Google.com – Address bar (Request object – Requesting URL – resource)

Client – Server Technology (http method – get /post )

Sending the data from client to server

SQL – Structured Query Language

Data – Information

Data – Structured Data, Un Structured Data

Structured Data -- Name : ABC Age: 25 Salary:35000

Unstructured Data --- 25985, “king”, image, audio, video, 987898.987,

Post – Text, Links with Text, Links with Images, Video, Audio

Flat File System – Every OS [Operating System] uses a file system called flat file system

Text content – unstructured way. ( Notepad, WordPad, Textpad, Notepad++, vi Editor ….)

Filename with extension (filename.txt, filename.rtf, filename.doc ….)

Main Challenges

1. It’s not easy to search the data
2. It’s not easy to fetch part of the data

Database – Organized way of storing data

In any database – Entity

Entity – Table, Query, View, Materialized view, Cursor, Sequence, Constraints, Triggers, stored procedures, functions

SQL – It’s a language for the DATABASE

SQL – Structured Query Language

Keywords – select, insert, update, delete, alter, create

DBMS – Database Management System

Oracle, MySQL, DB2, MSSQL ……

RDBMS – Relational Database Management System

In Maths, Relational Theory – Venn Diagram – Set, Union, Intersects

MySQL – Open Source RDBMS

1. Community Edition (Totally free)
2. Commercial Edition (More Features – Paid version)

Lotus – Father of Excel (Spread Sheet)

Foxpro – Father of DBMS (A Programming Language with built-in database support)

MySQL – Database Server

1. MySQL Server
2. MySQL Clients – Command Line Client (DOS Based) , GUI Client (Workbench) , MySQL Shell (JS Based)

Command Line Client – CUI (Character User Interface) - Everything is performed using commands

SQL Queries – SQL Commands

Localhost – ip address – 127.0.0.1

Localhost:8080 === 127.0.0.1:8080 (both are same)

Default port number for MySQL server is 3306

Default user name for MySQL Server is root

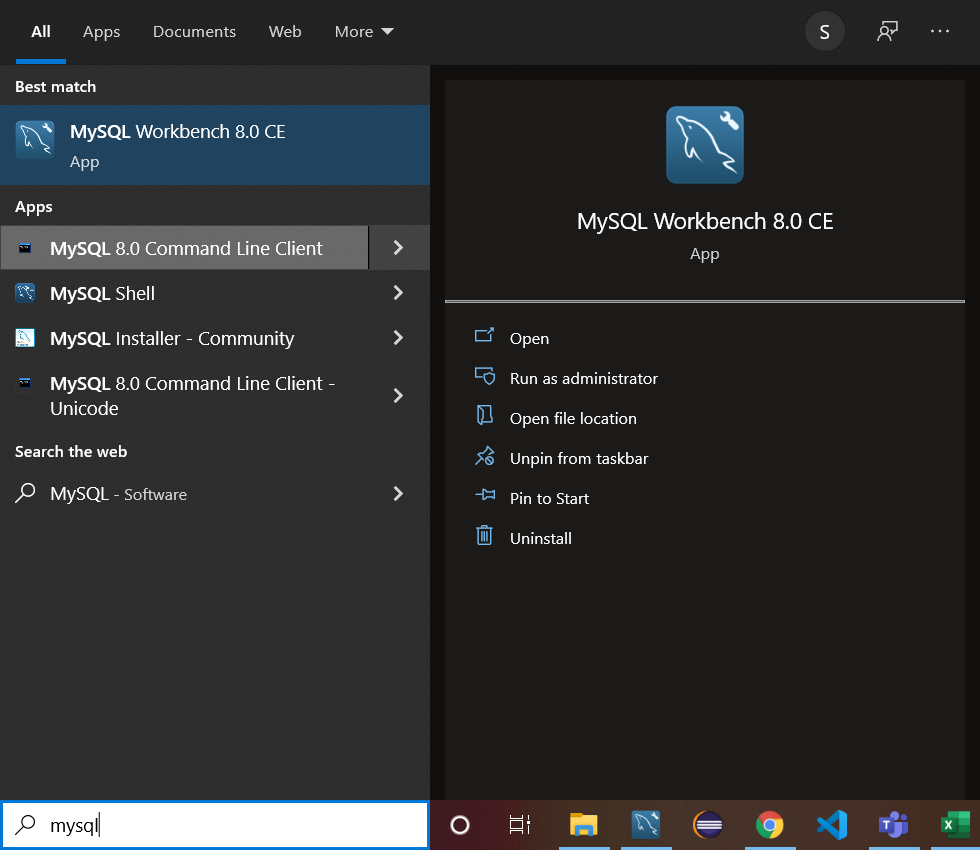
Database server (MySQL) can be installed in your laptop or it can be installed in a remote system (

In MySQL, the commands are case in-sensitive but the contents of “Entities” are case sensitive

Accessing the MySQL database Server

Method 1: Using Command Line Client

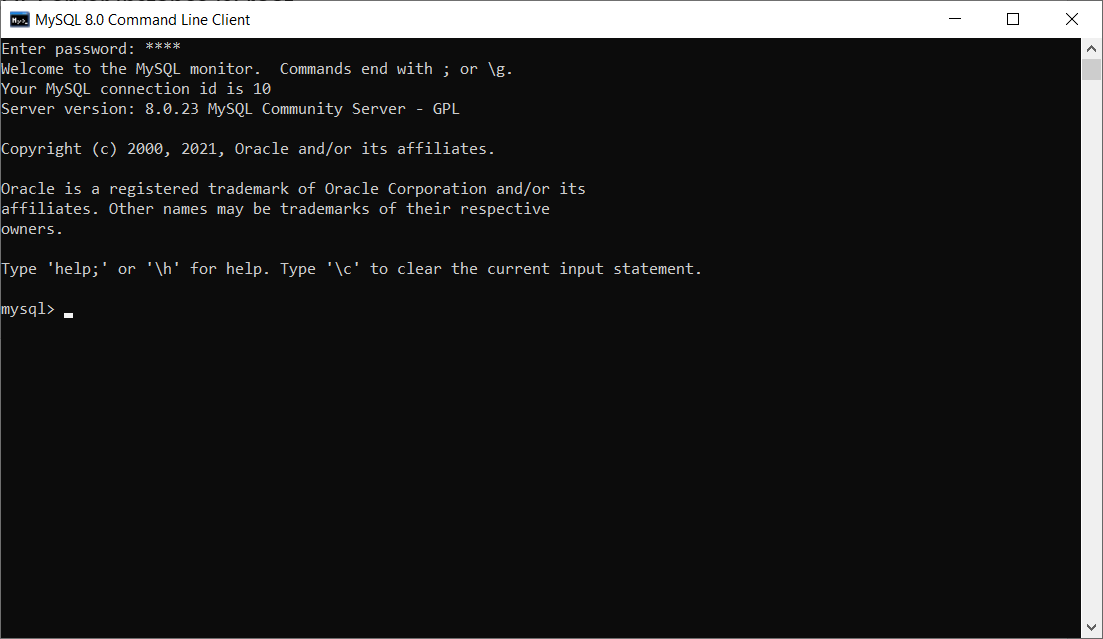
Step 1) Open Mysql command Line client by searching in the search box



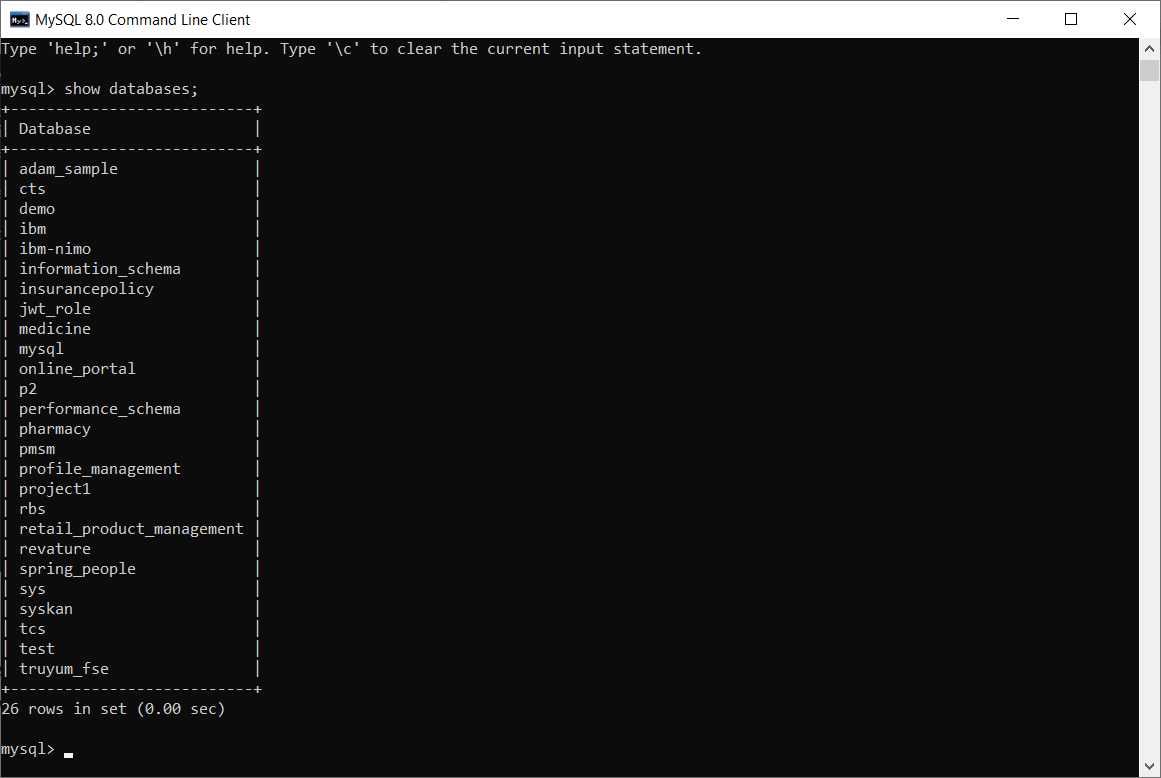
Step 2) Enter the password (which you have given while installing MySQL RDBMS )

Default username for MySQL is: root

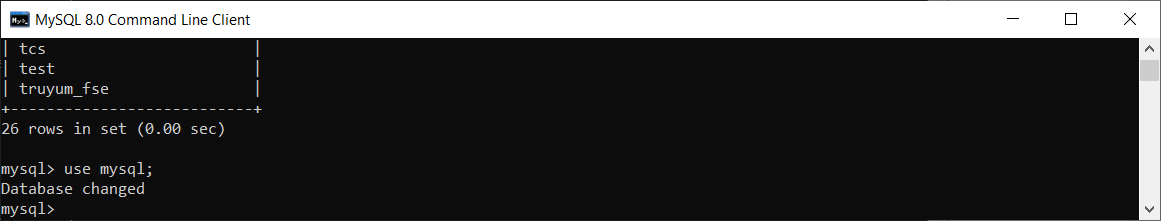
Password for my MySQL Server instance is: root



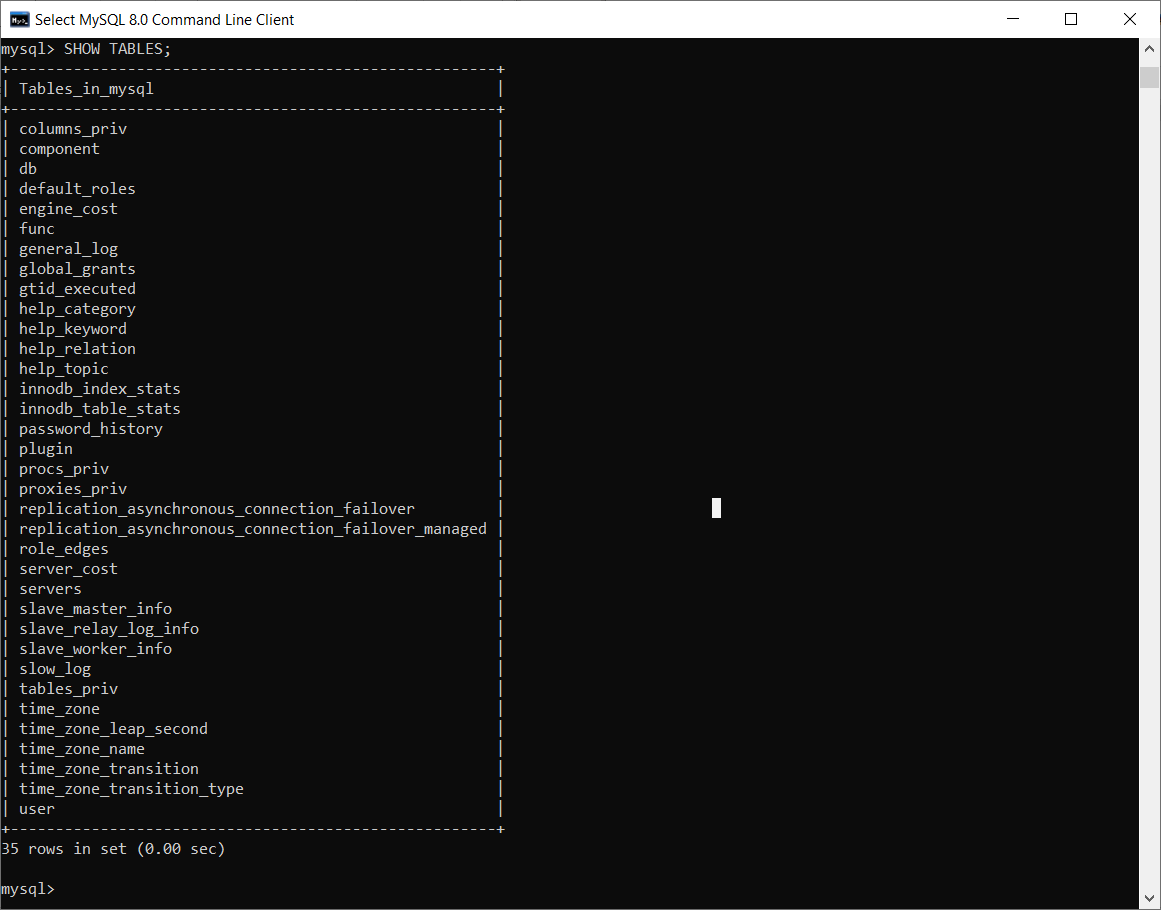
Step 3: Enter the command “show databases;” which will display all the databases available in your MySQL Server



Step 4: Connecting to a particular DB can be done using the following command “use mysql;”



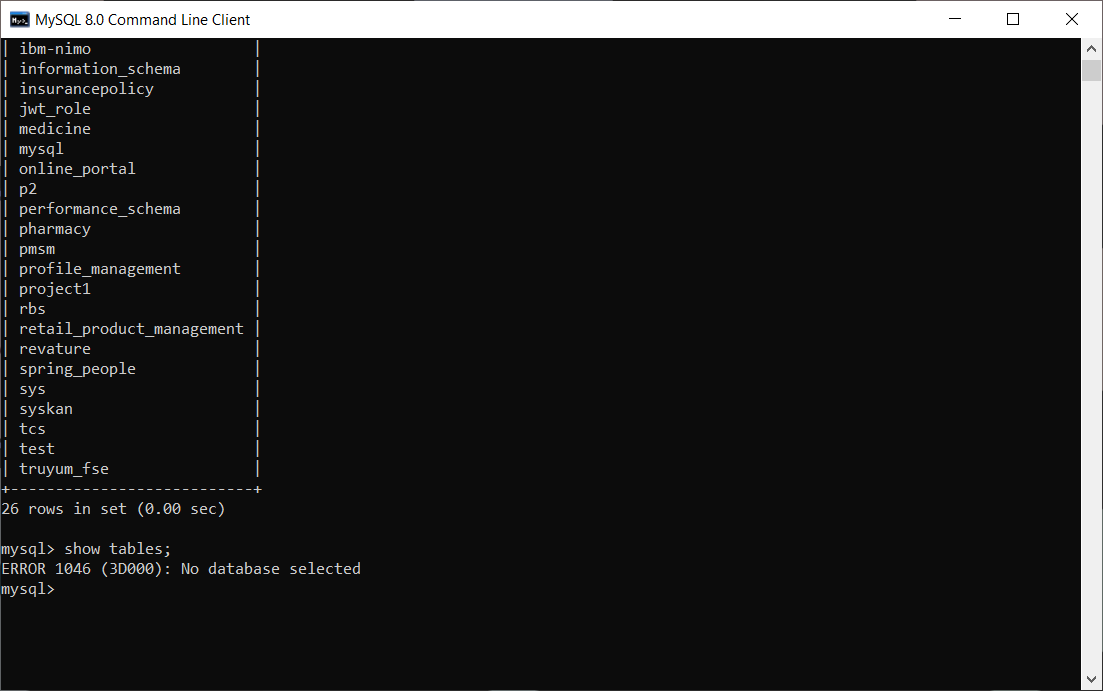
Step 5: To display all the tables in the selected database use the following command “show tables;”



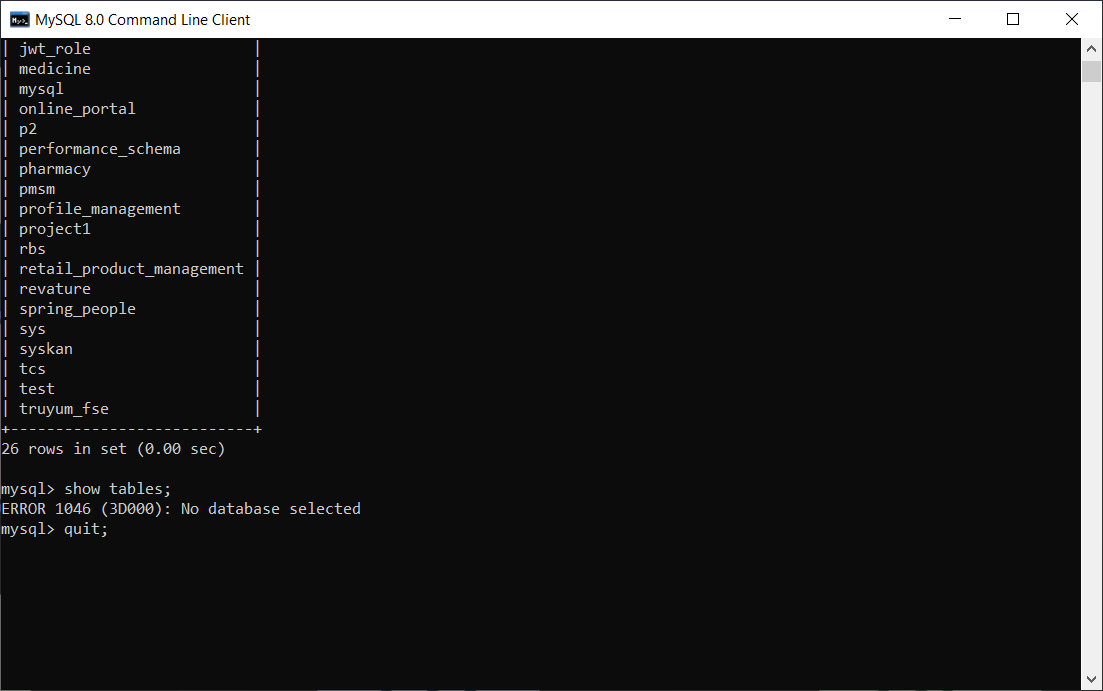
Step 6: Selecting a particular table – “select \* from user;”



Step 7: Resolving Error --- “giving the ‘show tables;’ command without selecting a database will give the following error.

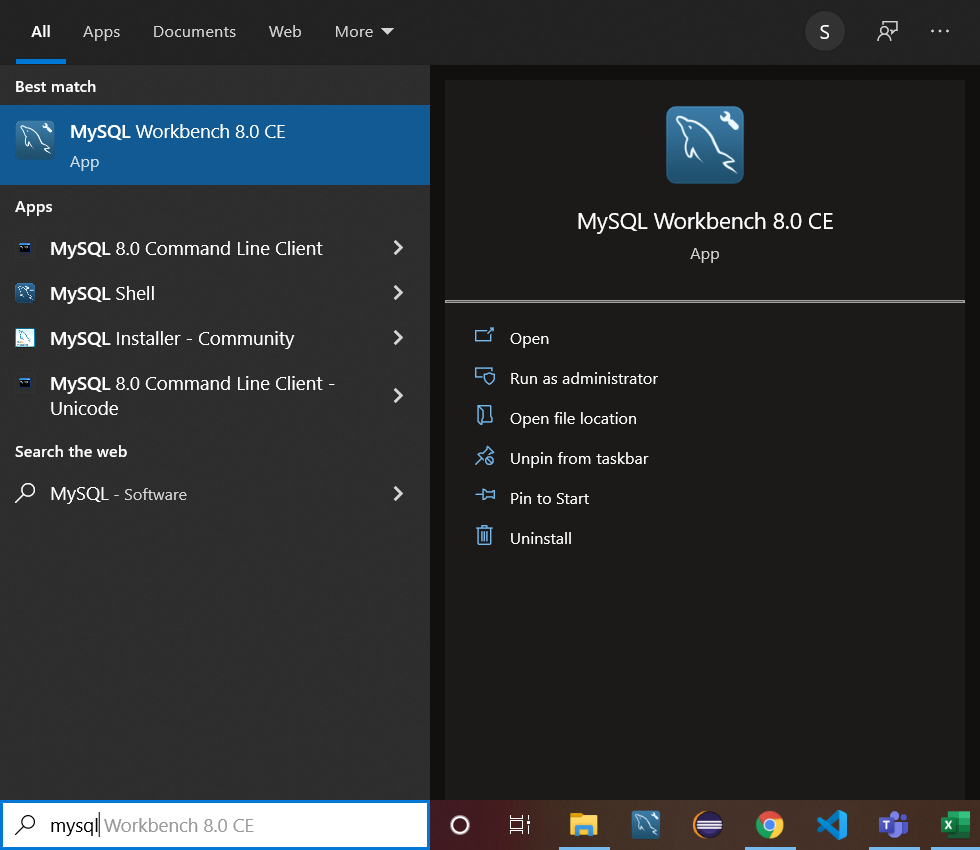


Step 8: To close the Command line client type the following command “quit;”

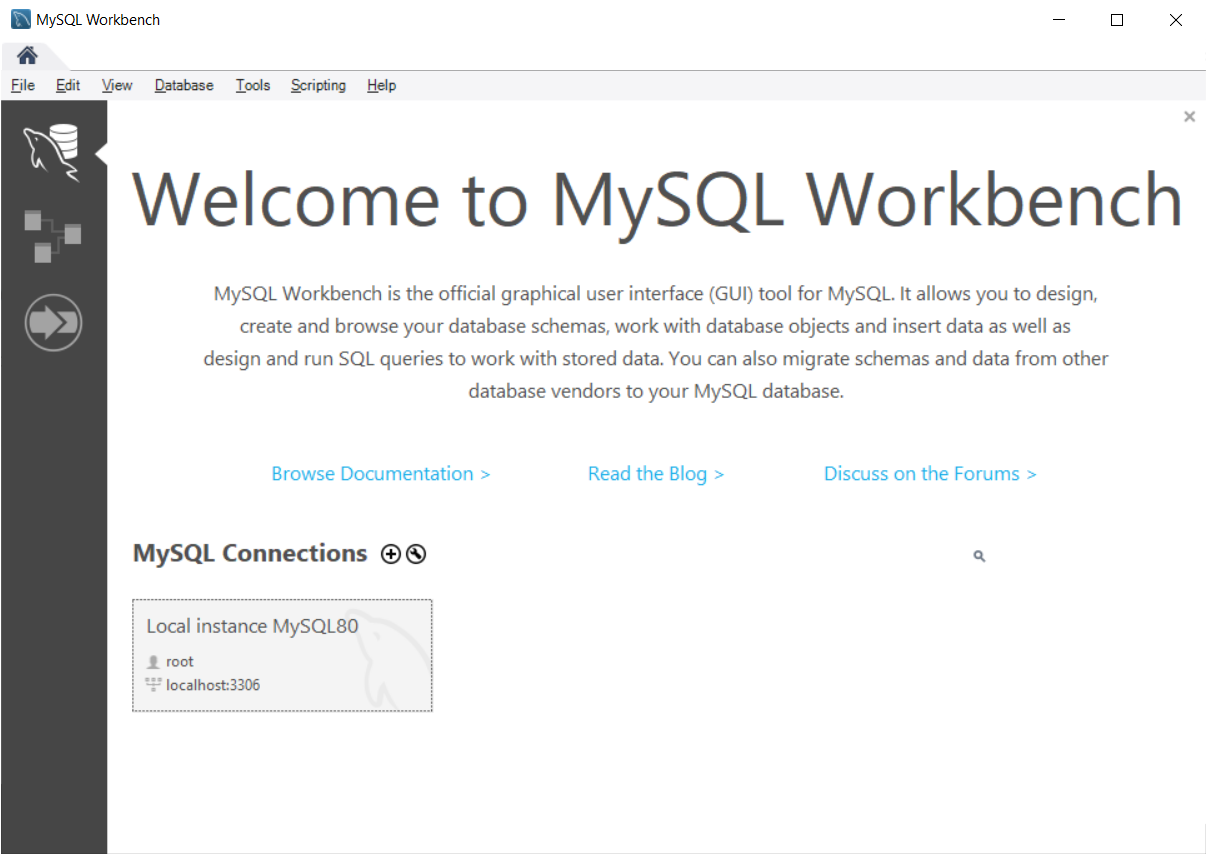


Connecting to MySQL server using WorkBench (GUI - Client)

Step 1) Search Mysql Workbench in search box

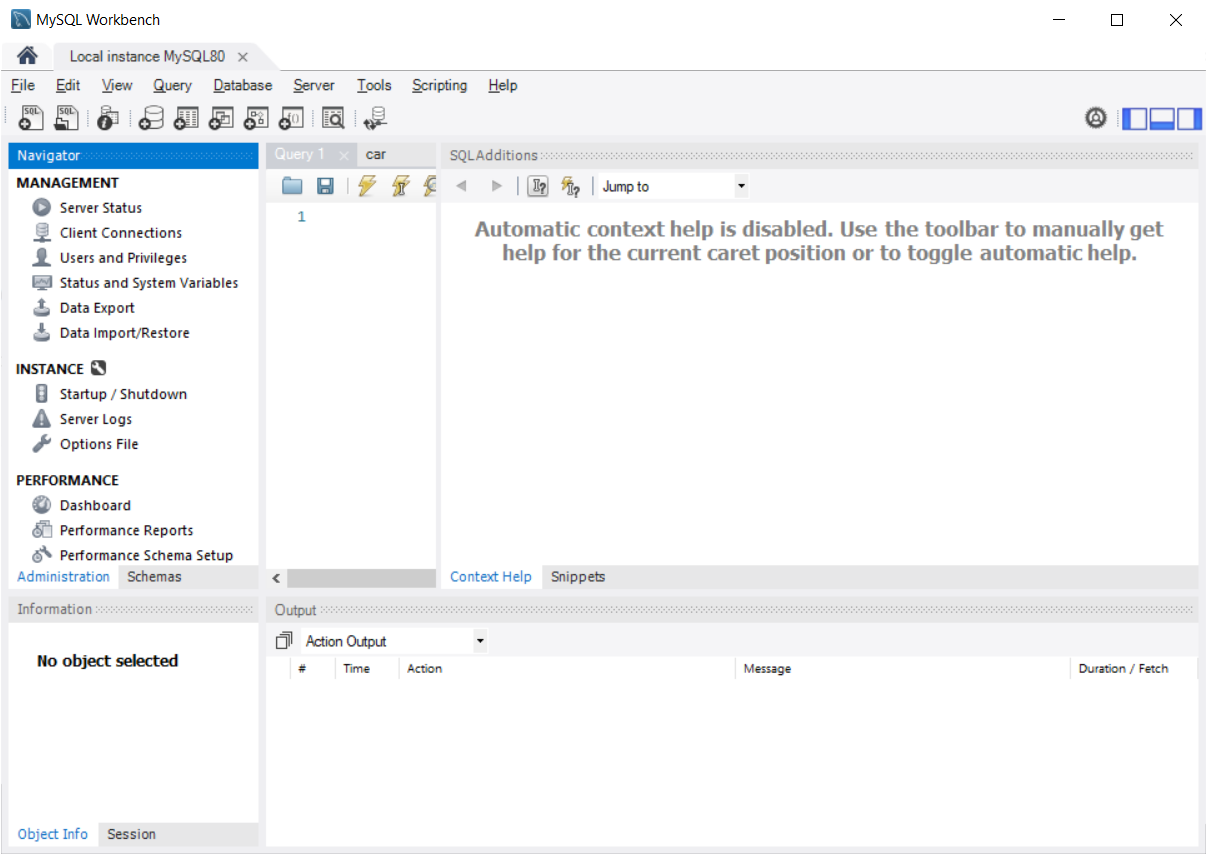


Step 2) select the local instance MySQL 8.0 as shown below



Step 3) Enter the password “select the checkbox to store the password in vault”

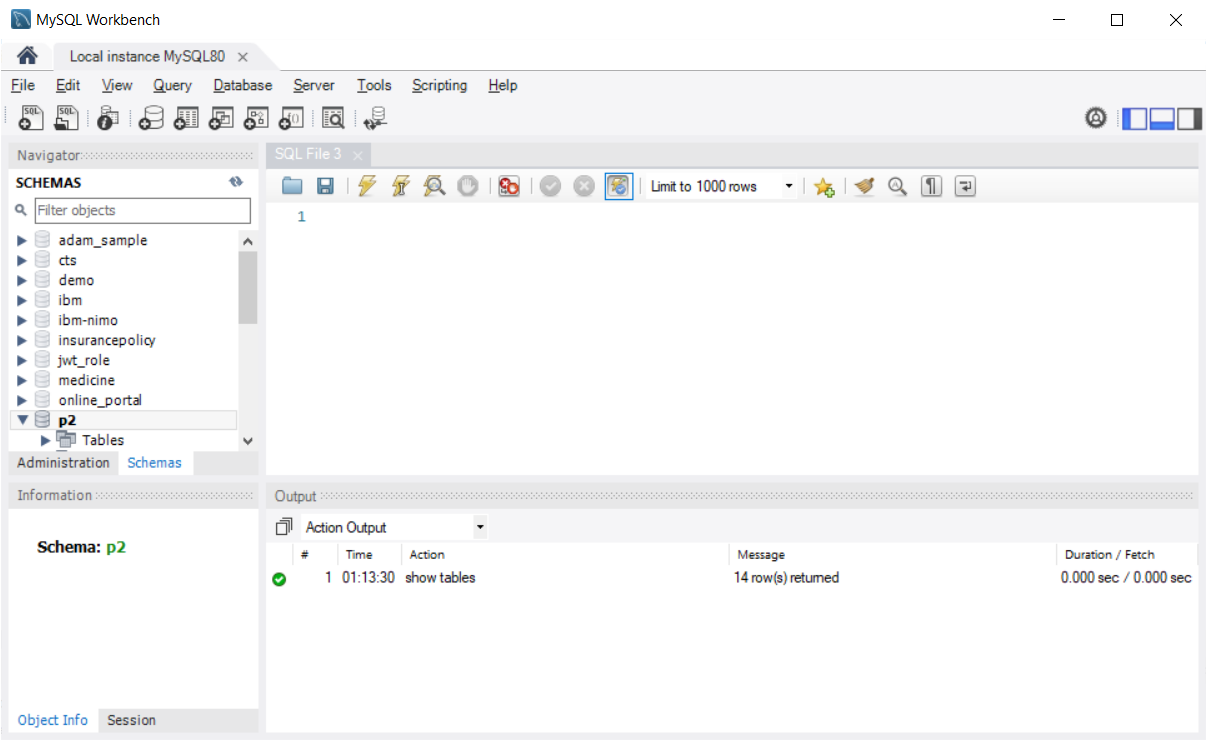
Step 4) Below is the landing page of workbench



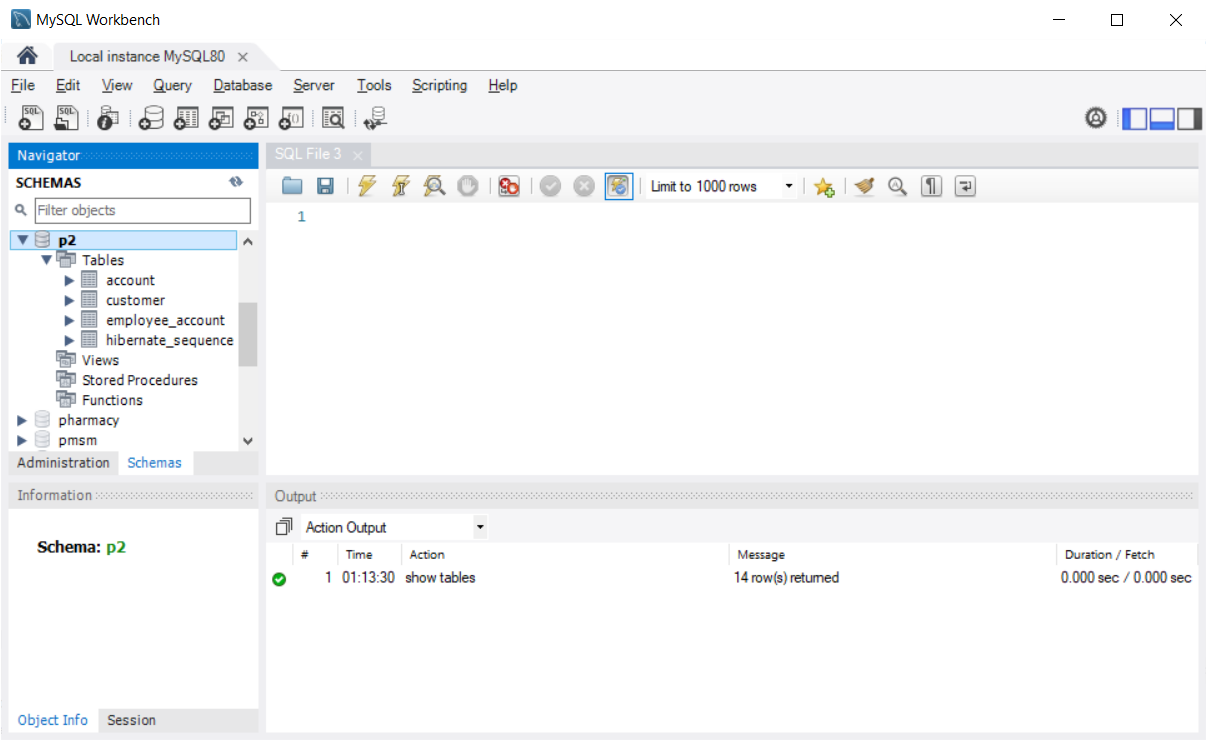
Step 5) Select the “Schemas” tab



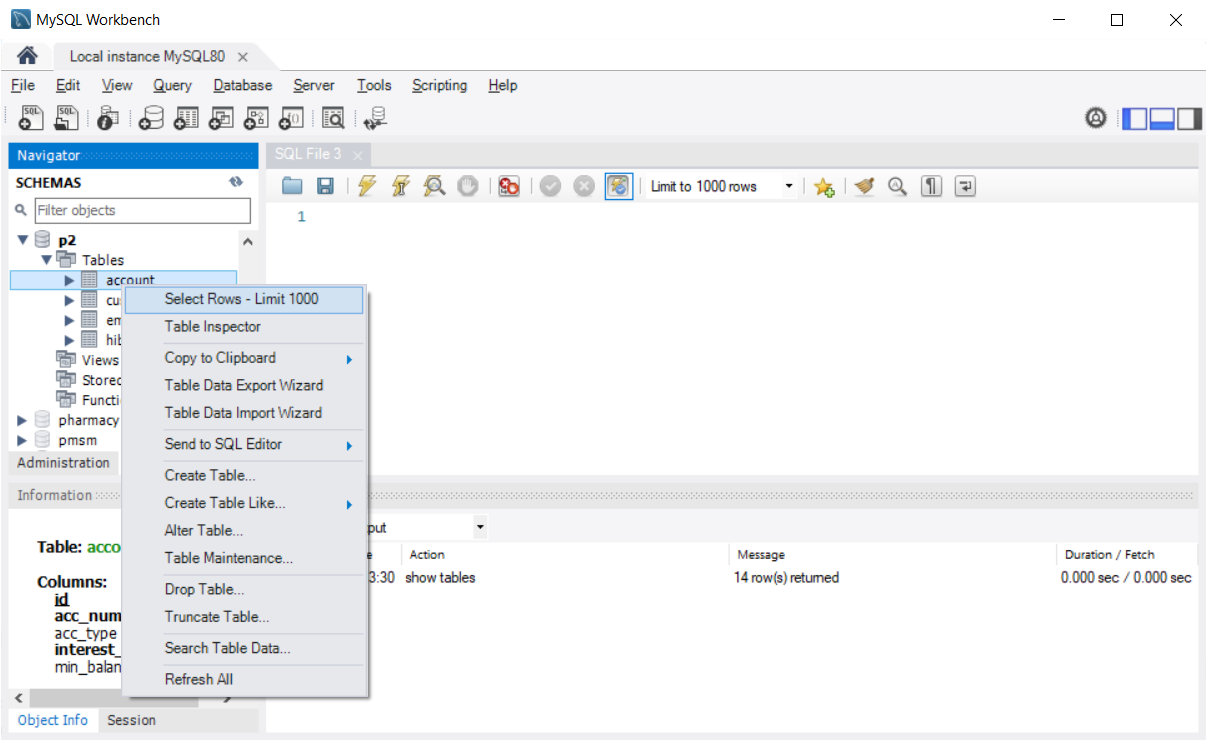
Step 6) double click on the database name to select it.

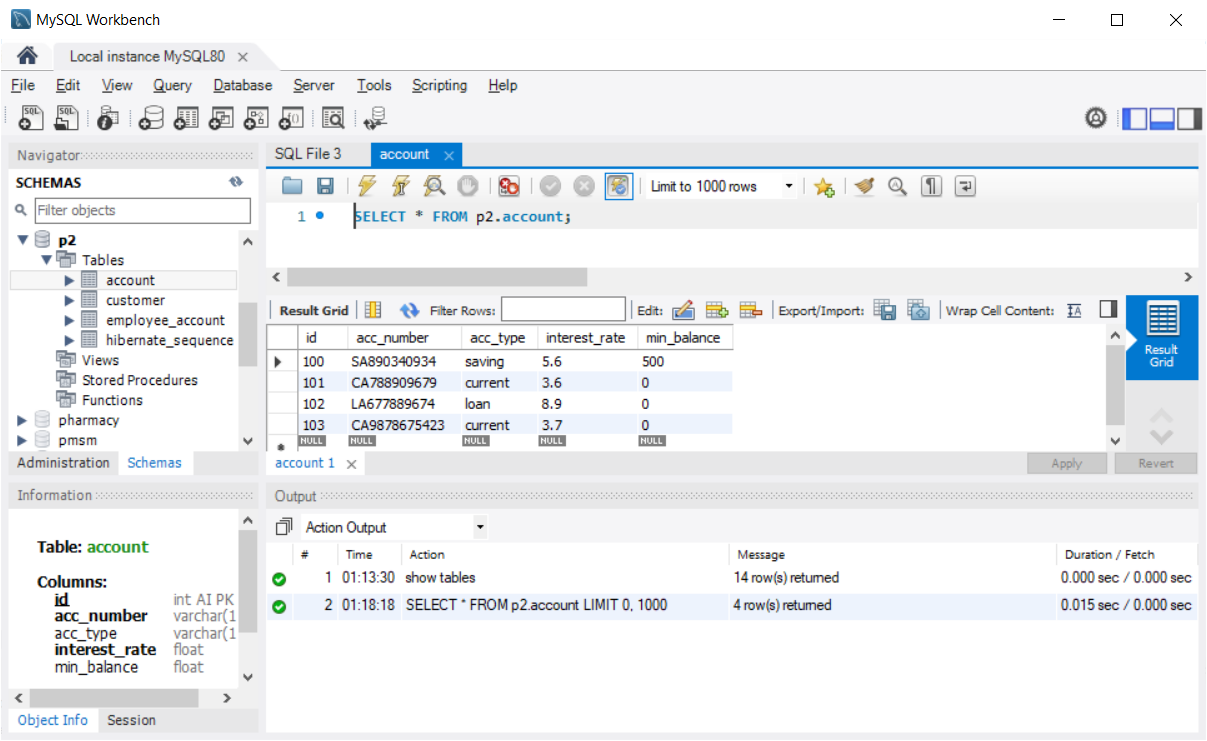


Step 7) Expand the “Tables” section to see all the available tables as shown below

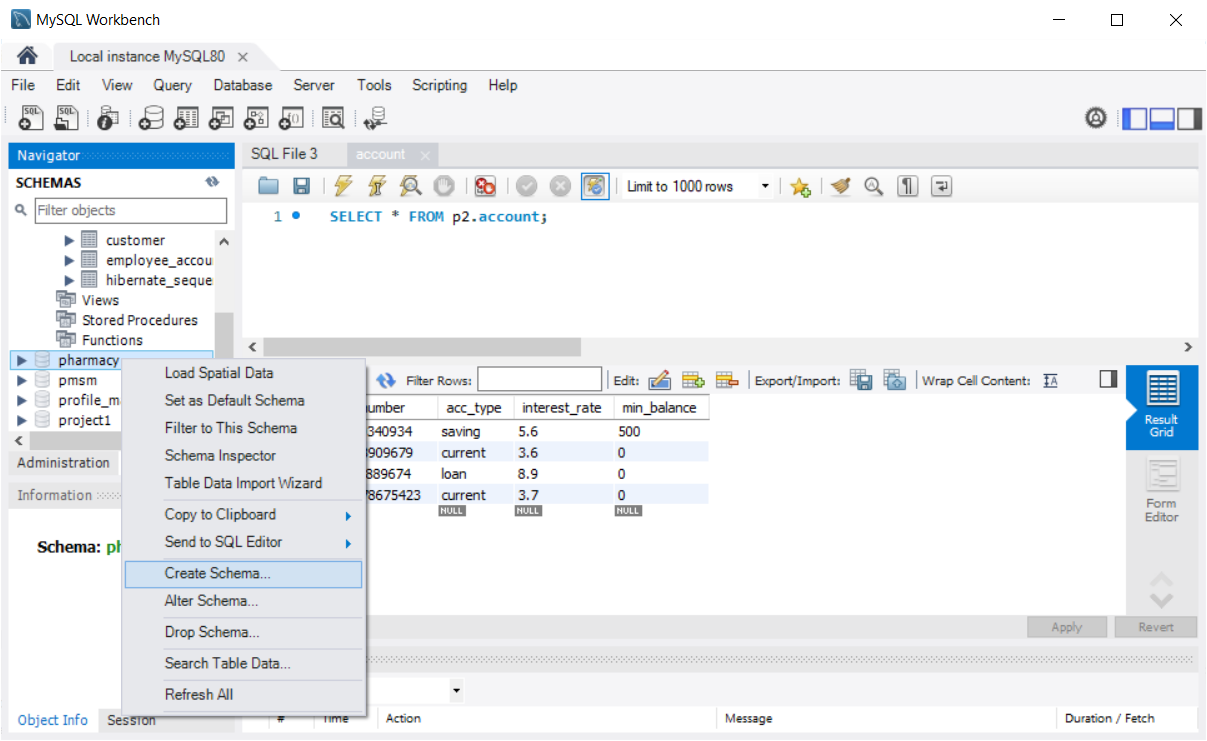


Step 8) To get the contents of the table, right click the table and choose “select Rows – Limit 1000” option.

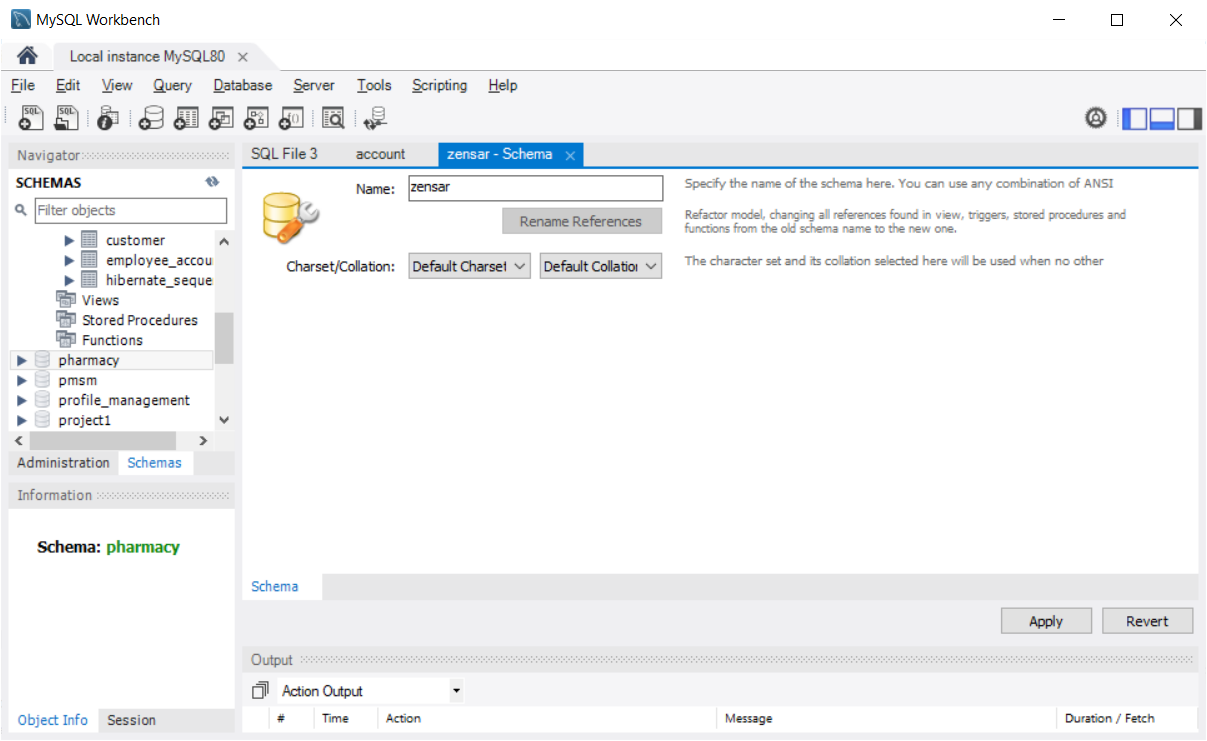




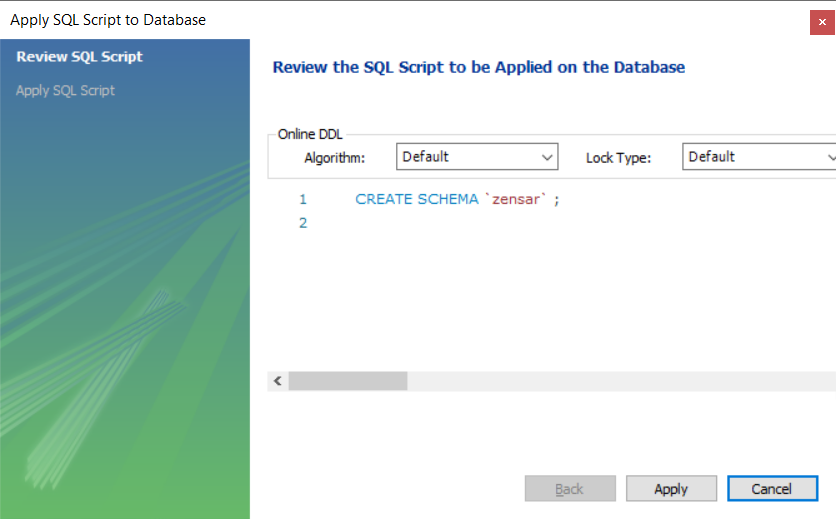
Step 9) To create a New Schema “Right Click on any Existing Schema/Database” select “Create Schema” option as shown below



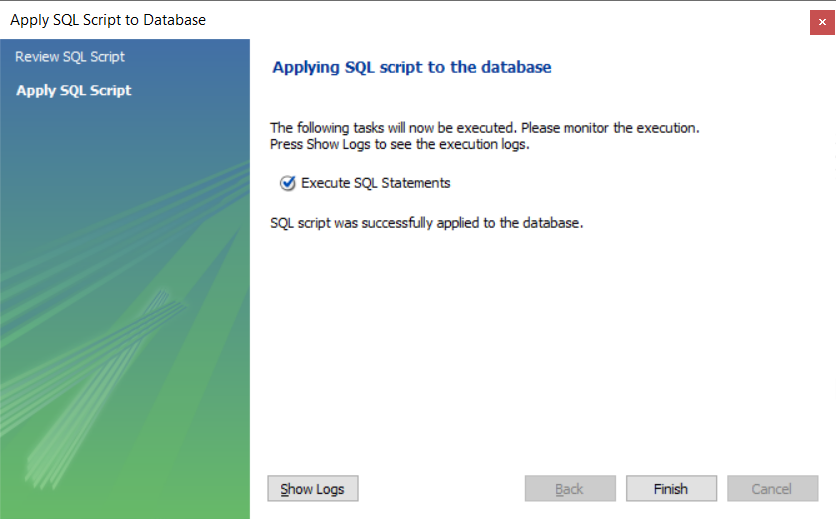
Step 10) Provide the name of the “Schema” in the given text box as shown below and click “Apply” button



Step 11) Click “Apply” button again in the “Apply SQL Script to Database” window



Step 12) After successful execution of the given query we will get the result of the execution and click “finish” button.

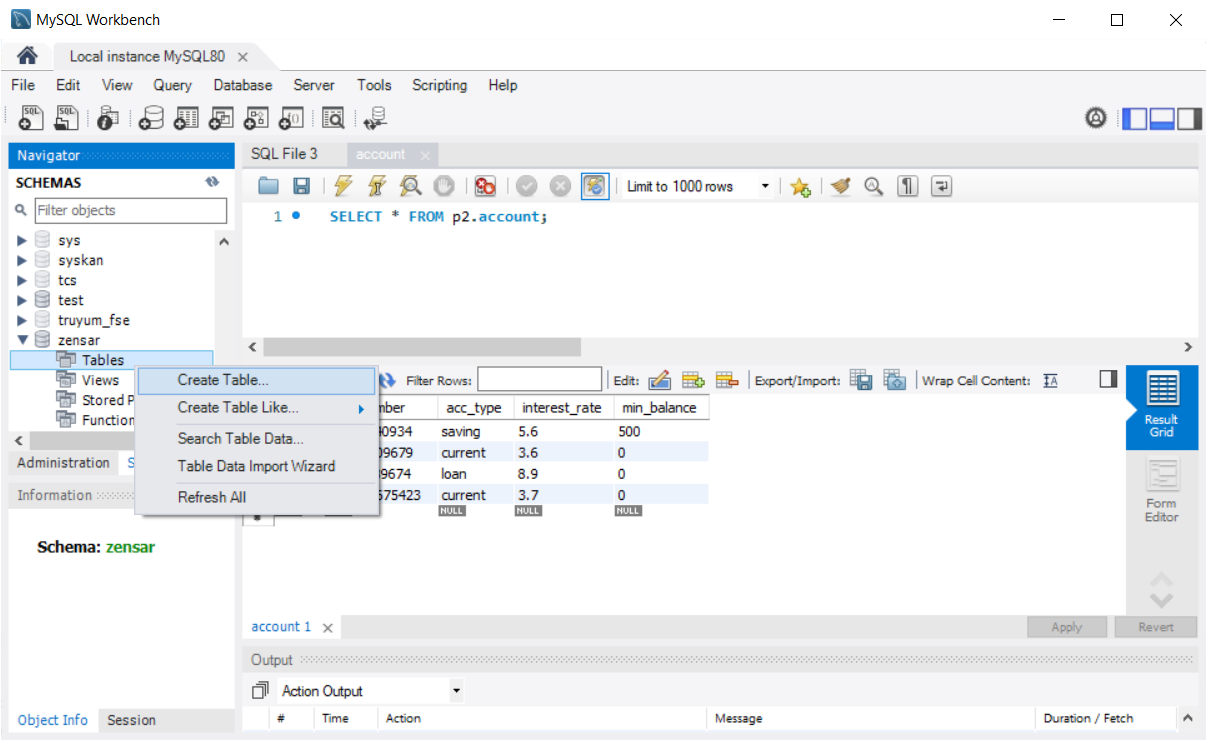


Step 13) Navigate to the newly created schema and check it’s content.



Step 14) Create a table called “employee” with the following columns (id, name, age, salary)

Right click on the tables section and select “create table” option as shown below



Step 15) Enter name of the table and column details as shown below and click “Apply” button

There should be a primary key for each table (It is compulsory in RDBMS)

Primary key is a unique and not null columns (Duplicate & null value is not allowed)

Primary keys can be Number based( INT) or string based (varchar)

Different Constraints

PK – Primary Key

NN – Not Null

UQ – Unique

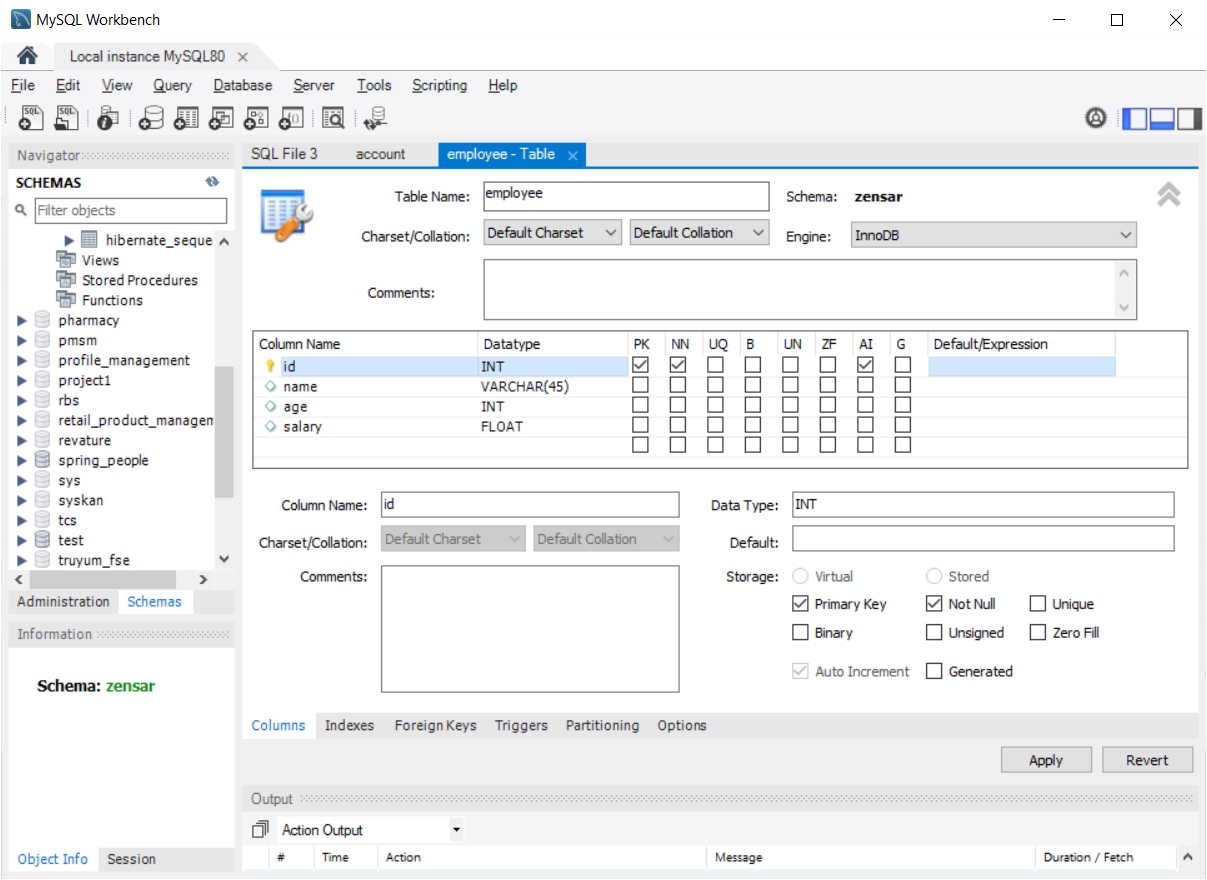
UN – UnSigned

AI – Auto Incremental

B – Binary (0/1 or true or false)

G – Generated Column (Calculated Field)

The convention for table & column names – All small case (multiple words are separated by underscore— Space is not allowed)



CREATE TABLE `zensar`.`employee` (

`id` INT NOT NULL AUTO\_INCREMENT,

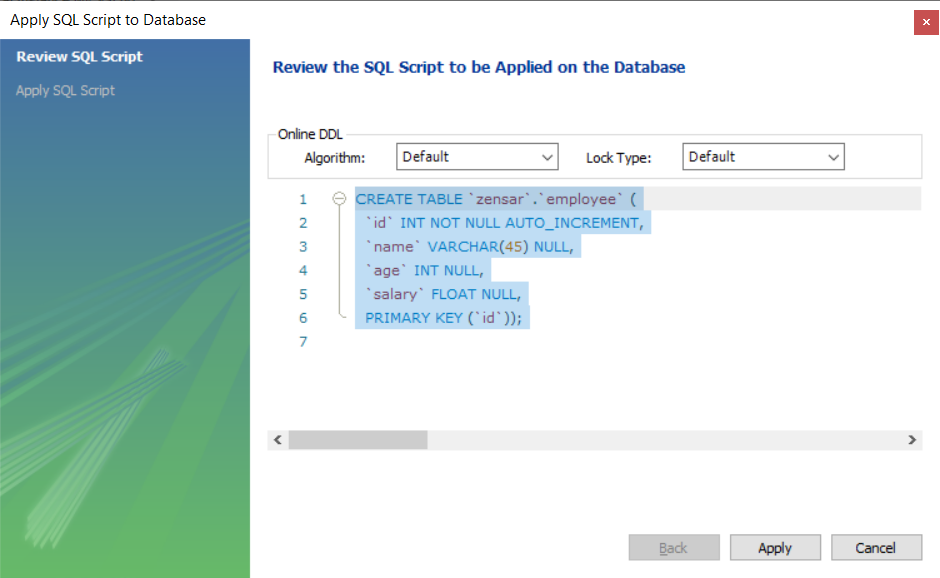
`name` VARCHAR(45) NULL,

`age` INT NULL,

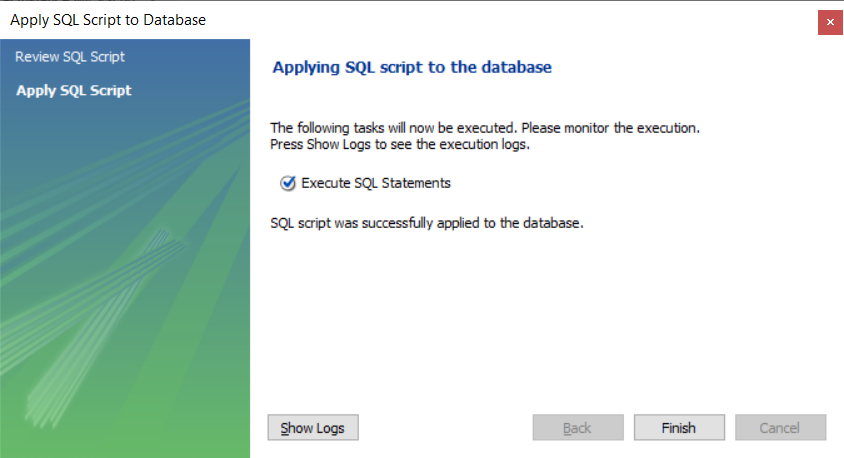
`salary` FLOAT NULL,

PRIMARY KEY (`id`));

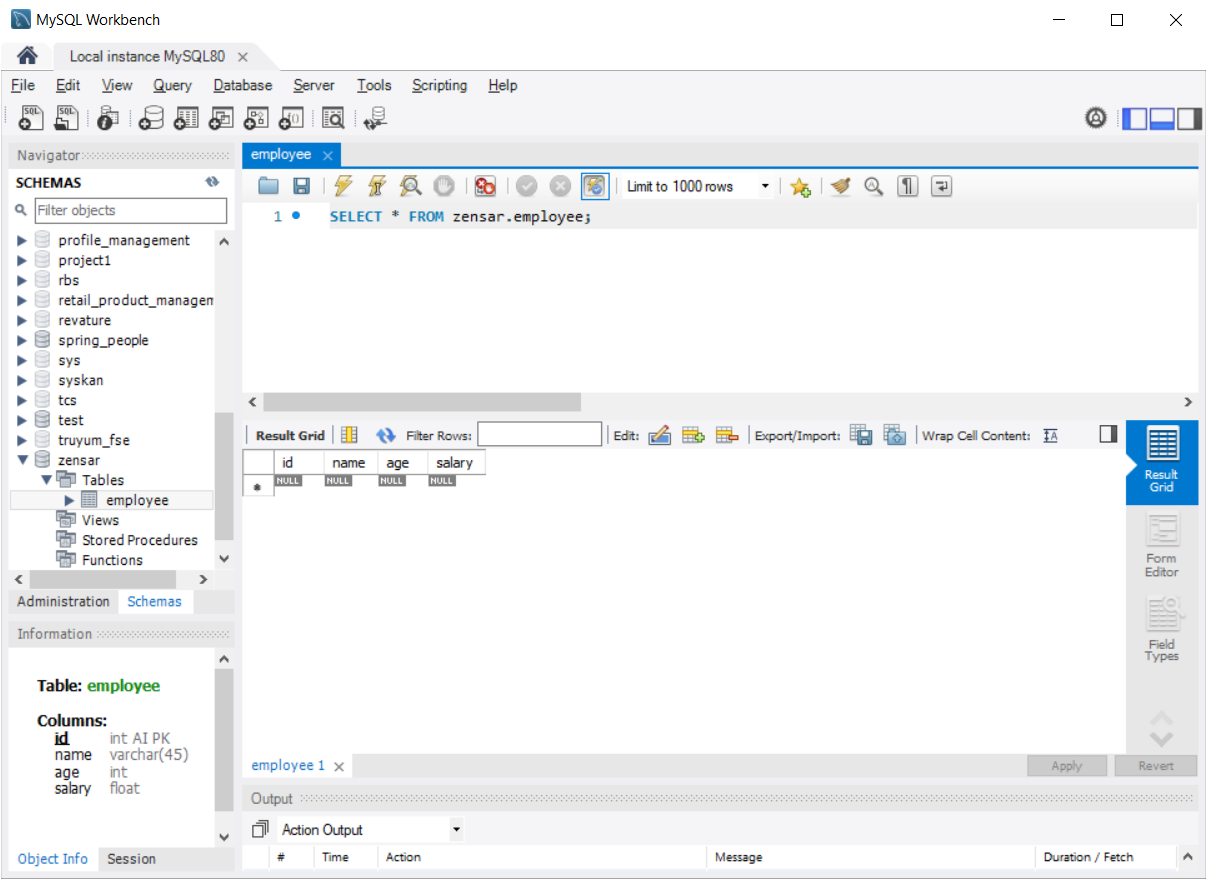
Step 16) Click on “Apply” button in the “Apply SQL Script to Database “ window as shown below



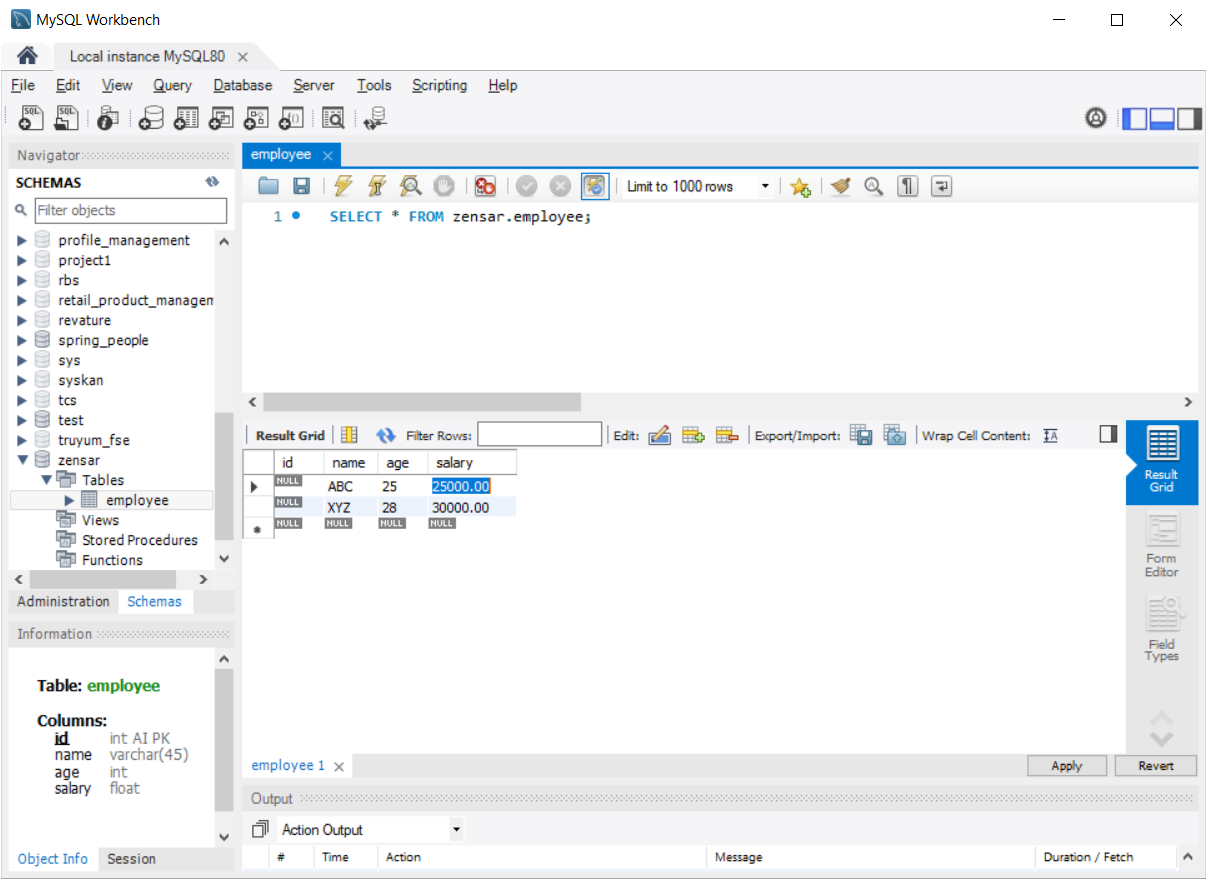
Step 17) You will get the result of Query Execution as shown below and click “Finish” button



Step 18) Right click on the newly created table and choose “Select Rows – Limit 1000” option, this will display empty table as shown below.

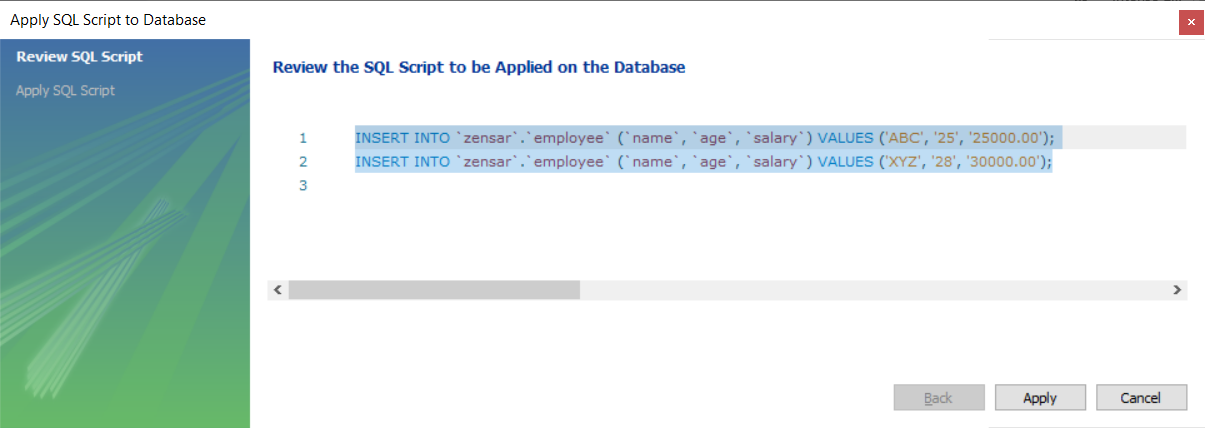


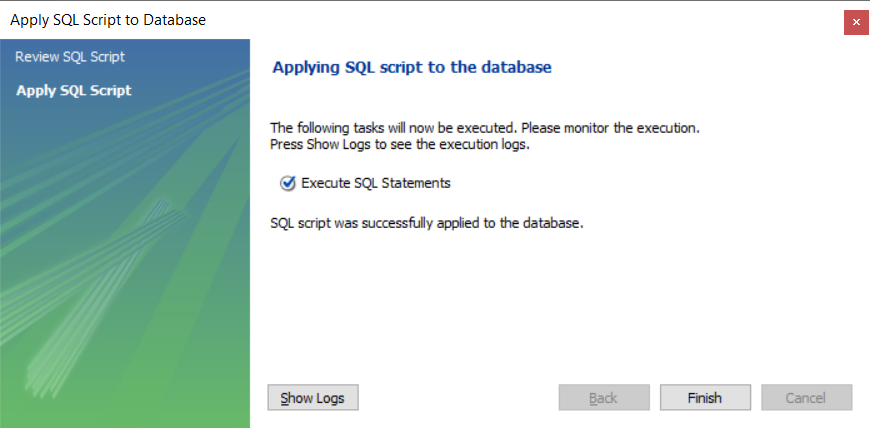
Step 19) Double click on any column to type data into it, as shown below & Click “Apply” button

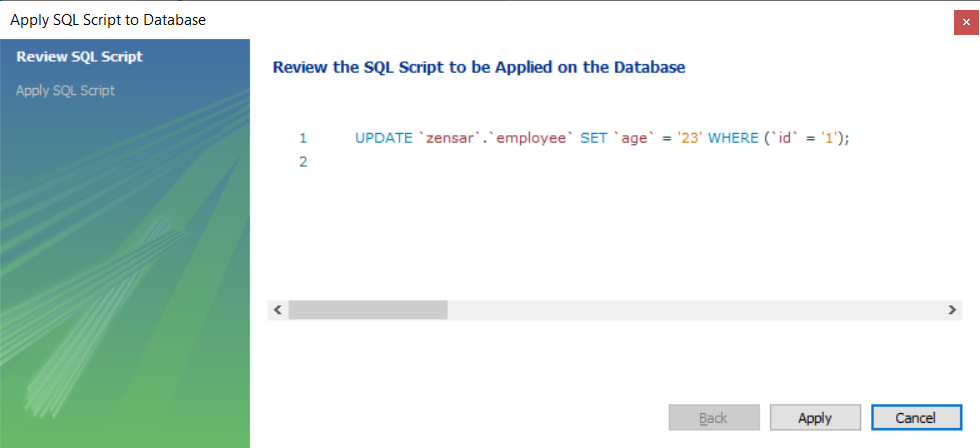


INSERT INTO `zensar`.`employee` (`name`, `age`, `salary`) VALUES ('ABC', '25', '25000.00');

INSERT INTO `zensar`.`employee` (`name`, `age`, `salary`) VALUES ('XYZ', '28', '30000.00');







Create a table called “student” in zensar schema, with the following column names

Roll\_no (pk), first\_name, last\_name, email, mobile(10 digit number)

Foreign Key – Restricting the insert operation in a table if there is no entry in it’s parent table

CREATE TABLE `zensar`.`department` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NULL,

PRIMARY KEY (`id`));

Partial Insert Queries

INSERT INTO `zensar`.`department` (`name`) VALUES ('HR');

INSERT INTO `zensar`.`department` (`name`) VALUES ('UI');

INSERT INTO `zensar`.`department` (`name`) VALUES ('BackEnd');

INSERT INTO `zensar`.`department` (`name`) VALUES ('Sales');

INSERT INTO `zensar`.`department` (`name`) VALUES ('Management');

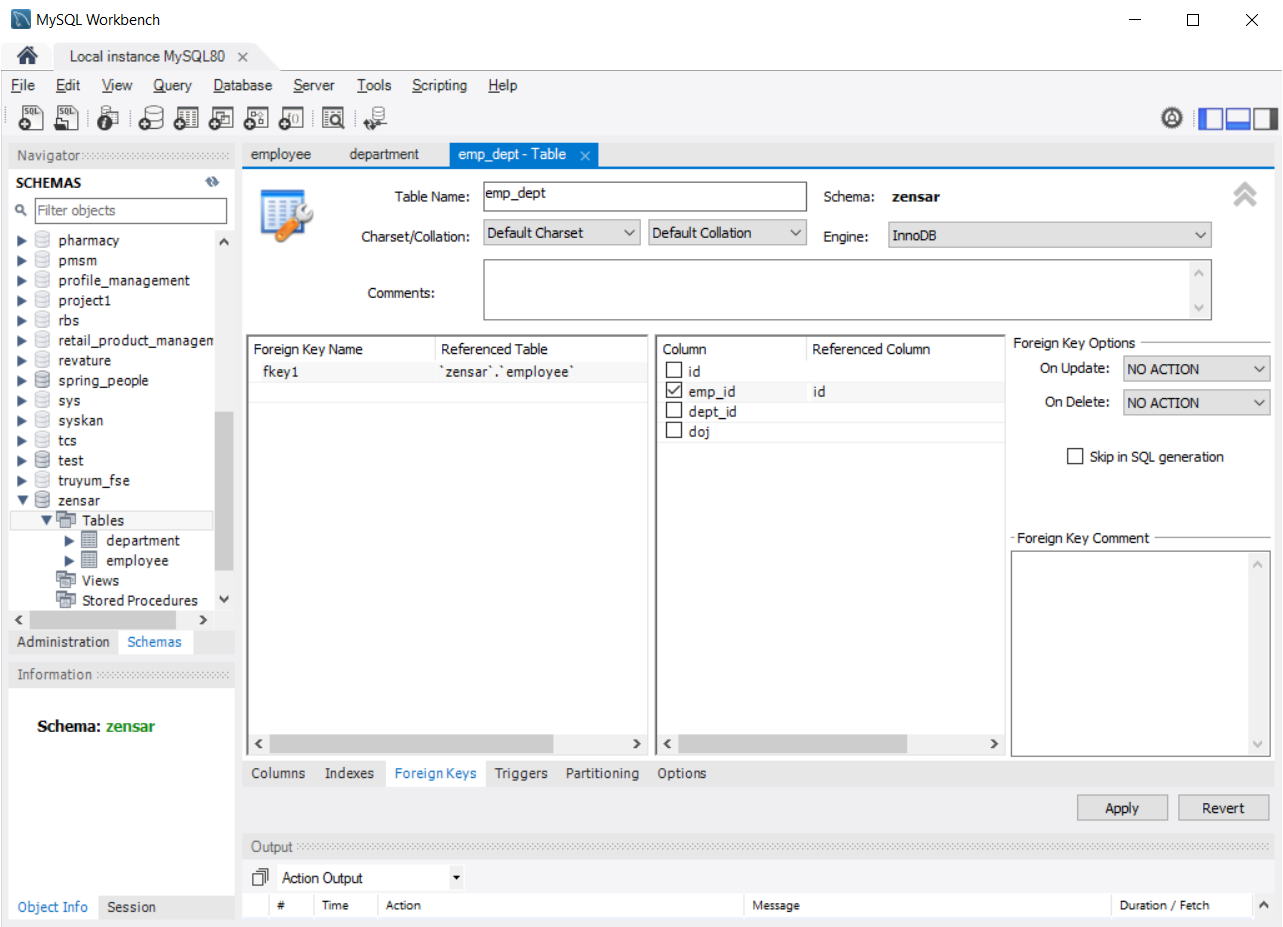
Applying a Foreign Key Constraint

Step 1) Create a table called “emp\_dept” with the following columns

(id, emp\_id, dept\_id, doj [date])



Step 2) Enter the foreign key name “fkey1” and select the reference table & column name of the reference table against the column name of the current table(emp\_dept)





CREATE TABLE `zensar`.`emp\_dept` (

`id` INT NOT NULL AUTO\_INCREMENT,

`emp\_id` INT NOT NULL,

`dept\_id` INT NOT NULL,

`doj` DATE NULL,

PRIMARY KEY (`id`),

INDEX `fkey1\_idx` (`emp\_id` ASC) VISIBLE,

INDEX `fkey2\_idx` (`dept\_id` ASC) VISIBLE,

CONSTRAINT `fkey1`

FOREIGN KEY (`emp\_id`)

REFERENCES `zensar`.`employee` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

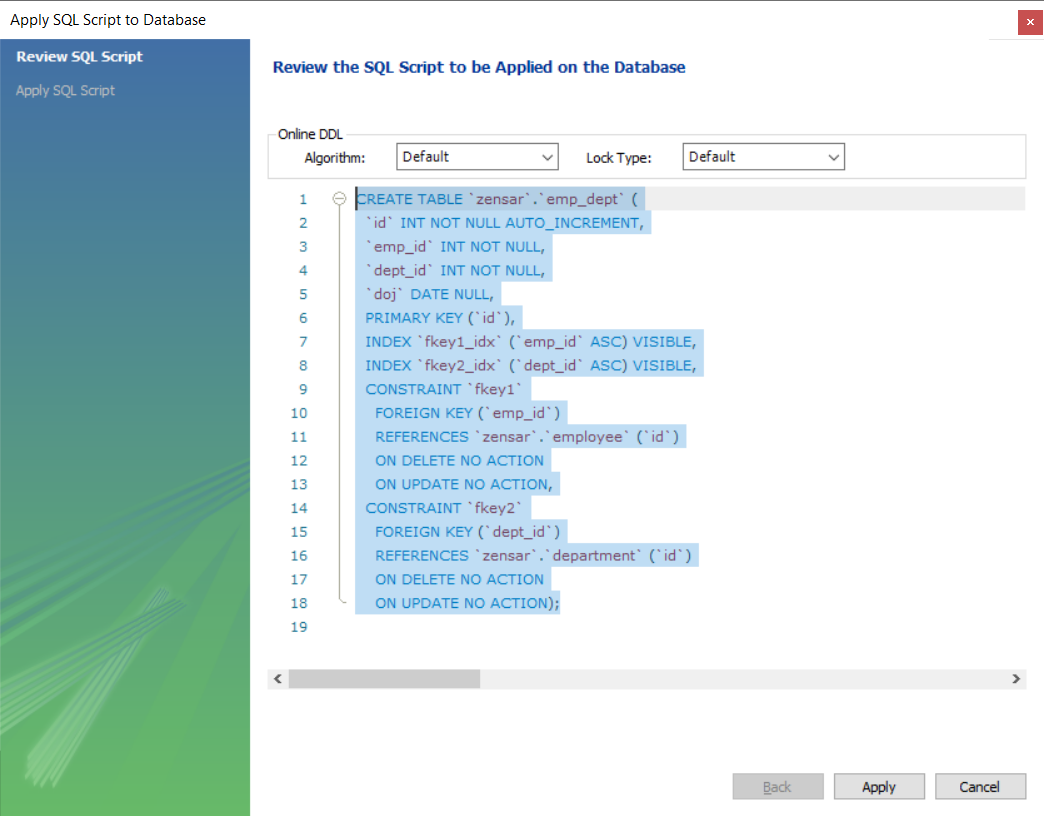
CONSTRAINT `fkey2`

FOREIGN KEY (`dept\_id`)

REFERENCES `zensar`.`department` (`id`)

ON DELETE NO ACTION

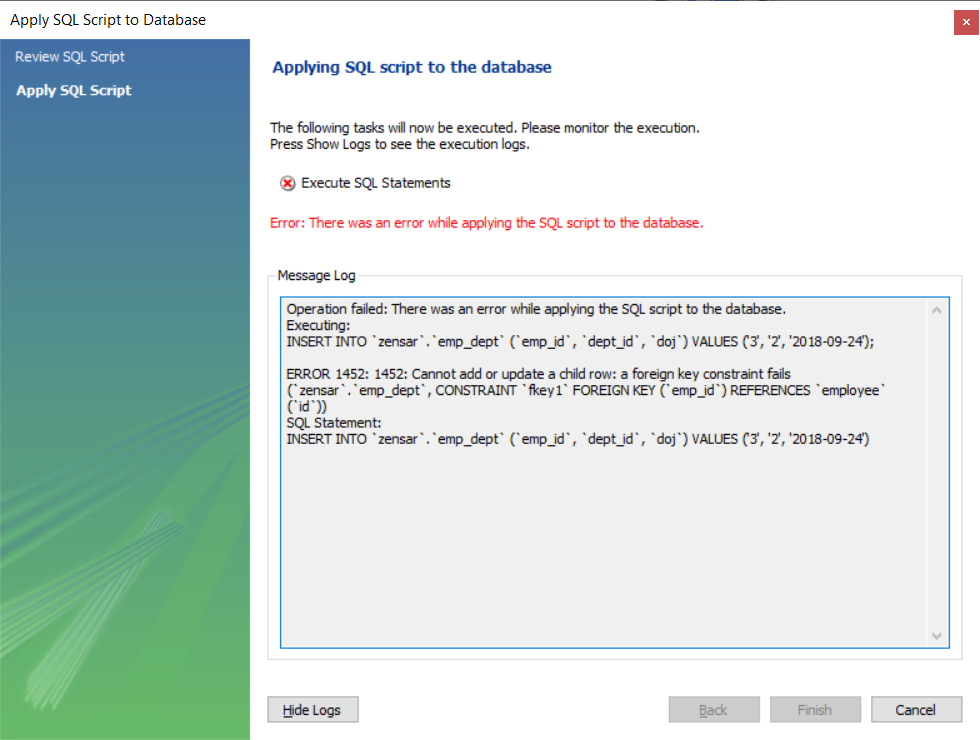
ON UPDATE NO ACTION);



INSERT INTO `zensar`.`emp\_dept` (`emp\_id`, `dept\_id`, `doj`) VALUES ('1', '1', '2020-03-10');

INSERT INTO `zensar`.`emp\_dept` (`emp\_id`, `dept\_id`) VALUES ('2', '4');

INSERT INTO `zensar`.`emp\_dept` (`emp\_id`, `dept\_id`, `doj`) VALUES ('3', '2', '2018-09-24'); -- Will get foreign key vilolated error because “emp\_id “ 3 is not available in employee table.



product table & order table (Zensar)

product\_id(pkey), name, unit\_price, category

order\_id(pkey), product\_id (fkey), qty, price (Calculated Field)

ER Diagram (Entity Relationship Diagram)

In Database, types of Relations

1. One to One Relationship (Table with a primary key – no duplicate, not null) 1..1
2. One to Many Relationship ( Foreign Key – It can have duplicates ) 1..n
3. Many to One Relationship (Foreign Key - ) n..1
4. Many to Many Relationship m..n

CREATE TABLE IF NOT EXISTS `mydb`.`order` (

`id` INT NOT NULL AUTO\_INCREMENT,

`qty` INT NULL,

`product\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_order\_product1\_idx` (`product\_id` ASC) VISIBLE,

CONSTRAINT `fk\_order\_product1`

FOREIGN KEY (`product\_id`)

REFERENCES `mydb`.`product` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

CREATE TABLE IF NOT EXISTS `mydb`.`product` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NULL,

`category` VARCHAR(45) NULL,

`unit\_price` FLOAT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB

-- MySQL Script generated by MySQL Workbench

-- Mon Jan 3 07:08:54 2022

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

DROP SCHEMA IF EXISTS `mydb` ;

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;

USE `mydb` ;

-- -----------------------------------------------------

-- Table `mydb`.`PAN`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`PAN` ;

CREATE TABLE IF NOT EXISTS `mydb`.`PAN` (

`PAN` VARCHAR(15) NOT NULL,

`emp\_id` INT NULL,

PRIMARY KEY (`PAN`),

INDEX `fky1\_idx` (`emp\_id` ASC) VISIBLE,

UNIQUE INDEX `emp\_id\_UNIQUE` (`emp\_id` ASC) VISIBLE,

CONSTRAINT `fky1`

FOREIGN KEY (`emp\_id`)

REFERENCES `mydb`.`employee` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`department`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`department` ;

CREATE TABLE IF NOT EXISTS `mydb`.`department` (

`id` INT NOT NULL,

`name` VARCHAR(45) NULL,

`establishment\_year` INT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`emp\_dept`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`emp\_dept` ;

CREATE TABLE IF NOT EXISTS `mydb`.`emp\_dept` (

`id` INT NOT NULL,

`emp\_id` INT NULL,

`dept\_id` INT NULL,

`doj` DATE NULL,

`emp\_deptcol` VARCHAR(45) NULL,

PRIMARY KEY (`id`),

INDEX `fk1\_idx` (`emp\_id` ASC) VISIBLE,

INDEX `fk2\_idx` (`dept\_id` ASC) VISIBLE,

CONSTRAINT `fk1`

FOREIGN KEY (`emp\_id`)

REFERENCES `mydb`.`employee` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk2`

FOREIGN KEY (`dept\_id`)

REFERENCES `mydb`.`department` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`employee`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`employee` ;

CREATE TABLE IF NOT EXISTS `mydb`.`employee` (

`id` INT NOT NULL,

`name` VARCHAR(45) NULL,

`age` INT NULL,

`salary` FLOAT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`order`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`order` ;

CREATE TABLE IF NOT EXISTS `mydb`.`order` (

`id` INT NOT NULL AUTO\_INCREMENT,

`qty` INT NULL,

`product\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_order\_product1\_idx` (`product\_id` ASC) VISIBLE,

CONSTRAINT `fk\_order\_product1`

FOREIGN KEY (`product\_id`)

REFERENCES `mydb`.`product` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`product`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `mydb`.`product` ;

CREATE TABLE IF NOT EXISTS `mydb`.`product` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NULL,

`category` VARCHAR(45) NULL,

`unit\_price` FLOAT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

Database Normalization (Process of removing anomalies from DB entity)

1. 1NF (First Normal Form) No Multiple values in a single cell - adding primary key
2. 2NF (divide the table into two, so that duplicate cells can be avoided) – adding foreign key
3. 3NF (No functional dependencies) -
4. BCNF (Boyce- Codd Normal Form)
5. 4NF
6. 5NF
7. 6NF

SQL functions

1. Row Level Functions
2. Column Level Functions - Max(), Min(), count(), Avg(), Sum(), cos(), sin(), log()

toUpperCase(), changing the date format()

In SQL, we have the following number constraints

Constraints are rules

1. Unique – No duplicates (Unique constraint violated) - UNIQUE
2. Not Null – NOT NULL
3. PK – Primary Key (Unique & Not Null) – PRIMARY KEY
4. B – binary
5. UN – Unsigned
6. AI – Auto Incremented (A sequence will be created and the seq.nextVal) – AUTO INCREMENT
7. G – Generated Value

CRUD – Create Read Update Delete

1. Create <entity\_name> (<column\_name> <column\_type> <constraint> ) -- insert
2. Select \* or (<names\_of\_column) from <table\_name>
3. Update table <table\_name> set <column\_name>=<updated\_value> where <pk\_column>=<value>
4. Delete from <table\_name> where <pk\_column>=<value>

DDL – Data Definition Lang (create, drop, alter, truncate)

DML – Data Manipulation Lang (insert, update, delete)

TCL – Transaction Control Lang (commit, rollback, savepoint)

DCL – Data Control Lang (Grant & Revoke)

Name Address Movies Rented (Multi Value) Charge(Multi-value)