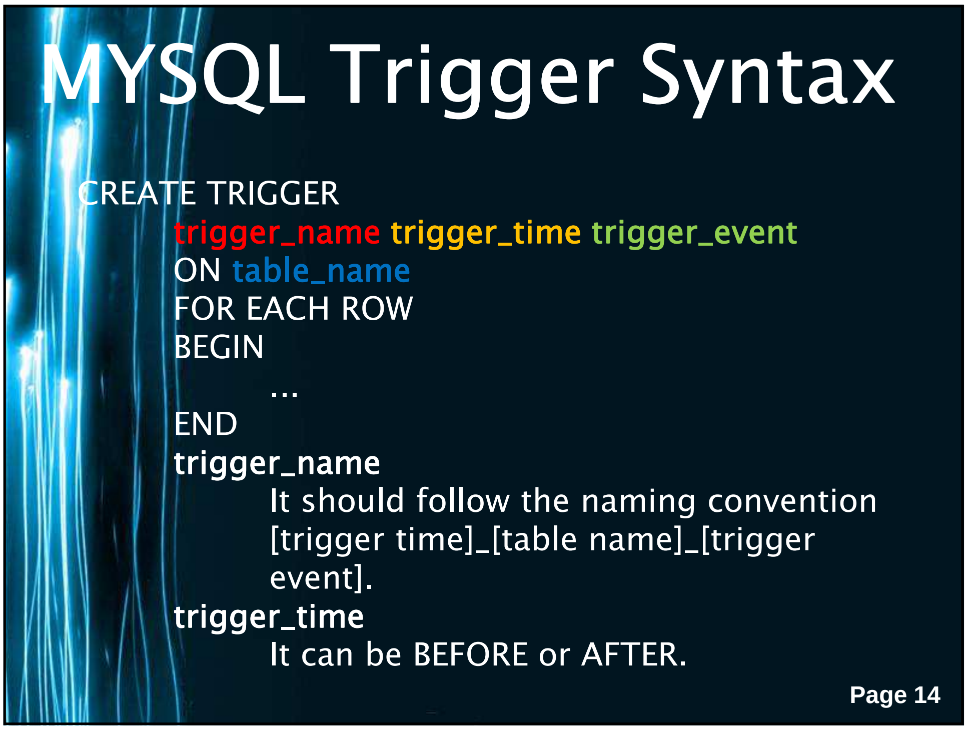
❑AFTER UPDATE – activated after data is updated.



6 Triggers..

- ❑BEFORE DELETE – activated before data is removed from the table.
- ❑AFTER DELETE – activated after data is removed from the table.

Page 13



MYSQL Trigger Syntax

```
CREATE TRIGGER
        trigger_name trigger_time trigger_event
        ON table_name
        FOR EACH ROW
        BEGIN
                ...
        END
```

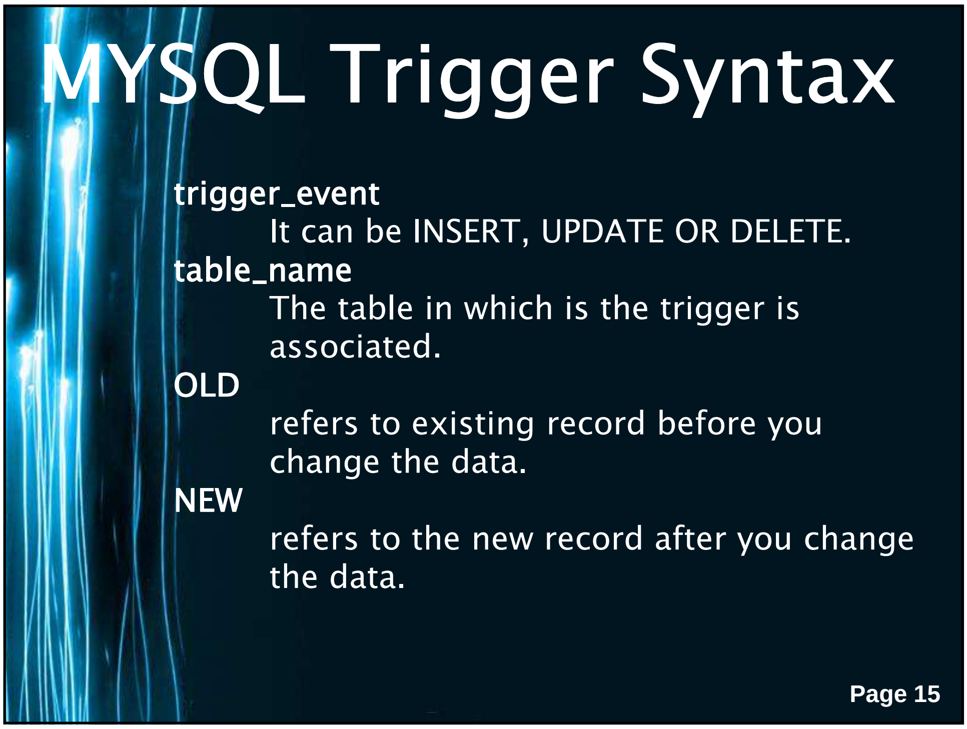
trigger_name

It should follow the naming convention [trigger time]_[table name]_[trigger event].

trigger_time

It can be BEFORE or AFTER.

Page 14



MYSQL Trigger Syntax

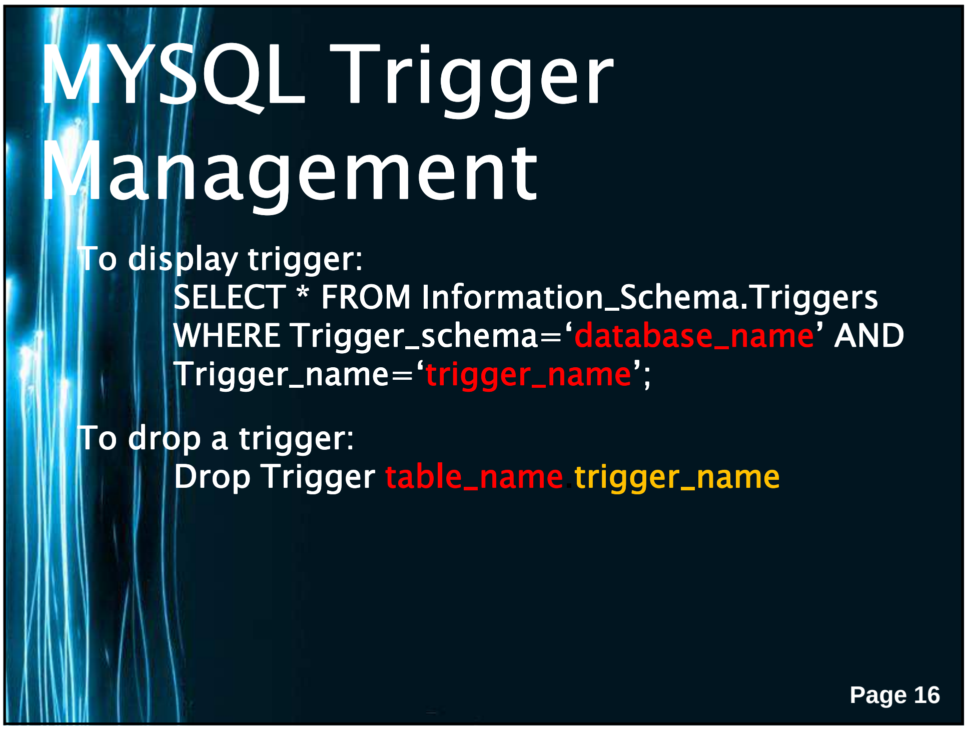
trigger_event
It can be INSERT, UPDATE OR DELETE.

table_name
The table in which is the trigger is associated.

OLD
refers to existing record before you change the data.

NEW
refers to the new record after you change the data.

Page 15

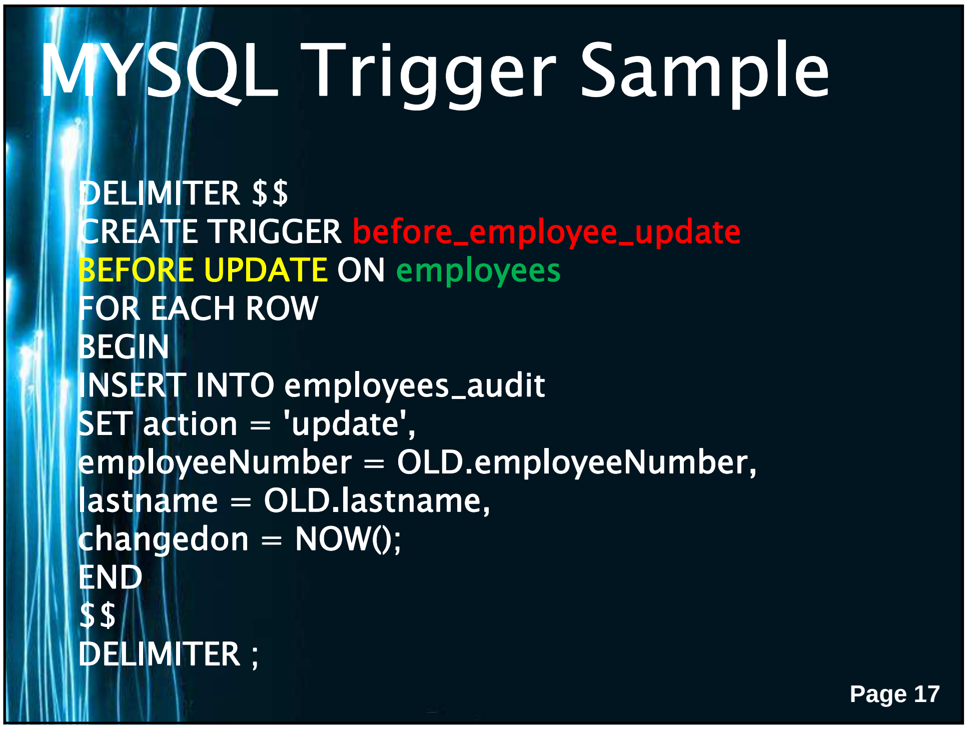


MYSQL Trigger Management

To display trigger:
SELECT * FROM Information_Schema.Triggers WHERE Trigger_schema='database_name' AND Trigger_name='trigger_name';

To drop a trigger:
Drop Trigger table_name trigger_name

Page 16



MYSQL Trigger Sample

```
DELIMITER $$
CREATE TRIGGER before_employee_update
BEFORE UPDATE ON employees
FOR EACH ROW
BEGIN
INSERT INTO employees_audit
SET action = 'update',
employeeNumber = OLD.employeeNumber,
lastname = OLD.lastname,
changedon = NOW();
END
$$
DELIMITER ;
```