DDL

By- Saurabh Kumar Sharma

Course Objective

- To create ,drop and alter the tables in MySQL Database
- To implement constraints in table while creating or altering the table

Session Objective

- DDL – create, alter, drop & truncate
- Constraints and its types

MySQL Introduction

- MySQL is a database management system used for many small and big businesses.
- MySQL is developed, marketed and supported by MySQL AB - a Swedish company.
- MySQL is a open source database
- MySQL supports large databases, up to 50 million rows or more in a table. The

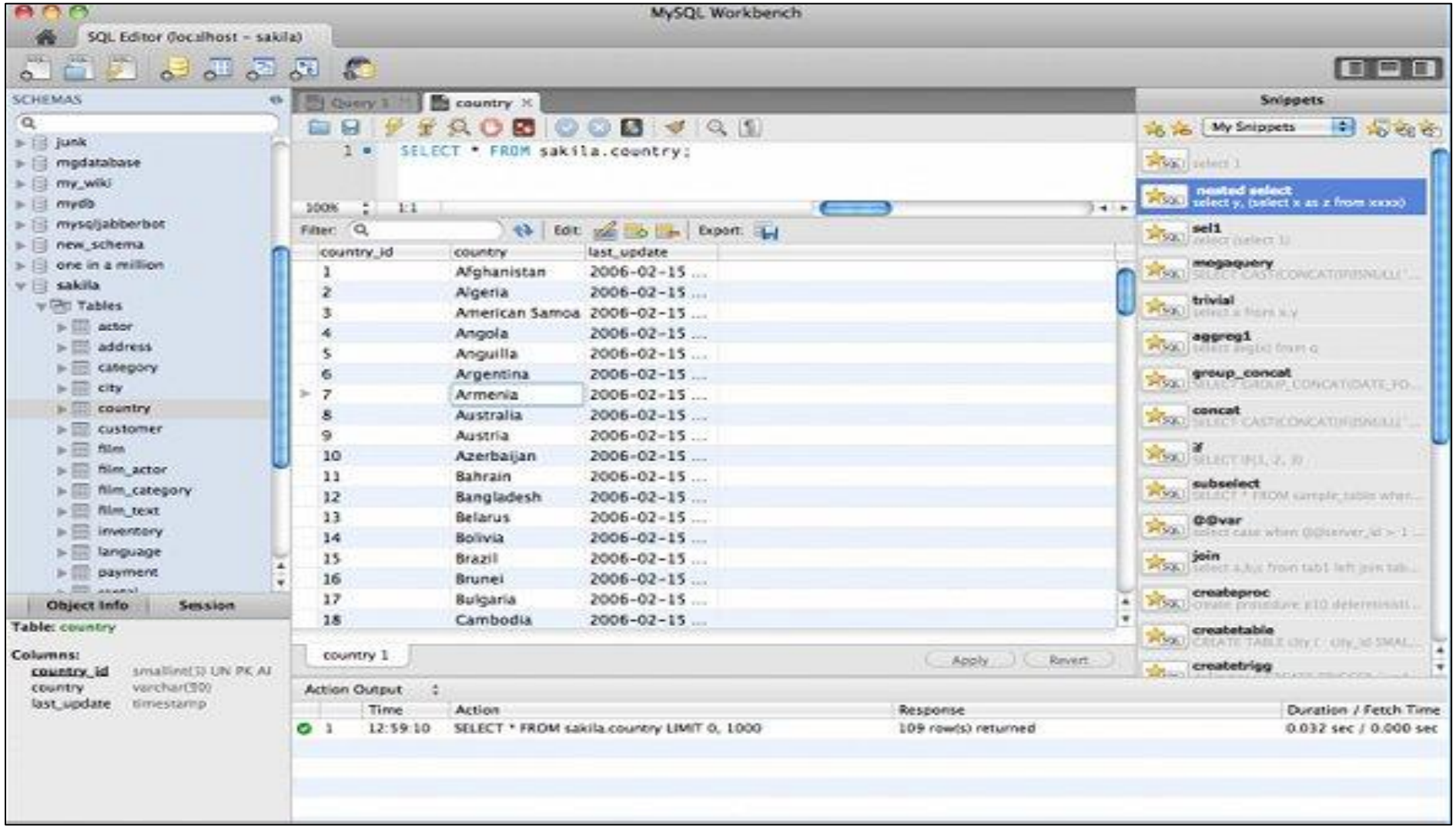
  default file size limit for a table is 4GB, but you can increase

  to a theoretical limit of 8 million terabytes (TB).

Database Client GUI

Database Client GUI
Workbench
Sequel Pro
HeidiSQL
SQLyog
SQLWave
DBTools Manager
MyDB Studio
Navicat for MySQL

Database Client GUI - Workbench



Show Database

```
mysql> SHOW DATABASES;


+--------------+
 | Database |
 +------------+
 | mysql      |
 | test         |
 +------------+

2 rows in set (0.13 sec)
```

Show databases command

Display all database instances in MySQL database

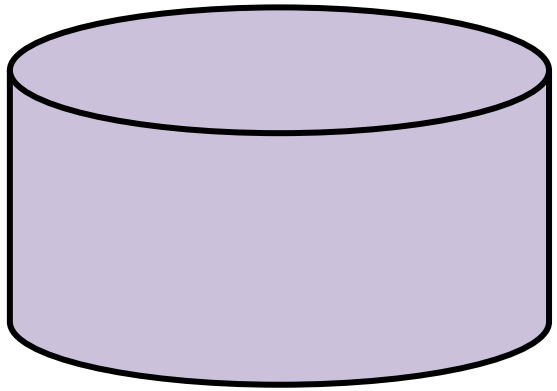
Create Database

You can create and drop a MySQL database instance by using My SQL Workbench by using the command

Create Database:

➔ Create database <<Database Name >>

```
Create database Training
```



Drop Database :

➔Drop Database <<Database Name >>

```
Drop database Training
```

DDL Statement

DDL

- DDL is short name of Data Definition Language.
- DDL deals with database schemas like table.

DDL Commands

- CREATE – create the structure of a data base object (ex: table).
- ALTER – alters the structure of the existing database
- DROP – delete objects from the database
- TRUNCATE – remove all records from a table, including all spaces allocated for the records are removed

Create Table

- CREATE TABLE Table_Name (column_specifications)
- Example

```
CREATE TABLE student
(
  student_ID INT UNSIGNED   NOT NULL,
  name       VARCHAR(20)    NOT NULL,
  major      VARCHAR(50),
grade        VARCHAR(5)
);
Query OK, 0 rows affected (0.00 sec)
```

Student_ID	Name	Major	Grade

Display Table Structure

- **show tables :** command display the tables from current database

➔show tables

Tables in Training
Student
Employee

2 row in set (0.00 sec)

- **describe** : command display the structure of the table

➔Describe student

Fields	Type	Null	Key	Default
Student_ID	Int(10)			0
Name	Varachar(20)			
Major	Varchar(20)	Yes		Null
Grade	Varchar(20)	Yes		null

4 rows in set (0.00 sec)

Modify Table Structure

- ALTER TABLE: alter the existing structure of the table

➔ alter table student add primary key (student_ID);

Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0

➔describe student;

Fields	Type	Null	Key	Default
Student_ID	Int(10)		PRI	0
Name	Varachar(20)			
Major	Varchar(20)	Yes		Null
Grade	Varchar(20)	Yes		null

4 rows in set (0.00 sec)

Drop

Syntax:

DROP TABLE table_name

Example

drop table student;

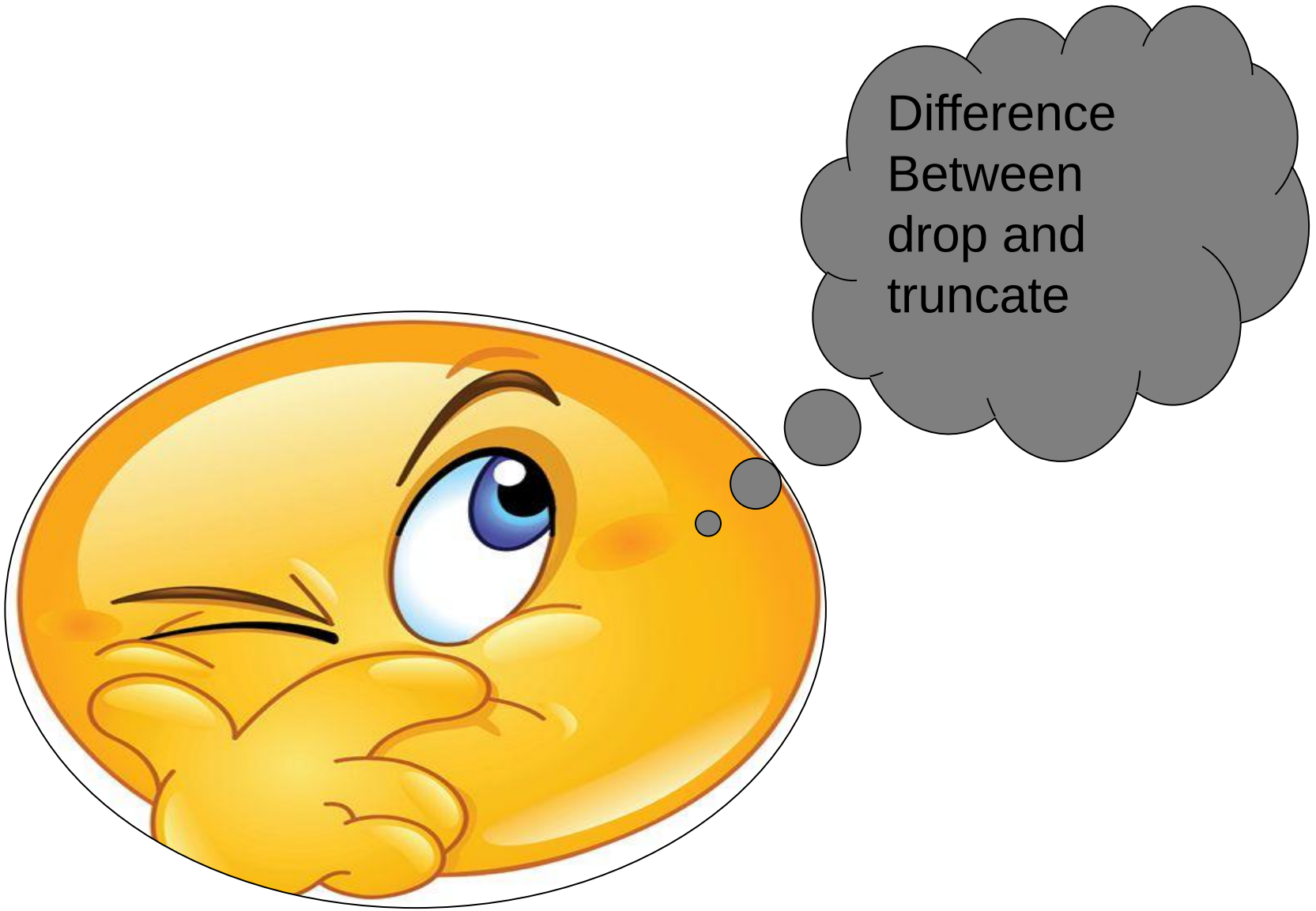
Query OK, 0 rows affected (0.00 sec)

Truncate

Syntax : TRUNCATE TABLE table_name;

Example:

Truncate table student


Difference
Between
drop and
truncate

SQL – Exercise

- Complete the following
  - **Practice - DDL**

Practice on DDL

Constraints

What are Constraints?

- **Constraints enforce rules at the table level.**
- **Constraints prevent the deletion of a table if there are dependencies.**

The following constraint types are valid:

- **NOT NULL**
- **UNIQUE**
- **PRIMARY KEY**
- **FOREIGN KEY**
- **CHECK**

Defining Constraints

Syntax:

CREATE TABLE [*schema*.]*table*
(*column datatype* [DEFAULT *expr*]
[*column_constraint*],
...
[*table_constraint*][,...]);

Example:

```
CREATE TABLE employees(
employee_id NUMBER(6),
first_name VARCHAR2(20),
...
job_id VARCHAR2(10) NOT NULL,
CONSTRAINT emp_emp_id_pk
PRIMARY KEY (EMPLOYEE_ID));
```

The NOT NULL Constraint

- **The NOT NULL Constraint Ensures that null values are not permitted for the column**

- **The NOT NULL constraint can be specified only at the column level, not at the table level.**

Example:

```
CREATE TABLE employees(
employee_id NUMBER(6),
last_name VARCHAR2(25) NOT NULL,
salary NUMBER(8,2),
commission_pct NUMBER(2,2),
hire_date DATE
CONSTRAINT emp_hire_date_nn
NOT NULL)
```

The UNIQUE Constraint

- **A UNIQUE key integrity constraint requires that every value in a column or set of columns (key) be unique**

- **Defined at either the table level or the column level**

Example:

```
CREATE TABLE employees(
employee_id NUMBER(6),
last_name VARCHAR2(25) NOT NULL,
email VARCHAR2(25),
salary NUMBER(8,2),
commission_pct NUMBER(2,2),
hire_date DATE NOT NULL,
…
CONSTRAINT emp_email_uk UNIQUE(email));
```

The PRIMARY KEY Constr

- **A PRIMARY KEY constraint creates a primary key for the table**
- **Defined at either the table level or the column level**

Example:

```
CREATE TABLE departments(
department_id NUMBER(4),
department_name VARCHAR2(30)
CONSTRAINT dept_name_nn NOT NULL,
manager_id NUMBER(6),
location_id NUMBER(4),
CONSTRAINT dept_id_pk PRIMARY KEY(department_id));
```

The FOREIGN KEY Constraint

- **The FOREIGN KEY, or referential integrity constraint, designates a column or combination of columns as a foreign key and establishes a relationship between a primary key or a unique key in the same table or a different table.**

Example:

```
CREATE TABLE employees(
employee_id NUMBER(6),
last_name VARCHAR2(25) NOT NULL,
email VARCHAR2(25),
salary NUMBER(8,2),
commission_pct NUMBER(2,2),
hire_date DATE NOT NULL,
department_id NUMBER(4),
CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)
REFERENCES departments(department_id),
CONSTRAINT emp_email_uk UNIQUE(email));
```

FOREIGN KEY Constraint Keywords

- **FOREIGN KEY: Defines the column in the child table at the table constraint level**
- **REFERENCES: Identifies the table and column in the parent table**
- **ON DELETE CASCADE: Deletes the dependent rows in the child table when a row in the parent table is deleted.**
- **ON DELETE SET NULL: Converts dependent foreign key values to null**

The CHECK Constraint

- **Defines a condition that each row must satisfy**

Example

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    CHECK (Age>=18)
);
```

Adding a Constraint Syntax

Use the ALTER TABLE statement to:

- **Add or drop a constraint, but not modify its structure**
- **Enable or disable constraints**
- **Add a NOT NULL constraint by using the MODIFY Clause**

Syntax

```
ALTER TABLE table
ADD [CONSTRAINT constraint] type (column);
```

Adding a Constraint

- **Add a FOREIGN KEY constraint to the EMPLOYEES table indicating that a manager must already exist as a valid employee in the EMPLOYEES table.**

- **Example:**

```
ALTER TABLE employees
ADD CONSTRAINT emp_manager_fk
FOREIGN KEY(manager_id)
REFERENCES employees(employee_id);
```

Dropping a Constraint

- **Remove the manager constraint from the EMPLOYEES table.**

Example:

```
ALTER TABLE employees
DROP CONSTRAINT emp_manager_fk;
```

- **Remove the PRIMARY KEY constraint on the DEPARTMENTS table and drop the associated FOREIGN KEY constraint on the**

Example:

```
EMPLOYEES.DEPARTMENT_ID column.
Table altered.
ALTER TABLE departments
DROP PRIMARY KEY CASCADE;
```

Questions