Day13

Database - A way storing the data in an organised format.

What is DataBase?

Database is a method of storing, organizing, retrieving the Data.

What is Data?

Data is nothing but a useful information. List of Associates. Details of customers.

What is DBMS?

DBMS stands for Database Management System. It’s Software which helps in storing, retrieving, organising the data in a proper format.

What is the primary Entity in Database?

Table is the primary Entity in database. Table consist of rows and columns. Row is also called tuple.

Intersection of a row & column is a cell.

What are all the important entities in database?

Schema, Table, View, Query, Functions, Stored Procedures, Triggers, Sequence, Materialized View, Cursors

Which Language is used in DBMS/RDBMS?

In DBMS/RDBMS – SQL (Structured Query Language) is used to interact with DB entities.

What is RDBMS?

RDBMS stands for Relational Database Management System. It’s a software used to handle to databases and their relationships.

What is the working model of Database ??

Database is working in Client – Server Model.

Database Server Examples: MS-SQL Server, MySQL Server, Oracle Server, DB2 Server, Postgres Server

Database Client Examples: Dbeaver, MySQL WorkBench, Oracle Web, PgAdmin

What are all the ways of accessing the Database?

1. Using Command Line Client [CUI Mode] – Knowledge of SQL Query is important
2. Using GUI Clients [GUI Mode] – Click, generate & execute with buttons [knowledge of SQL is not much important]
3. Using Programming Lang

What are all the different editions of Database, DBMS & RDBMS?

1. Community Edition (Open Source) – No need to pay for the usage.
2. Enterprise Edition (Proprietary Software) – Need to pay for Licensing

What is AWS?

AWS stands for Amazon Web Service. It’s cloud service provided by amazon.

What are all the important services provided in AWS?

1. Computing Services like EC2
2. Storage Services like S3
3. Networking Services like IGW
4. Database Services like RDS, DynamoDB etc.,
5. Migration Services
6. Business Application Deployment Services

What are all the important Services Names

1. EC2- Elastic Cloud Compute (Scalable Virtual Machine which can be accessed remotely from anywhere)
2. S3 – Simple Storage Service (Scalable Storage service in cloud)
3. IGW – Internet Gate Way
4. RDS – Relational Database Service. Can be used to connect with any Database
5. SG – Security Group
6. AMI – Amazon Machine Image

How to access MySQL server using command line client?

1. In the search area of your OS, simply type MySQL, you will “mysql command line client” option (or) open command prompt or terminal. Simply type mysql/mysqld which will ask username and password provided during installation.
2. “show databases” – To list all the database/schema
3. “use <db\_name>” – To use any existing database
4. “show tables” – To display all the tables present in the current schema
5. “select \* from <table\_name>” – To get all the / few records based on some pre-defined condition.

Agenda

* SQL Language
* Using GUI Mode to access Database
* ACID Properties
* Normalization

SQL stands for Structured Query Language. – Query based Language.

In order to perform any task/operation we need to write a query.

|  |  |  |
| --- | --- | --- |
| Sl No | Operation | Query |
| 1 | Create / Insert | Insert into <table\_name> [(list of column\_names)] values (‘’,numbers, etc., ); |
| 2 | Read | Select \* from <table\_name>; |
| 3 | Update | Update <table\_name> <column\_name1=value1, column\_name2= value2)… where pkey=””; |
| 4 | Delete |  |
|  |  |  |

CRUD Operation – **C**reate, **R**ead, **U**pdate & **D**elete

What is the Difference between SQL & MySQL?

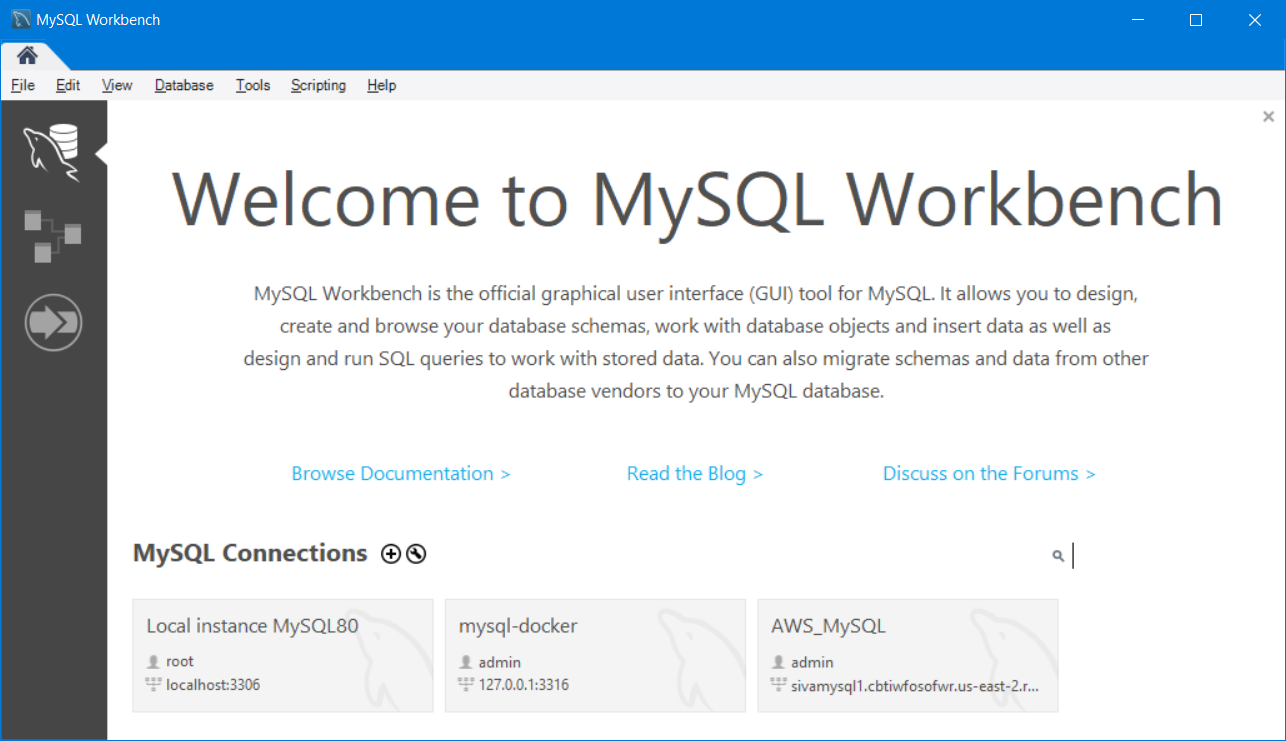
SQL – Structured Query Language (It’s a Language to interact with any Database) [Class]

MySQL – It’s RDBMS. It’s a Database Implementation [Object]

Dbeaver is a General Database client (GUI Mode) which will help to interact with any database.

Accessing the Database using GUI mode

1. Open MySQL workbench.



1. Click on the “Local Instance MySQL8.0”
2. Enter the password for the “root” user

A database or schema is a place to store the data in an organised format.

To Create any Entity in database, the syntax is

Create <entity\_type\_name>

|  |  |  |
| --- | --- | --- |
| **Sl No** | **Entity Type** | **Usage** |
| 1 | Table | It’s the primary Entity used to store the data in a pre-defined format |
| 2 | View | Storing the result of joining two or more tables based on conditions |
| 3 | Trigger | It’s automatically executed SQL code which will be called before or after insert or update or delete operation |
| 4 | Functions | Programming Logic of SQL. A Collection of SQL query with a name and return value |
| 5 | Stored Procedures | Programming Logic of SQL. Set of SQL code with a name and optional return |
| 6 | Sequence | To Generate Numerical Primary key in table |
| 7 | Index | Used for Search & sort purpose |
| 8 | Schema | It’s a place or container which stores other entities. |
| 9 | role | To identify the accessibility of a user |
| 10 | user | Entity to interact with the database |
| 11 | Cursors | Storing & Temporary processing Query output |
| 12 | Materialized views | Similar to views |

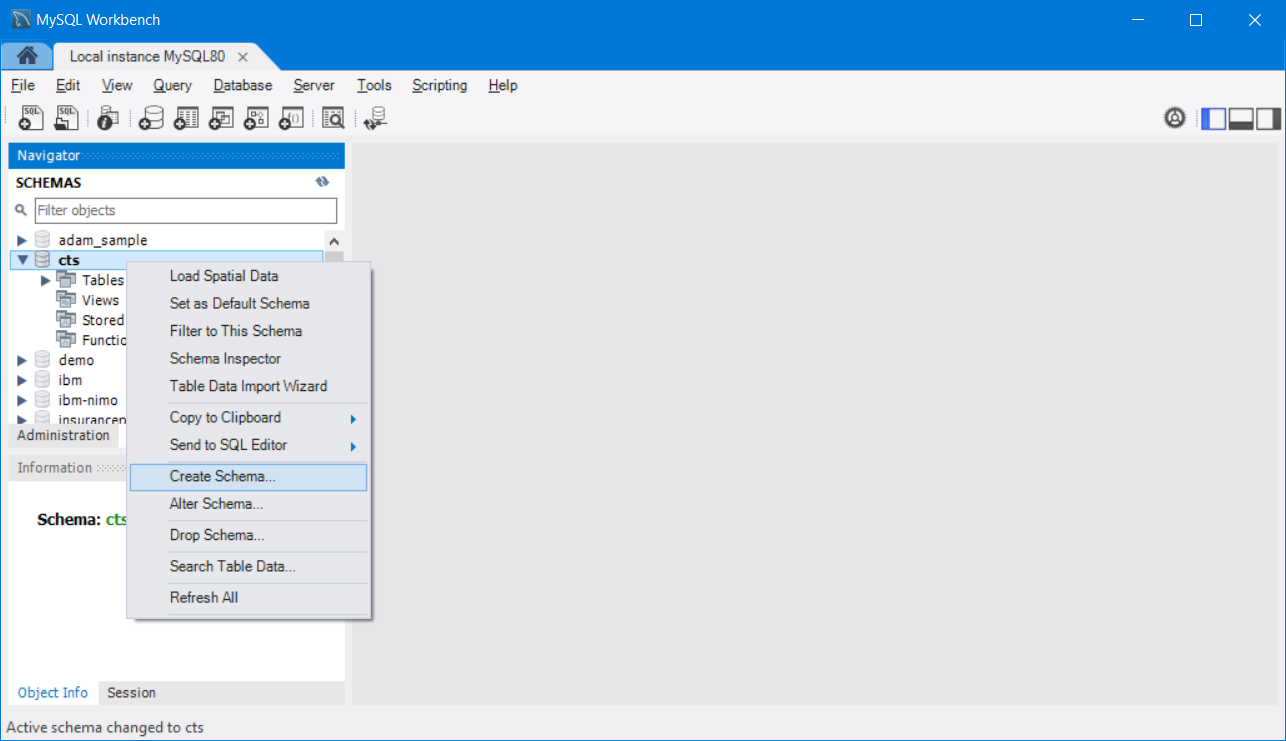
Database, DBMS,RDBMS also will follow naming convention.

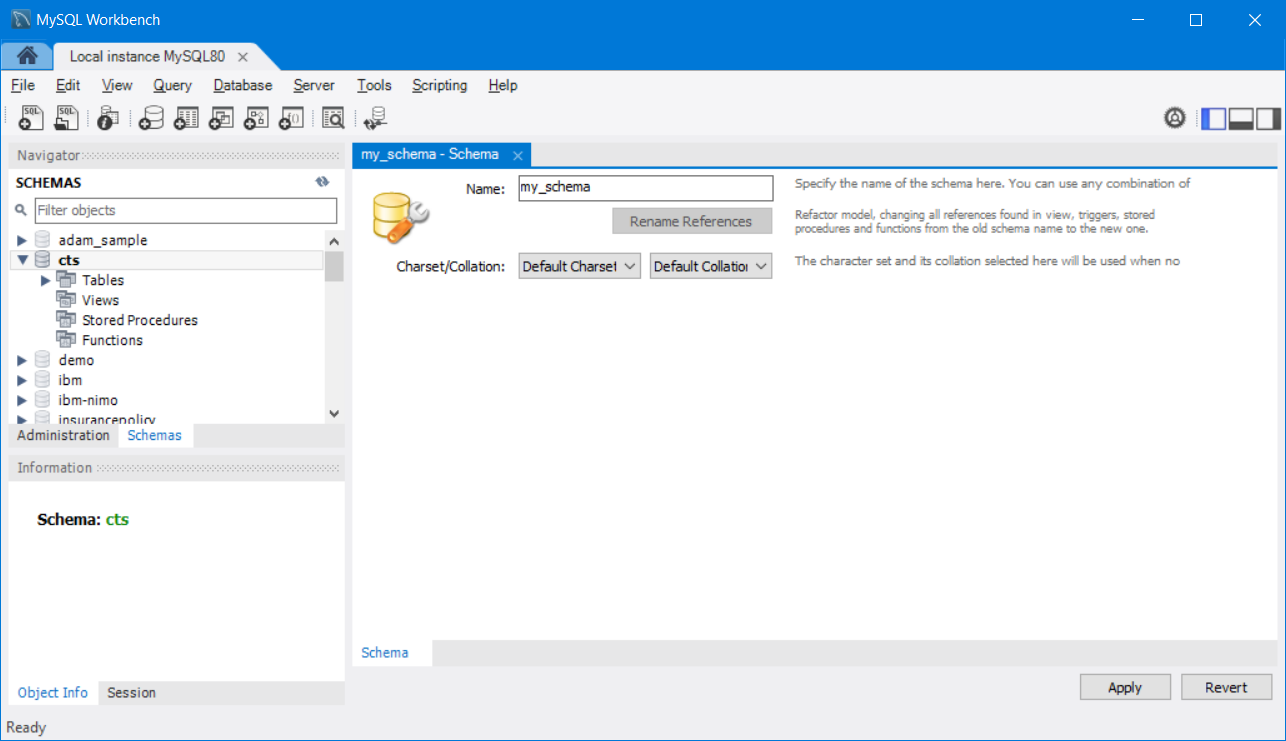
Typically all the table names and column names will be in lower case only.

If the name is more than one word, then words will be separated by underscore character.

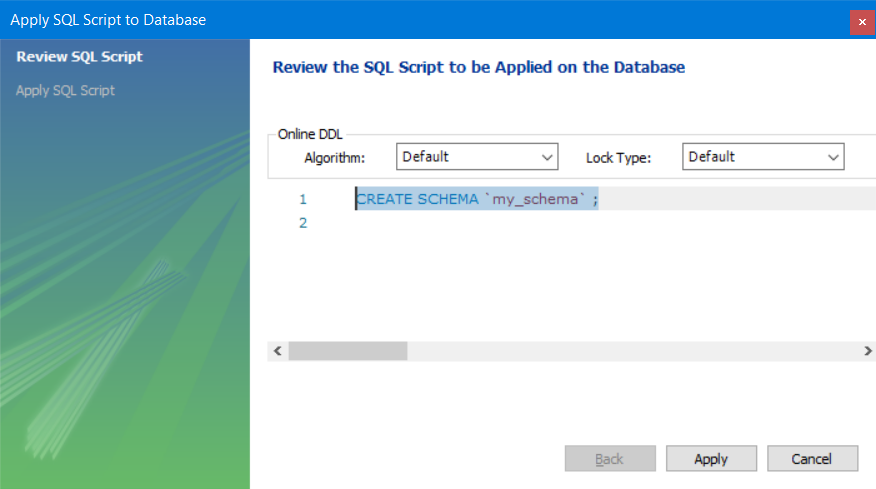
Example table & column names : employee, employee\_name, date\_of\_joining

Creating new Schema in MySQL workbench

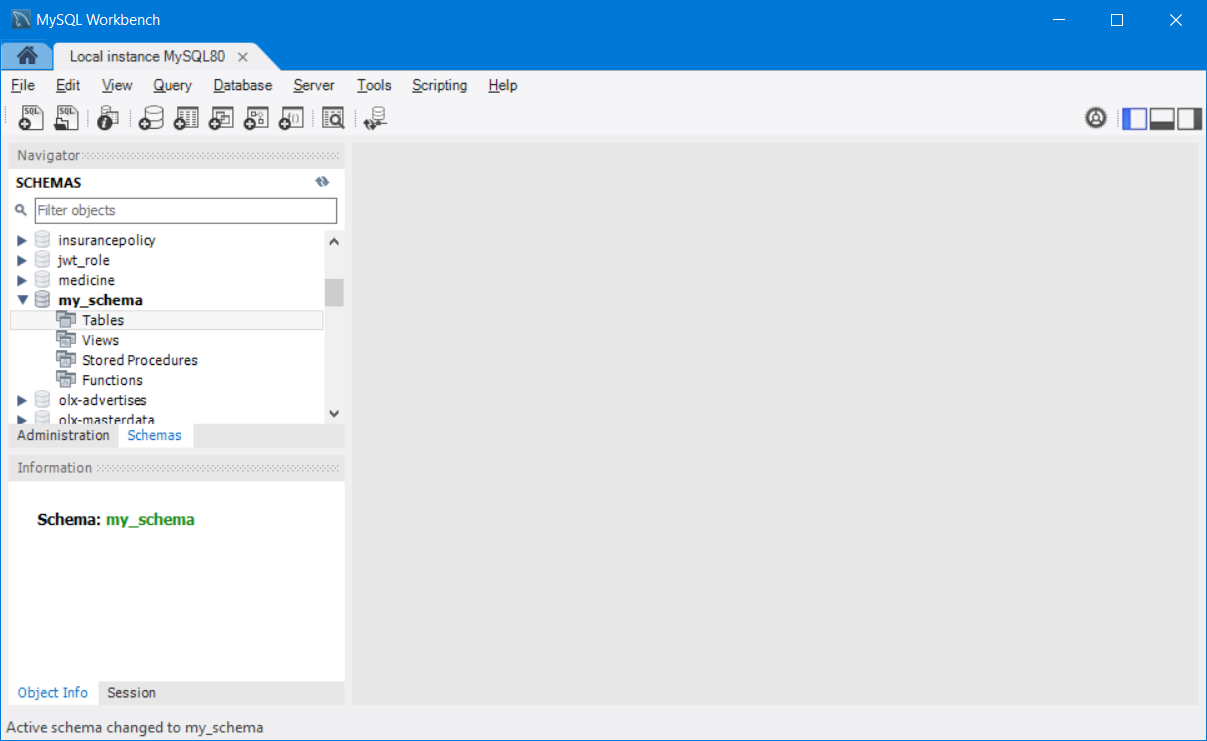




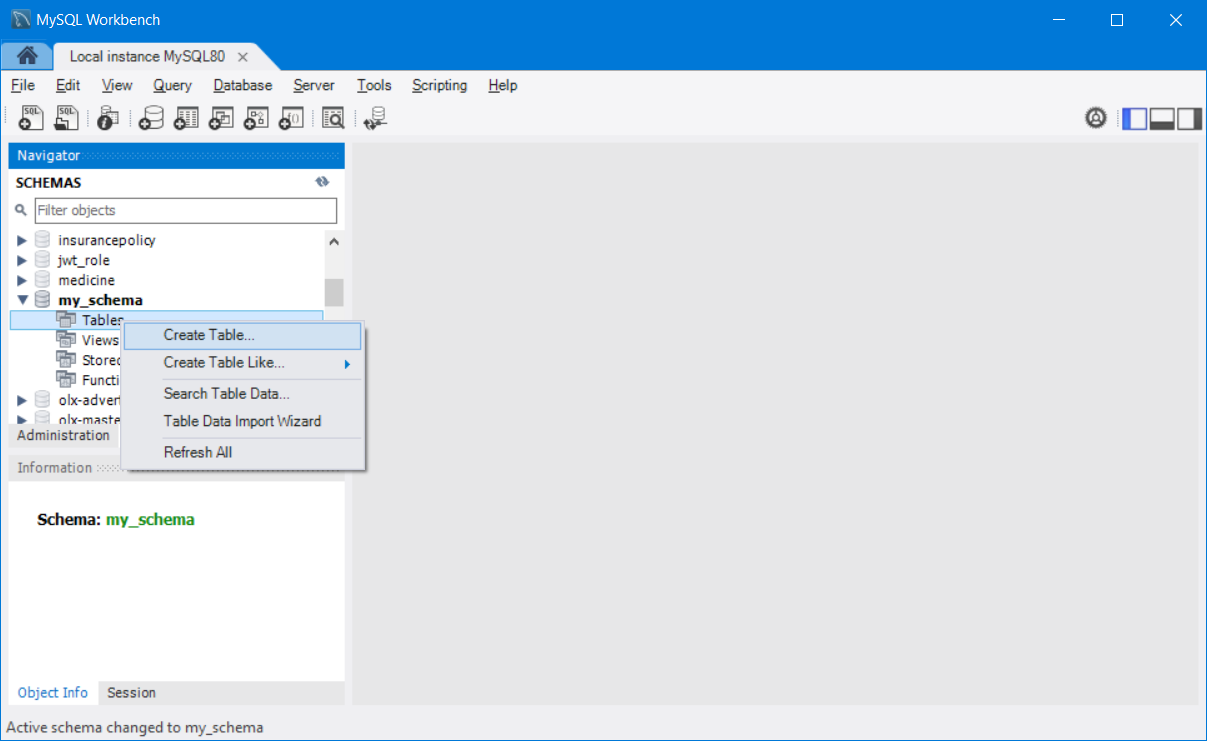
CREATE SCHEMA `my\_schema` ;



Creating a Table in the active/selected schema.







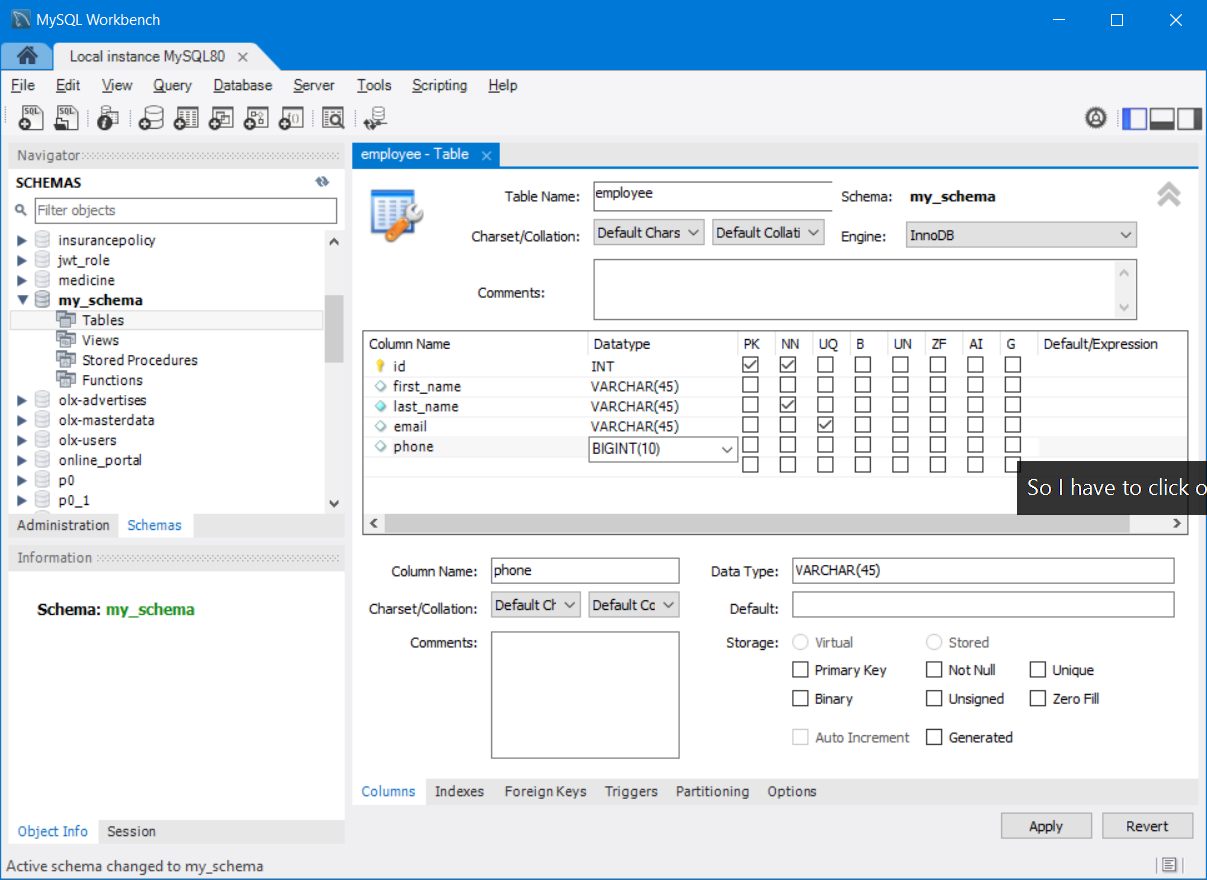
Column Constraints – Enforce a check on the data before inserting (It will test the data for valid type, if it is invalid, it will not allow to store)

Constraints means Check or Condition.

Each Table needs to have a primary key.

Primary Key – It’s a column, where it will not allow duplicate values and it will not allow null value (no value)

|  |  |  |
| --- | --- | --- |
| Sl No | Constraint Name | Purpose |
| 1 | Primary Key (PK) | To uniquely identify a record or row (Unique & Not Null) |
| 2 | Not Null (NN) | To avoid storing null values in a particular column |
| 3 | Auto Increment(AI) | To Generate Sequence and use it as primary Key |
| 4 | Unique (UQ) | To store unique values and to avoid duplicates |
| 5 | Binary (B) | To Store true/false value [0/1] |
| 6 | Zero Fill (ZF) | To fill Zero in numerical columns |
| 7 | Unsigned (UN) | To store only positive numbers in numerical columns |
| 8 | Generated (G) | To specify auto-calculated or generated columns |



CREATE TABLE `my\_schema`.`employee` (

`id` INT NOT NULL AUTO\_INCREMENT,

`first\_name` VARCHAR(45) NULL,

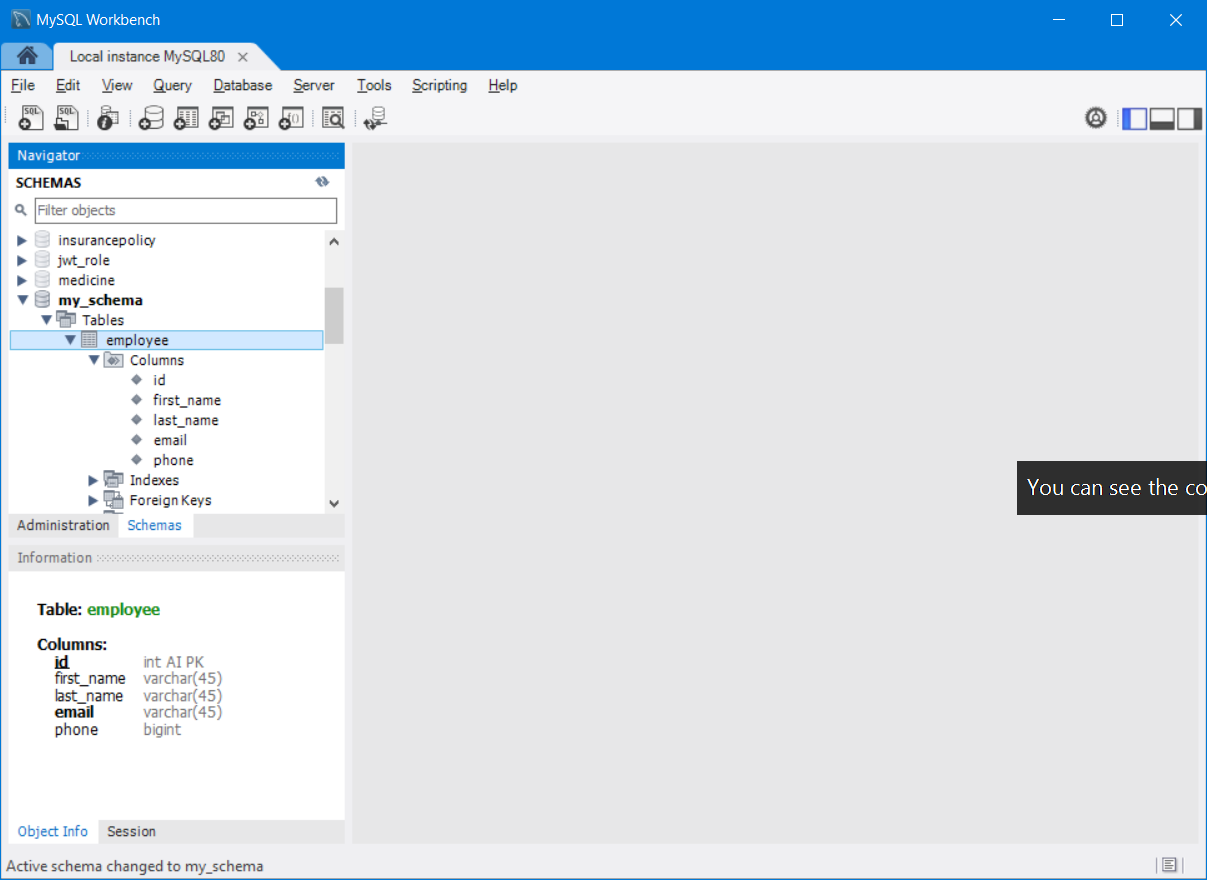
`last\_name` VARCHAR(45) NOT NULL,

`email` VARCHAR(45) NULL,

`phone` BIGINT(10) NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `email\_UNIQUE` (`email` ASC) VISIBLE);



CRUD Operation

1. Create or Insert [ Creating a table/view/trigger/cursor/sequence/role/user/function/stored procedure] Inserting to the table
2. INSERT INTO `my\_schema`.`employee` (`first\_name`, `last\_name`, `email`, `phone`) VALUES ('abc', 'xyz', 'abc\_xyz@gmail.com', '987898786756');
3. INSERT INTO `my\_schema`.`employee` (`first\_name`, `last\_name`, `email`, `phone`) VALUES ('Mark', 'Berg', 'mark@facebook.com', '786798786756');
4. INSERT INTO `my\_schema`.`employee` (`first\_name`, `last\_name`, `email`, `phone`) VALUES ('Sundar', 'Pitchai', 'sundar@alphabet.com', '7867897856');
5. INSERT INTO `my\_schema`.`employee` (`first\_name`, `last\_name`, `email`, `phone`) VALUES ('Satya', 'Nadella', 'satya@microsoft.com', '7898789878');
6. INSERT INTO `my\_schema`.`employee` (`first\_name`, `last\_name`, `email`, `phone`) VALUES ('James', 'Ghosling', 'james@java.com', '879878987');

Update Operation

UPDATE `my\_schema`.`employee` SET `phone` = '8798789877' WHERE (`id` = '5');

UPDATE `my\_schema`.`employee` SET `phone` = '7867987856' WHERE (`id` = '2');

UPDATE `my\_schema`.`employee` SET `phone` = '9878986756' WHERE (`id` = '1');

Types of Insert

1. Partial Insert – Providing value for few columns of table not all the columns
2. Full insert – providing values for all the columns of the table

Types of Edit/Update

1. Partial update – Updating one or more column of a particular row/record
2. Full update – Updating all the columns except the primary key column of a row/record
3. Whole update – Updating a particular column value of all records

Types of Delete

1. Deleting a Single Row (Delete using where clause)
2. Deleting all the rows of a table (delete query without where clause)

Types of Read

1. Full Read – Reading all the records of the table (Select Query without where clause)
2. Partial Read – Reading only the required rows/records (Select query with where clause)

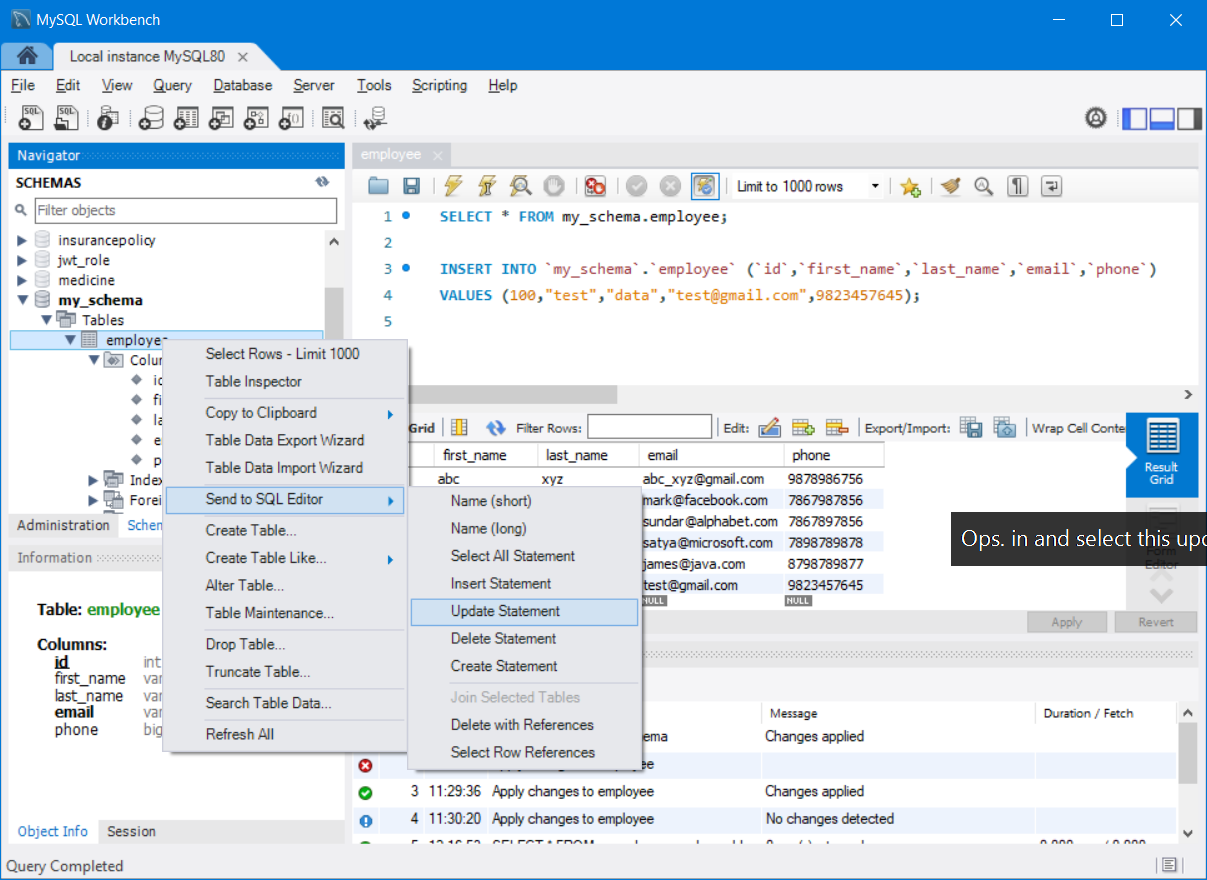
Insert Query Syntax

INSERT INTO `my\_schema`.`employee` (`id`,`first\_name`,`last\_name`,`email`,`phone`)

VALUES (<{id: }>,<{first\_name: }>,<{last\_name: }>,<{email: }>,<{phone: }>);

INSERT INTO `my\_schema`.`employee` (`id`,`first\_name`,`last\_name`,`email`,`phone`)

VALUES (100,"test","data","test@gmail.com",9823457645);



Update Query Syntax

UPDATE `my\_schema`.`employee` SET `first\_name` = <{first\_name: }>, `last\_name` = <{last\_name: }>,`email` = <{email: }>, `phone` = <{phone: }>

WHERE `id` = <{expr}>;

/\* Full read or complete read operation \*/ /\* use ctrl+Enter key to execute the current line query only \*/

select \* from employee;

/\* Partial read operation\*/

select \* from employee where id>4;

SQL Queries – (Structured Query Language)

DDL – Data Definition Language (Create, Alter, Drop,Truncate)

**DDL**

Data Definition Language statements are utilized to define the database schema or skeleton. It is how we implement the design structure of it. Some of the keywords in this sublanguage are:

* CREATE, to create new objects or tables.

CREATE TABLE TABLE\_NAME (C\_NAME C\_TYPE C\_SYZE [NULL | NOT NULL], [CONSTRAINT])

* ALTER, to modify existing objects or tables.

ALTER TABLE TABLE\_NAME [ADD | MODIFY | DROP] C\_NAME

ALTER USER IDENTIFIED BY PASSWORD

* DROP, to delete existing objects or tables.

DROP TABLE TABLE\_NAME [CASCADE]

* TRUNCATE, to delete all the data existing within a table leaving the skeleton of the table only.

TRUNCATE TABLE\_NAME

This is similar to performing DELETE TABLE with no where clause, the key difference is that TRUNCATE commits at the end of the operation.

Drop will delete the entire table along with it’s content.

Truncate deletes all the data of the table leaving the structure as it is. It commits at the end.

Truncate === Delete All Records

<https://www.javatpoint.com/dbms-sql-command>

<https://www.guru99.com/sql-commands-dbms-query.html>

<https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/>



DML – Data Manipulation Lang

Constraints

**Constraints (or) Integrity Constraints**

We can put integrity **constraints** on specific columns in our database when defining tables, which allow us to enforce the schema by ensuring consistency and integrity of the data in the table. The different constraints are listed below:

* PRIMARY KEY (Unique & Not Null) – No Duplicate & No Null value
* FOREIGN KEY – Referential Integrity – Which checks the data available in parent table before adding it to a child table.
* NOT NULL – Will not allow null value for that column
* UNIQUE – No duplicates allowed in Unique Column
* CHECK – It will check for given expression (age>0<100)
* DEFAULT – “trainee”
* AUTO INCREMENT – Helps to create a sequence and use it as a primary key of the table. (This will automatically create a trigger to insert to id value using sequence.current value)

Composite – Primary Key – Two or more columns to gether will act as a primary key. This is called Composite Primary Key.

Personal Details

Account Details

Address Details

Order Details

Billing Details

Shipping Details

Rating Details

Feedback Details

**Normalization**

[**Normalization**](https://en.wikipedia.org/wiki/Database_normalization) refers to an optimization process of structuring a relational database in a way that *reduces redundancy* of data and improves data integrity and consistency. There are many different normal forms, which relate to the degree to which a database has been normalized. We will look at the first three normal forms, each of which build upon the previous:

* 1NF - must have a primary key, no repeating groups, and atomic columns
* 2NF - must already be in 1NF, plus have no partial dependencies
* 3NF - must already be in 2NF, plus have no transitive dependencies

The first normal form enforces that a table **must**:

* Have a primary key
* Each column should be as granular as possible (e.g. "Name" column should be broken up into: "First Name", "Last Name", "Middle Name", etc..)

To be in second normal form, a table must **also**:

* Cannot have columns that are dependent on only one part of the key
* If there are no composite primary keys, you are automatically in 2NF

Finally, to get to third normal form, a table must **also**:

* Not have transitive dependencies
* This means that if column C relates to column B which relates to column A which is the primary key, this is not in 3NF because C is related to the primary key but indirectly (it is a transitive dependency)

To advance into higher normal forms, we typically "break up" tables into multiple tables and relate them to each other via foreign keys.

A good way of remembering these normal forms in order is to remember the legal proceeding of swearing to tell the truth, the whole truth, and nothing but the truth. In relational databases, we must have **the key (1NF), the whole key (2NF), and nothing but the key (3NF)**.

Interacting with Database using JDBC (Java)

JDBC – Java Database Connectivity

Assume, you bought a new printer for your laptop.

* 1. Install Drivers
  2. Establishing the connection between the laptop & printer
  3. Load paper & do test print
  4. Print actual document/image
  5. Power off the printer

Same steps for interacting with Database using Java (JDBC)

1. Load & Register the Driver (Depends on the DBMS used)
2. Establish the Connection between Java and RDBMS
3. Create Query to Execute (Statement, Prepared Statement, CallableStatement)
4. Execute the Query, Store the Result in ResultSet and process it
5. Close all the resources.

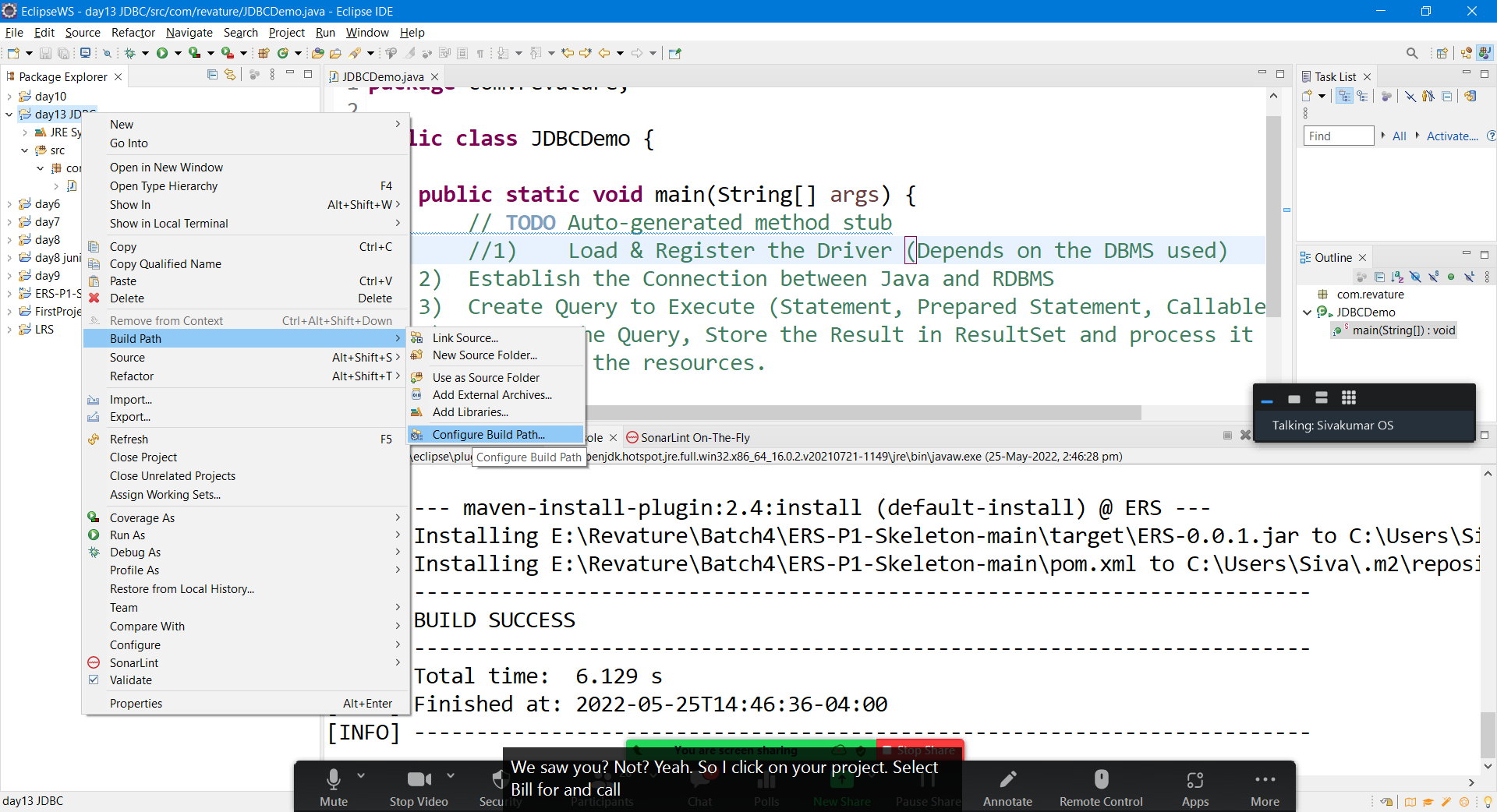
RDBMS – Is working as a Client/Server

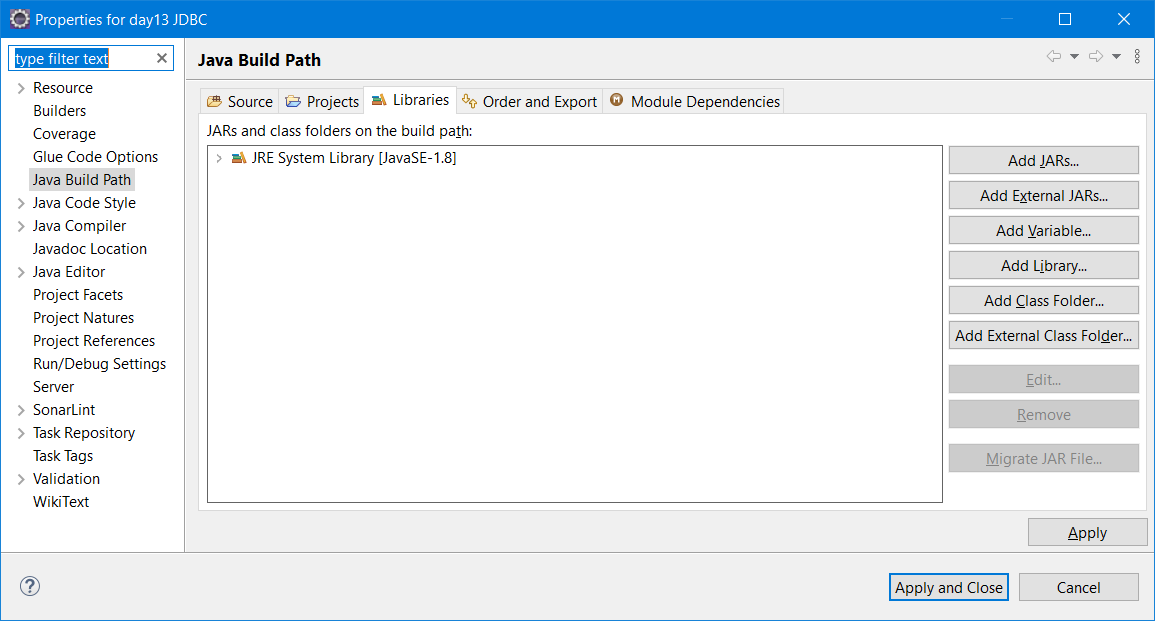
If you like your gmail, what details you need?

* 1. Username & password (login credentials)
  2. URL to access the Resources (Browser)

Download MySQL-ConnectorJ from the following site

<https://dev.mysql.com/downloads/connector/j/> (Zip file)





Select the MySQL Driver jar from the below location

C:\Program Files (x86)\MySQL\Connector J 8.0

