

LAB MANUAL
of
Database Management Systems
Laboratory
(CS238)
Bachelor of Technology (CSE)
By
Ramoliya Kaushal (22000409)
Second Year, Semester 4
Course In-charge: Prof. Ujwala Petigara



NAV RACHANA
UNIVERSITY
a UGC recognized University

Department of Computer Science and Engineering
School Engineering and Technology
Navrachana University, Vadodara
Spring Semester
(2024)

INDEX

Lab No.	Lab Exercise	Date	Page No.
1	Introduction to the MySQL database ecosystem.	16-01-2024	3
2	Study of SQL Commands	23-01-2024	14
3	To study DDL-create and DML-insert commands	30-01-2024	17
4	To perform various data manipulation commands & aggregate functions on all created tables.	08-02-2024	31
5	W3School SQL Queries	15-02-2024	42
6	W3School Tables Creation	22-02-2024	52
7	W3School SQL Practice.	29-02-2024	68
8	SQL Practice	14-03-2024	82
9	Use the concept of PL/SQL Control Structure	21-03-2024	98
10	Write a program to Print Number using PL/SQL Loop.	28-03-2024	106
11	To study TCL Commands.	04-04-2024	114
12	Create Connection Between Programming Language and Database.	18-04-2024	120

LAB-1

Aim: introduction to the MySQL database ecosystem.

- (i) What is MySQL Server.
- (ii) What is MySQL Client.
- (iii) What is MySQL Workbench.

(i) What is MySQL Server.

- The MySQL server provides a database management system with querying and connectivity capabilities, as well as the ability to have excellent data structure and integration with many different platforms.
- It can handle large databases reliably and quickly in high-demanding production environments.
- The MySQL server also provides rich function such as its connectivity, speed, and security that make it suitable for accessing databases.
- The MySQL server works in a client and server system.

(ii) What is MySQL Client.

- A MySQL client is a program that allows users to communicate with a MySQL server to manipulate the information in the databases that the server manages.
- The MySQL client is a common name for tools that are designed to connect to MySQL Server.
- The client programs are used to send commands or queries to the server and allow managing data in the databases.
- The MySQL client program is one of the clients included in MySQL distributions.

(iii) What is MySQL Workbench.

- MySQL Workbench is a unified visual tool for database architects, developers, and DBAs.
- It provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and more.
- MySQL Workbench is available on Windows, Linux, and Mac OS X. It enables users to visually design, model, generate, and manage databases, as well as create, execute, and optimize SQL queries.
- Additionally, it provides a visual console for administering MySQL environments and offers a suite of tools to improve the performance of MySQL servers.

MySQL Workbench Installation: A Step-by-Step

1. Open the MySQL website on a browser. Click on the following link: [MySQL Downloads](#).
2. Select the Downloads option.

The screenshot shows the MySQL website homepage. At the top, there's a navigation bar with links for 'MySQL.COM', 'DOWNLOADS' (which is underlined in orange), 'DOCUMENTATION', and 'DEVELOPER ZONE'. On the far right, there are social media icons for Facebook, Twitter, LinkedIn, and YouTube, along with links for 'Contact MySQL', 'Login', and 'Register'. Below the navigation, there's a large blue banner for 'MySQL Database Service with HeatWave for Real-time Analytics'. The banner features two sections: 'Faster Performance' (listing 400x MySQL query acceleration, 1100x faster than Amazon Aurora, and 2.7x faster than Amazon Redshift) and 'Lower Total Cost of Ownership' (listing 1/3 the cost of Amazon RDS, 1/3 the cost of Amazon Redshift, and easy migration from Amazon RDS). At the bottom left of the banner is a 'Try Now' button.

3. Select MySQL Installer for Windows.

⬇ MySQL Community Downloads

- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository
- MySQL Community Server
- MySQL Cluster
- MySQL Router
- MySQL Shell
- MySQL Workbench
- MySQL Installer for Windows
- MySQL for Visual Studio
- C API (libmysqlclient)
- Connector/C++
- Connector/J
- Connector/.NET
- Connector/Node.js
- Connector/ODBC
- Connector/Python
- MySQL Native Driver for PHP
- MySQL Benchmark Tool
- Time zone description tables
- Download Archives



© 2021, Oracle Corporation and/or its affiliates

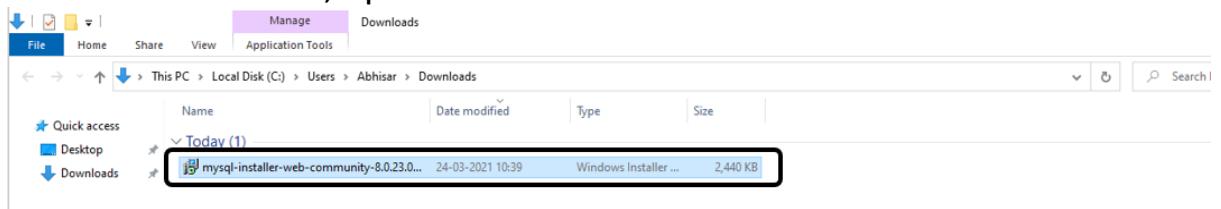
4. Choose the desired installer and click on download.

The screenshot shows the MySQL Installer 8.0.23 download page. At the top, there are tabs for "General Availability (GA) Releases" (highlighted in orange), "Archives", and a help icon. Below the tabs, the title "MySQL Installer 8.0.23" is displayed. A dropdown menu "Select Operating System:" shows "Microsoft Windows". To the right, a link says "Looking for previous GA versions?". Two download options are listed:

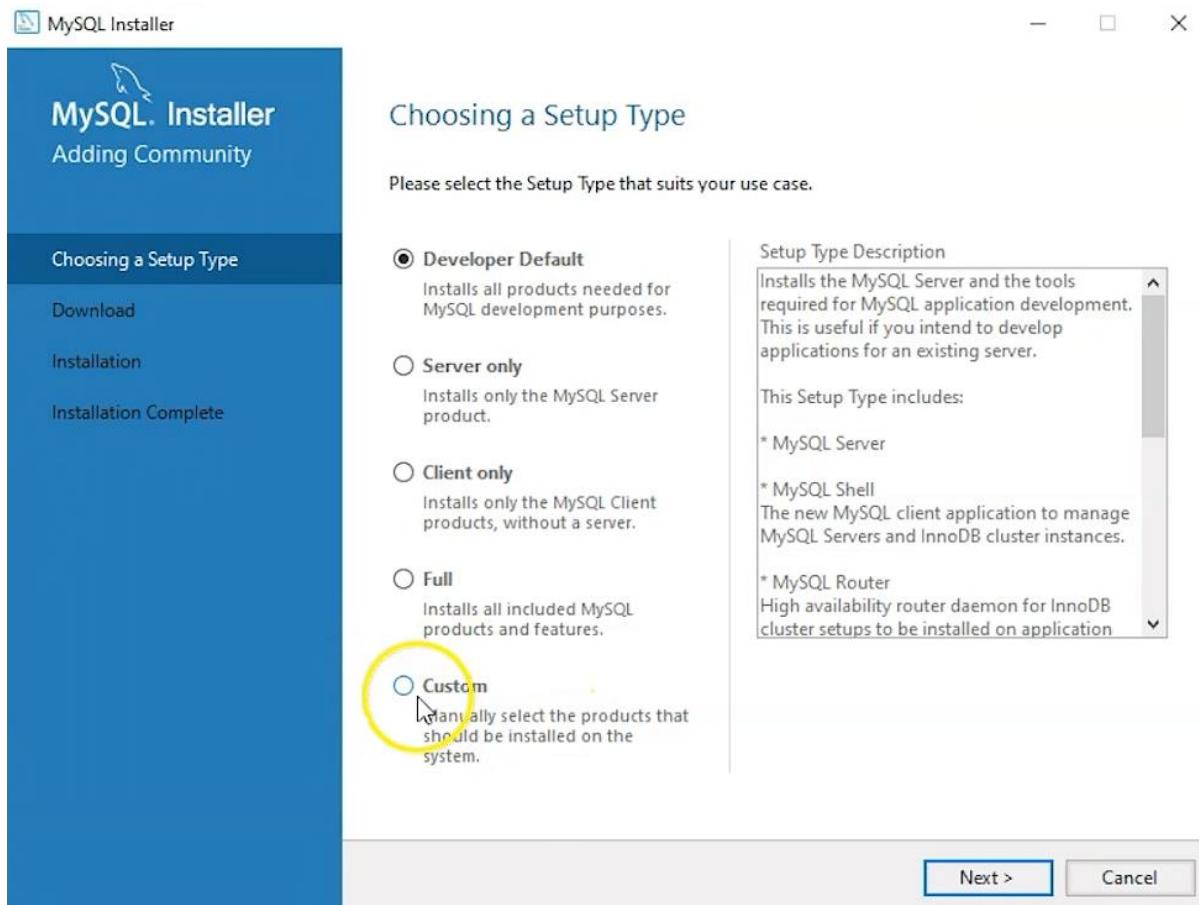
Installer Type	Version	File Size	Action
Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.23.0.msi)	8.0.23	2.4M	Download
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.23.0.msi)	8.0.23	422.4M	Download

A note at the bottom left says: "We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download."

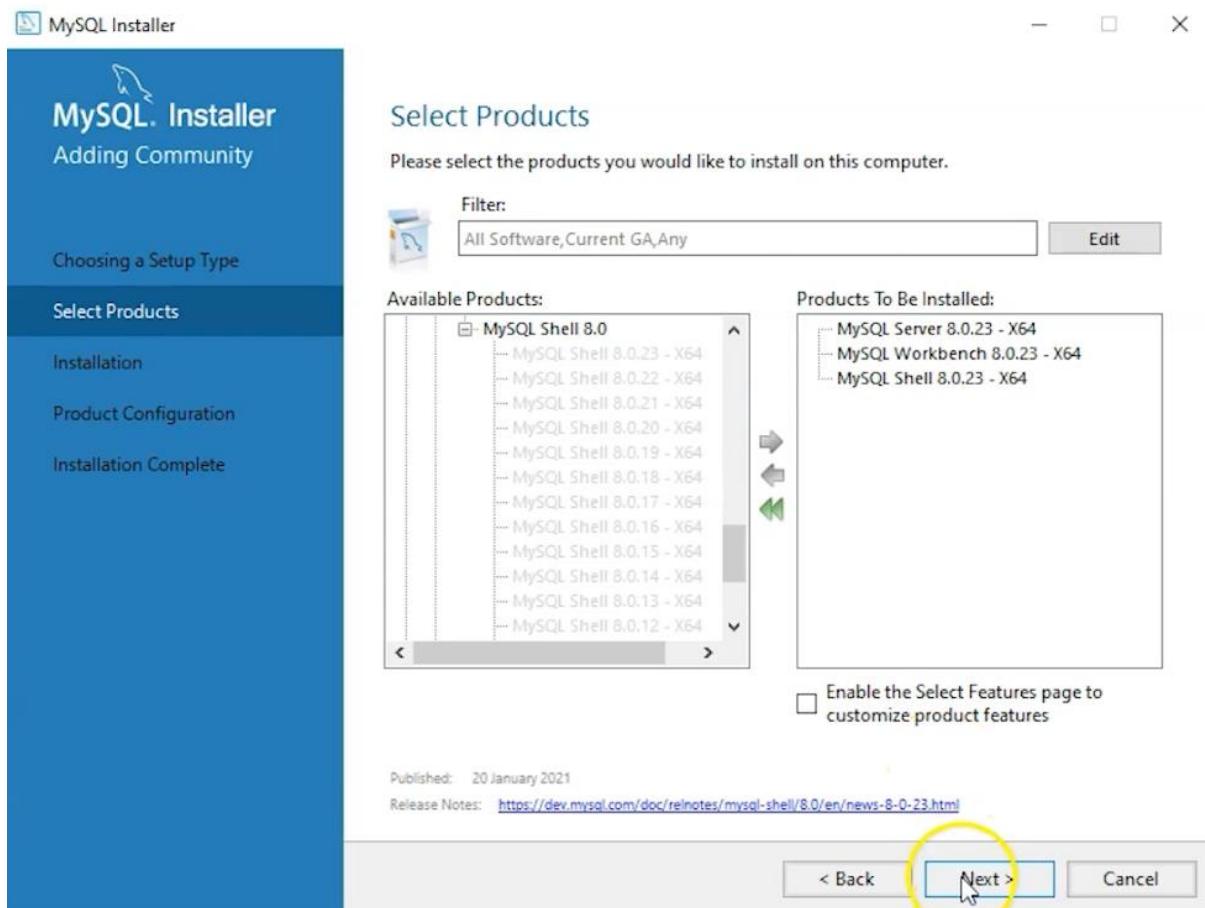
5. After the download, open the installer.



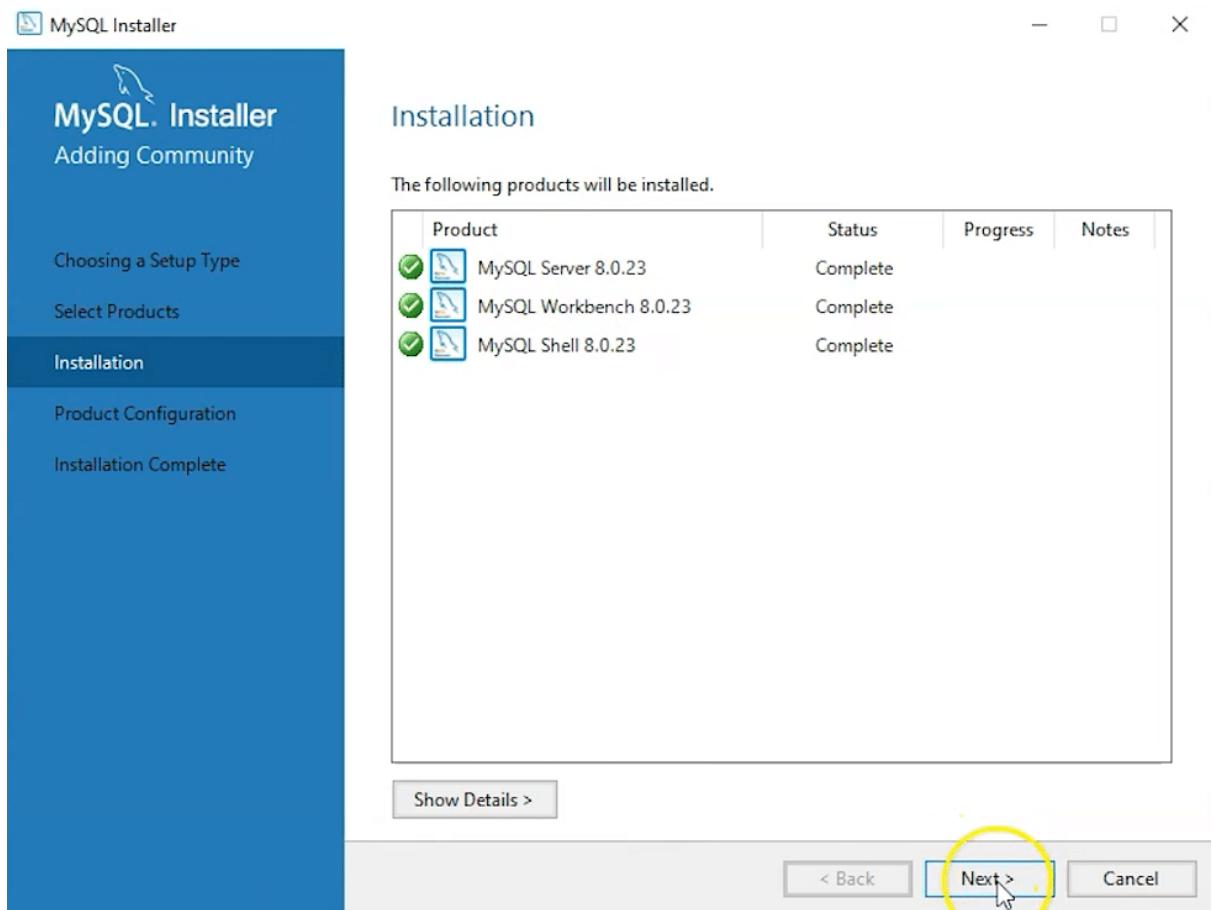
6. It will ask for permission; when it does, click Yes. The installer will then open. Now, it will ask to choose the setup type. Here, select Custom.



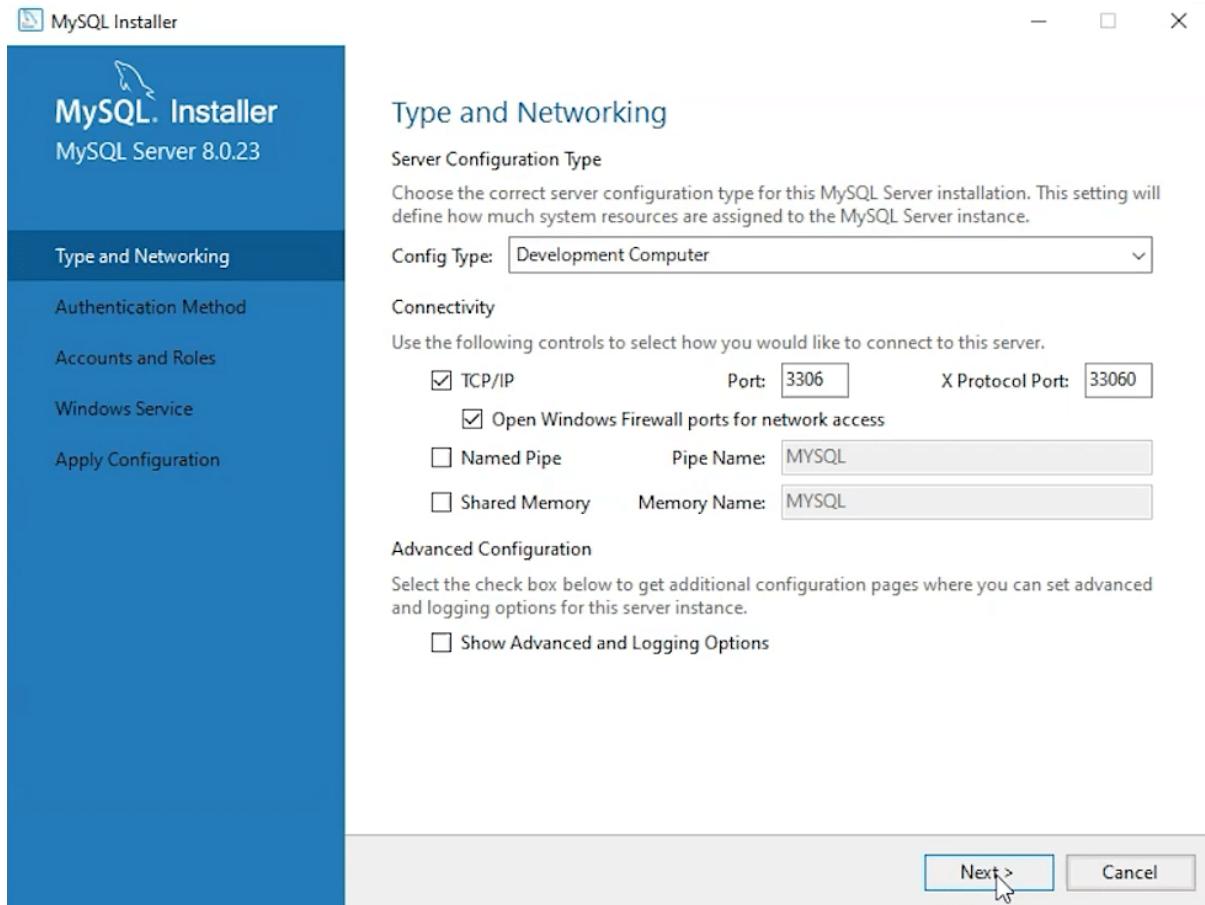
7. Click on Next. With this, you will install MySQL server, MySQL Workbench, and MySQL shell.
8. Open MySQL Servers, select the server you want to install, and move it to the Products/Features to be installed window section. Now, expand Applications, choose MySQL Workbench and MySQL shell. Move both of them to ‘Products/Features to be installed’.



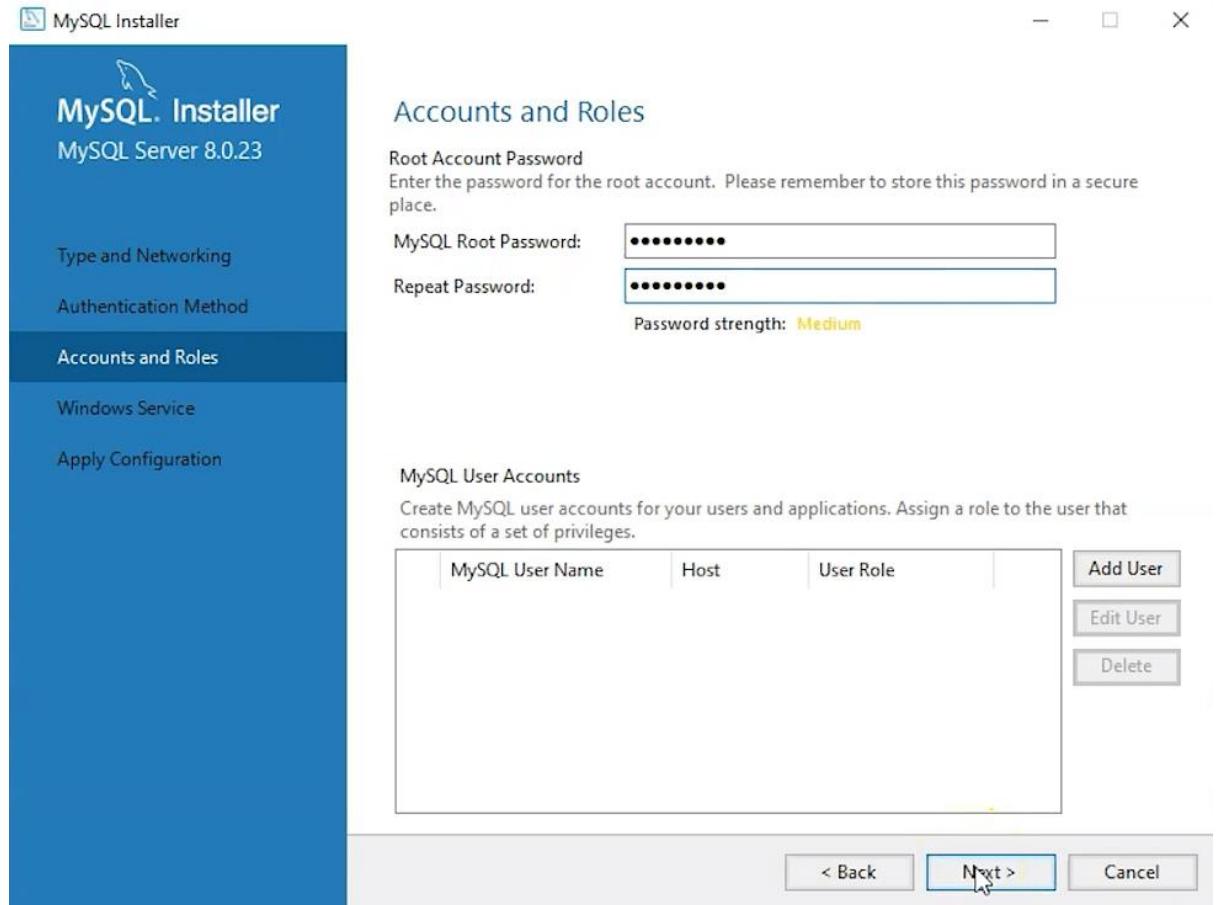
9. Click on the Next button. Now, click on the Execute button to download and install the MySQL server, MySQL Workbench, and the MySQL shell.



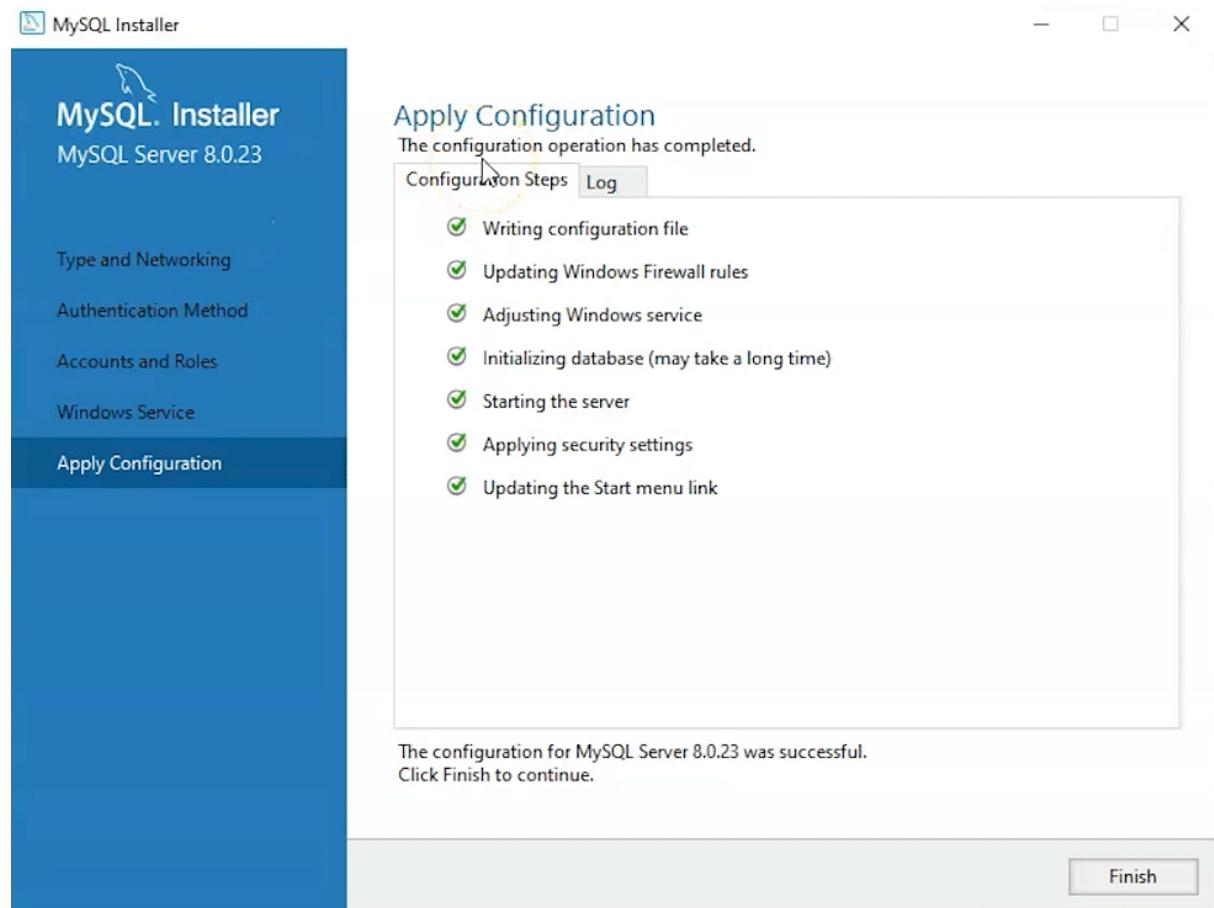
10. Once the product is ready to configure, click on Next. Under Type and Networking, go with the default settings and select Next.



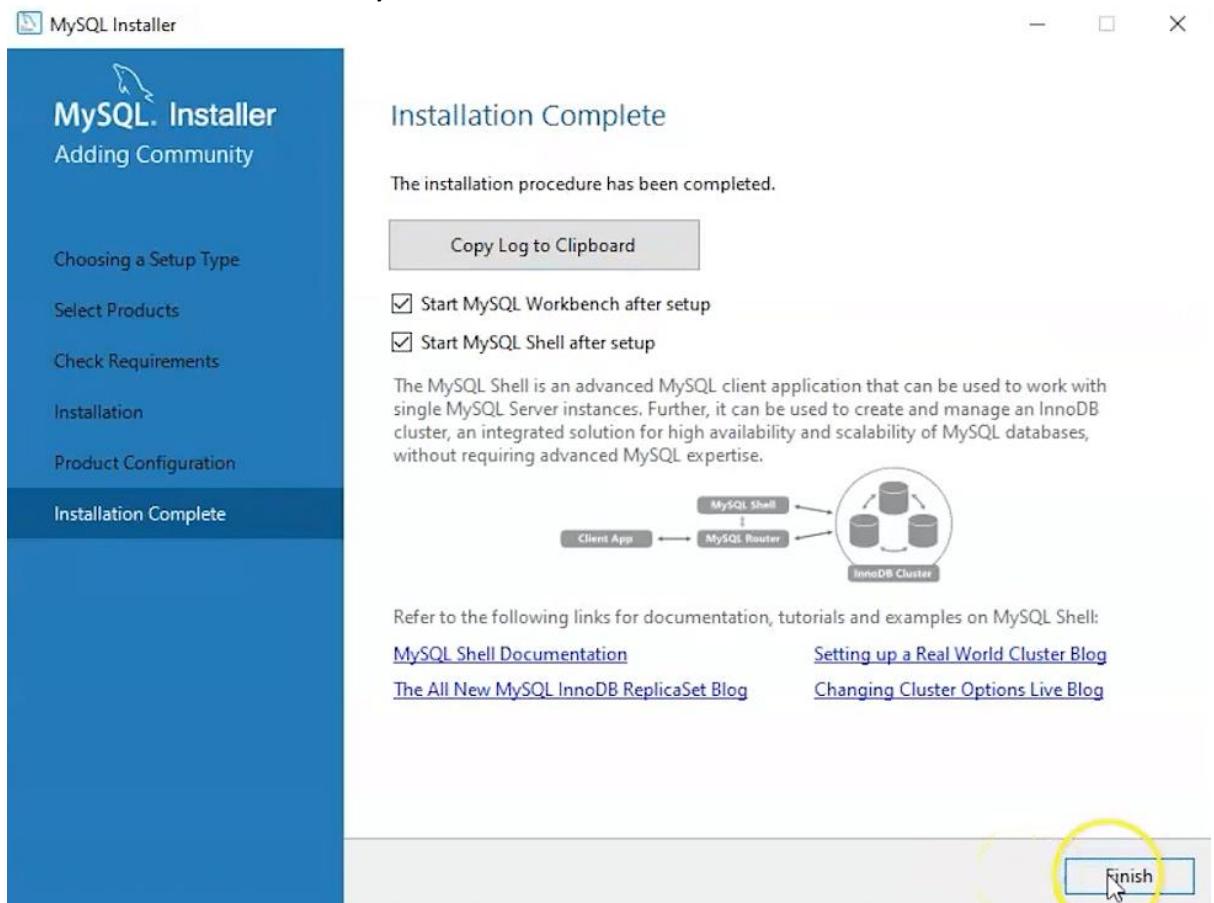
11. For authentication, use the recommended strong password encryption
12. Set your MySQL Root password and click on next.



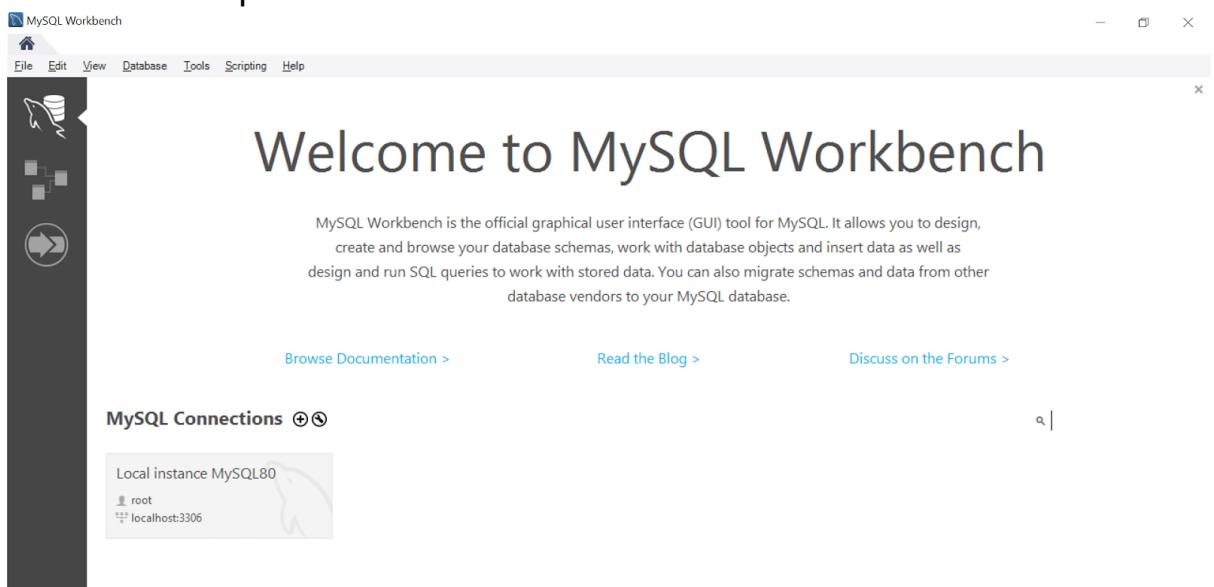
13. Go for the default windows service settings and under apply configuration, click on execute. Once the configuration is complete, click on finish.



- 14.** Complete the installation. This will now launch the MySQL Workbench and the MySQL Shell.



- 15.** Once MySQL Workbench is installed, select the Local instance and enter the password.



LAB-2

Aim: Study of SQL Commands.

Step-1:

Open Command Prompt.

Navigate to the bin folder. For example:

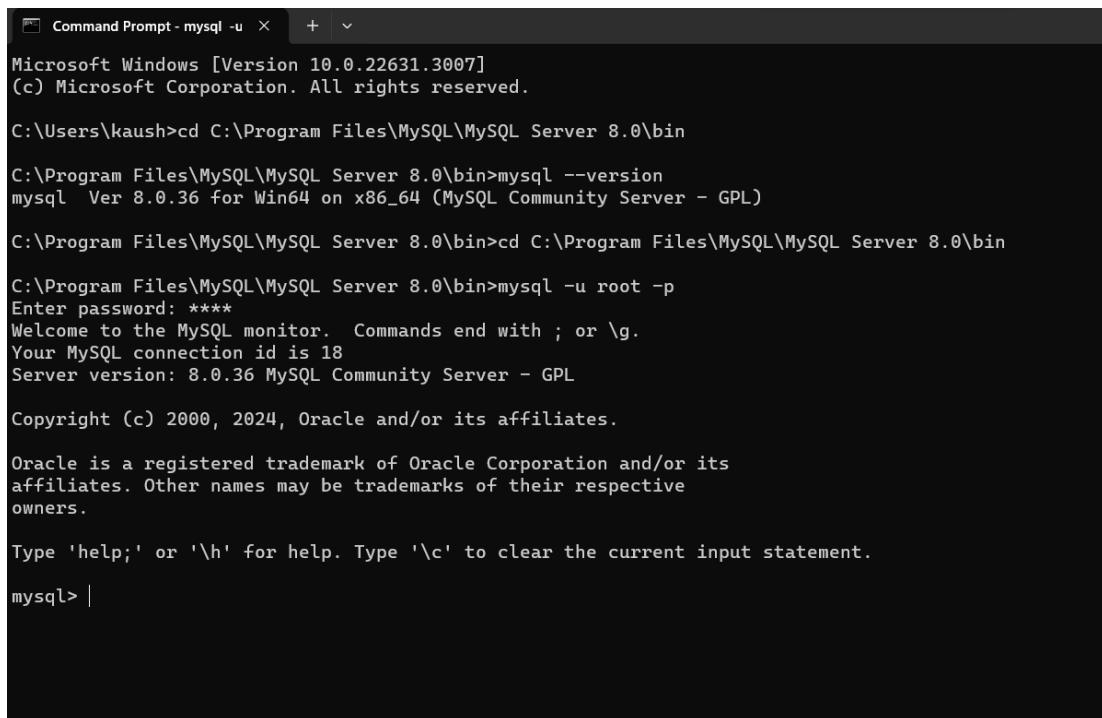
```
cd "C:\Program Files\MySQL\MySQL Server 8.0\bin"
```

Step-2:

Run the mysql -u root -p command.

Step-3:

Enter the password.



The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u". The window displays the following MySQL session:

```
Microsoft Windows [Version 10.0.22631.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaush>cd C:\Program Files\MySQL\MySQL Server 8.0\bin

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql --version
mysql Ver 8.0.36 for Win64 on x86_64 (MySQL Community Server - GPL)

C:\Program Files\MySQL\MySQL Server 8.0\bin>cd C:\Program Files\MySQL\MySQL Server 8.0\bin

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> |
```

1. Create Database:

```
mysql> CREATE DATABASE firstlabdbms;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> |
```

2.use Database:

```
mysql> USE firstlabdbms;
Database changed
mysql> |
```

3.Create table:

```
mysql> create table student(
    -> id int not null, name varchar(30) not null, semester int not null);
Query OK, 0 rows affected (0.03 sec)
```

4.Create table:

```
mysql> create table Student(ID int, Name varchar(255), Marks int);
Query OK, 0 rows affected (0.05 sec)
```

5.Insert Data: insert into tablename value.

```
mysql> insert into Student value (22000409, 'kaushal', 92), (22000404, 'harsh', 98), (22000407, 'Poojit', 89);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

6. Show table : select from * tablename.

```
mysql> select * from Student;
+----+----+----+
| ID | Name | Marks |
+----+----+----+
| 22000409 | kaushal |    92 |
| 22000404 | harsh |    98 |
| 22000407 | Poojit |    89 |
+----+----+----+
3 rows in set (0.00 sec)
```



LAB-3

Aim: To study DDL-create and DML-insert commands.

A. Create tables according to the following definition.

- CREATE TABLE DEPOSIT (ACTNO VARCHAR2(5), CNAME VARCHAR2(18) , BNAME VARCHAR2(18), AMOUNT NUMBER (8,2), ADATE DATE);

```
mysql> create table deposit (ACTION varchar(10),CNAME varchar(18),BNAME varc  
har(18),AMOUNT dec(8,2), ADATE date);  
Query OK, 0 rows affected (0.12 sec)
```

- CREATE TABLE BRANCH (BNAME VARCHAR2(18), CITY VARCHAR2(18));

```
mysql> create table branch (BNAME varchar(18),CITY varchar(18));  
Query OK, 0 rows affected (0.03 sec)
```

- CREATE TABLE CUSTOMERS (CNAME VARCHAR2 (19), CITY VARCHAR2(18));

```
mysql> create table customers (CNAME varchar(18),CITY varchar(18));  
Query OK, 0 rows affected (0.03 sec)
```

- CREATE TABLE BORROW (LOANNO VARCHAR2(5), CNAME VARCHAR2(18), BNAME VARCHAR2(18), AMOUNT NUMBER (8,2));

```
mysql> create table borrow (LOANNO varchar(5),CNAME varchar(18),BNAME varcha  
r(18), AMOUNT dec(8,2));  
Query OK, 0 rows affected (0.07 sec)
```

B. Insert the data as shown below:

DEPOSIT: ACTNO	CNAME	BNAME	AMOUNT	ADATE
100	ANIL	VRCE	1000.00	1-MAR-95
101	SUNIL	AJNI	5000.00	4-JAN-96
102	MEHUL	KAROLBAGH	3500.00	17-NOV-95
104	MADHURI	CHANDI	1200.00	17-DEC-95
105	PRAMOD	M.G.ROAD	3000.00	27-MAR-96
106	SANDIP	ANDHERI	2000.00	31-MAR-96
107	SHIVANI	VIRAR	1000.00	5-SEP-95
108	KRANTI	NEHRU PLACE	5000.00	2-JUL-95
109	MINU	POWAI	7000.00	10-AUG-95

Code:

```
mysql> insert into deposit value ('100','ANIL','VRCE',1000.00,'1995-03-01'),
('101','SUNIL','AJNI',5000.00,'1996-01-04'),('102','MEHUL','KAROLBAGH',3500.
00,'1995-11-17'),('104','MADHURI','CHANDI',1200.00,'1995-12-17'),('105','PRA
MOD','M.G.ROAD',3000.00,'1996-03-31'),('106','SANDIP','ANDHERI',2000.00,'199
6-03-31'),('107','SHIVANI','VIRAR',1000.00,'1995-09-05'),('108','KRANTI','NE
HRU PLACE',5000.00,'1995-07-02'),('109','MINU','POWAI',7000.00,'1995-08-10')
;
Query OK, 9 rows affected (0.04 sec)
Records: 9  Duplicates: 0  Warnings: 0
```

Output:

```
mysql> select * from deposit;
+-----+-----+-----+-----+-----+
| ACTION | CNAME | BNAME | AMOUNT | ADATE |
+-----+-----+-----+-----+-----+
| 100 | ANIL | VRCE | 1000.00 | 1995-03-01 |
| 101 | SUNIL | AJNI | 5000.00 | 1996-01-04 |
| 102 | MEHUL | KAROLBAGH | 3500.00 | 1995-11-17 |
| 104 | MADHURI | CHANDI | 1200.00 | 1995-12-17 |
| 105 | PRAMOD | M.G.ROAD | 3000.00 | 1996-03-31 |
| 106 | SANDIP | ANDHERI | 2000.00 | 1996-03-31 |
| 107 | SHIVANI | VIRAR | 1000.00 | 1995-09-05 |
| 108 | KRANTI | NEHRU PLACE | 5000.00 | 1995-07-02 |
| 109 | MINU | POWAI | 7000.00 | 1995-08-10 |
+-----+-----+-----+-----+-----+
9 rows in set (0.01 sec)
```

BRANCH: BNAME	CITY
VRCE	NAGPUR
AJNI	NAGPUR
KAROLBAGH	DELHI
CHANDI	DELHI
DHARAMPETH	NAGPUR
M.G.ROAD	BANGLORE
ANDHERI	BOMBAY
VIRAR	BOMBAY
NEHRU PLACE	DELHI
POWAI	BOMBAY

Code:

```
mysql> insert into branch value
    -> ('VRCE','NAGPUR'),
    -> ('AJNI','NAGPUR'),
    -> ('KAROLBAGH','DELHI'),
    -> ('CHANDI','DELHI'),
    -> ('DHARAMPETH','NAGPUR'),
    -> ('M.G.ROAD','BANGLORE'),
    -> ('ANDHERI','BOMBAY'),
    -> ('VIRAR','BOMBAY'),
    -> ('NEHRU PLACE','DELHI'),
    -> ('POWAI','BOMBAY');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

Output:

```
mysql> select * from branch;
+-----+-----+
| BNAME      | CITY   |
+-----+-----+
| VRCE       | NAGPUR |
| AJNI       | NAGPUR |
| KAROLBAGH | DELHI  |
| CHANDI     | DELHI  |
| DHARAMPETH | NAGPUR |
| M.G.ROAD   | BANGLORE|
| ANDHERI    | BOMBAY |
| VIRAR      | BOMBAY |
| NEHRU PLACE | DELHI  |
| POWAI      | BOMBAY |
+-----+-----+
10 rows in set (0.01 sec)
```

CUSTOMERS:	CITY
CNAME	
ANIL	CALCUTTA

SUNIL	DELHI
MEHUL	BARODA
MANDAR	PATNA
MADHURI	NAGPUR
PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	BOMBAY
KRANTI	BOMBAY
NAREN	BOMBAY

Code:

```
mysql> insert into customers value
    -> ('ANIL','CALCUTTA'),
    -> ('SUNIL','DELHI'),
    -> ('MEHUL','BARODA'),
    -> ('MANDAR','PATNA'),
    -> ('MADHURI','NAGPUR'),
    -> ('SANDIP','SURAT'),
    -> ('SHIVANI','BOMBAY'),
    -> ('KRANTI','BOMBAY'),
    -> ('NAREN','BOMBAY');
Query OK, 9 rows affected (0.01 sec)
Records: 9  Duplicates: 0  Warnings: 0
```

Output:

```
mysql> select * from customers;
+-----+-----+
| CNAME | CITY  |
+-----+-----+
| ANIL  | CALCUTTA |
| SUNIL | DELHI   |
| MEHUL | BARODA  |
| MANDAR | PATNA  |
| MADHURI | NAGPUR |
| SANDIP | SURAT   |
| SHIVANI | BOMBAY |
| KRANTI | BOMBAY |
| NAREN  | BOMBAY |
+-----+-----+
9 rows in set (0.00 sec)
```

BORROW: LOANNO	CNAME	BNAME	AMOUNT
201	ANIL	VRCE	1000.00
206	MEHUL	AJNI	5000.00
311	SUNIL	DHARAMPETH	3000.00
321	MADHURI	ANDHERI	2000.00
375	PRAMOD	VIRAR	8000.00
481	KRANTI	NEHRU PLACE	3000.00

Code:

```
mysql> insert into borrow value
-> ('201','ANIL','VRCE',1000.00),
-> ('206','MEHUL','AJNI',5000.00),
-> ('311','SUNIL','DHARAMPETH',3000.00),
-> ('321','MADHURI','ANDHERI',2000.00),
-> ('375','PRAMOD','VIRAR',8000.00),
-> ('481','KRANTI','NEHRU PLACE',3000.00);
Query OK, 6 rows affected (0.04 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

Output:

```
mysql> select * from borrow;
+-----+-----+-----+-----+
| LOANNO | CNAME | BNAME | AMOUNT |
+-----+-----+-----+-----+
| 201   | ANIL  | VRCE  | 1000.00 |
| 206   | MEHUL | AJNI  | 5000.00 |
| 311   | SUNIL | DHARAMPETH | 3000.00 |
| 321   | MADHURI | ANDHERI | 2000.00 |
| 375   | PRAMOD | VIRAR | 8000.00 |
| 481   | KRANTI | NEHRU PLACE | 3000.00 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

From the above given tables perform the following queries:

1. Display branch, borrow, customer.

Command: describe branch;

```
describe borrow;
```

```
describe customer;
```

Output:

```
mysql> describe branch;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| BNAME | varchar(18) | YES  |     | NULL    |       |
| CITY  | varchar(18)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
```

```
mysql> describe borrow;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| LOANNO | varchar(5)  | YES  |     | NULL    |       |
| CNAME  | varchar(18) | YES  |     | NULL    |       |
| BNAME  | varchar(18) | YES  |     | NULL    |       |
| AMOUNT | decimal(8,2) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

```
mysql> describe customers;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| CNAME | varchar(18) | YES  |     | NULL    |       |
| CITY  | varchar(18) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

2. Describe deposit.

Command: describe deposit;

Output:

Field	Type	Null	Key	Default	Extra
ACTION	varchar(10)	YES		NULL	
CNAME	varchar(18)	YES		NULL	
BNAME	varchar(18)	YES		NULL	
AMOUNT	decimal(8,2)	YES		NULL	
ADATE	date	YES		NULL	

5 rows in set (0.00 sec)

3. List all data from table DEPOSIT.

Command: select * from deposit;

Output:

mysql> select * from deposit;					
ACTION	CNAME	BNAME	AMOUNT	ADATE	
100	ANIL	VRCE	1000.00	1995-03-01	
101	SUNIL	AJNI	5000.00	1996-01-04	
102	MEHUL	KAROLBAGH	3500.00	1995-11-17	
104	MADHURI	CHANDI	1200.00	1995-12-17	
105	PRAMOD	M.G.ROAD	3000.00	1996-03-31	
106	SANDIP	ANDHERI	2000.00	1996-03-31	
107	SHIVANI	VIRAR	1000.00	1995-09-05	
108	KRANTI	NEHRU PLACE	5000.00	1995-07-02	
109	MINU	POWAI	7000.00	1995-08-10	

9 rows in set (0.01 sec)

4. List all data from table BORROW.

Command: select * from borrow;

Output:

```
mysql> select * from borrow;
+-----+-----+-----+-----+
| LOANNO | CNAME   | BNAME    | AMOUNT  |
+-----+-----+-----+-----+
| 201    | ANIL     | VRCE     | 1000.00 |
| 206    | MEHUL    | AJNI     | 5000.00 |
| 311    | SUNIL    | DHARAMPETH | 3000.00 |
| 321    | MADHURI  | ANDHERI  | 2000.00 |
| 375    | PRAMOD   | VIRAR    | 8000.00 |
| 481    | KRANTI   | NEHRU PLACE | 3000.00 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

5. List all data from table CUSTOMERS.

Command: select * from customers;

Output:

```
mysql> select * from customers;
+-----+-----+
| CNAME | CITY    |
+-----+-----+
| ANIL  | CALCUTTA |
| SUNIL | DELHI   |
| MEHUL | BARODA  |
| MANDAR | PATNA |
| MADHURI | NAGPUR |
| SANDIP | SURAT   |
| SHIVANI | BOMBAY |
| KRANTI | BOMBAY |
| NAREN  | BOMBAY |
+-----+-----+
9 rows in set (0.00 sec)
```

6. List all data from table BRANCH.

Command: select * from branch;

Output:

```
mysql> select * from branch;
+-----+-----+
| BNAME      | CITY       |
+-----+-----+
| VRCE        | NAGPUR    |
| AJNI        | NAGPUR    |
| KAROLBAGH   | DELHI     |
| CHANDI      | DELHI     |
| DHARAMPETH  | NAGPUR    |
| M.G.ROAD    | BANGLORE  |
| ANDHERI     | BOMBAY    |
| VIRAR       | BOMBAY    |
| NEHRU PLACE | DELHI     |
| POWAI       | BOMBAY    |
+-----+-----+
10 rows in set (0.01 sec)
```

7. Give account no and amount of depositors.

Command: select ACTION,AMOUNT from deposit;

Output:

```
mysql> select ACTION,AMOUNT from deposit;
+-----+-----+
| ACTION | AMOUNT   |
+-----+-----+
| 100    | 1000.00  |
| 101    | 5000.00  |
| 102    | 3500.00  |
| 104    | 1200.00  |
| 105    | 3000.00  |
| 106    | 2000.00  |
| 107    | 1000.00  |
| 108    | 5000.00  |
| 109    | 7000.00  |
+-----+-----+
9 rows in set (0.00 sec)
```

8. Give name of depositors having amount greater than 4000.

Command: select CNAME from deposit where AMOUNT>4000;

Output:

```
mysql> select CNAME from deposit where AMOUNT>4000;
+-----+
| CNAME |
+-----+
| SUNIL |
| KRANTI |
| MINU  |
+-----+
3 rows in set (0.00 sec)
```

9. Give name of customers who opened account after date '1-12-95'.

Command: select CNAME from deposit where ADATE>1995-12-01;

Output:

```
mysql> select CNAME from deposit where ADATE>1995-12-01;
+-----+
| CNAME |
+-----+
| ANIL   |
| SUNIL  |
| MEHUL  |
| MADHURI|
| PRAMOD  |
| SANDIP  |
| SHIVANI|
| KRANTI |
| MINU   |
+-----+
9 rows in set, 1 warning (0.01 sec)
```

10. Write a query to get the records from deposit tables in which branch of the employee is VIRAR and amount is 1000.

Command: select * from deposit where BNAME = 'VIRAR' AND AMOUNT=1000.00;

Output:

```
mysql> select * from deposit where BNAME='VIRAR' and AMOUNT=1000.00;
+-----+-----+-----+-----+
| ACTION | CNAME | BNAME | AMOUNT | ADATE      |
+-----+-----+-----+-----+
| 107    | SHIVANI | VIRAR | 1000.00 | 1995-09-05 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

11. Write a query to get the records from BRANCH tables in which CITY of the employee is BOMBAY and branch name is POWAI.

Command: select * from branch where CITY='BOMBAY' AND BNAME='POWAI';

Output:

```
mysql> select * from branch where CITY='BOMBAY' and BNAME='POWAI';
+-----+
| BNAME | CITY   |
+-----+
| POWAI | BOMBAY |
+-----+
1 row in set (0.00 sec)
```

12. SQL "AND" example with "UPDATE" statement

Write a query to update the records in DEPOSIT tables in which BNAME of the employee is POWAI, and the first name is MINU. For that employee, set the updated value of the location as Delhi.

Command: update branch set CITY='DELHI' where BNAME='POWAI';

Output:

```
mysql> update branch set CITY='DELHI' where BNAME='POWAI';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1    Changed: 1    Warnings: 0

mysql> select * from branch;
+-----+-----+
| BNAME      | CITY      |
+-----+-----+
| VRCE        | NAGPUR    |
| AJNI        | NAGPUR    |
| KARBOLBAGH | DELHI     |
| CHANDI      | DELHI     |
| DHARAMPETH | NAGPUR    |
| M.G.ROAD    | BANGLORE  |
| ANDHERI    | BOMBAY    |
| VIRAR       | BOMBAY    |
| NEHRU PLACE | DELHI     |
| POWAI       | DELHI     |
+-----+-----+
10 rows in set (0.00 sec)
```

13. Write a query to get the records from deposit tables in which name of the employee is Sandip or location is VIRAR.

Command: select * from deposit where CNAME='SANDIP' or BNAME='VIRAR';

Output:

```
mysql> select * from deposit where CNAME='SANDIP' OR BNAME='VIRAR';
+-----+-----+-----+-----+-----+
| ACTION | CNAME   | BNAME   | AMOUNT | ADATE   |
+-----+-----+-----+-----+-----+
| 106    | SANDIP  | ANDHERI | 2000.00 | 1996-03-31 |
| 107    | SHIVANI | VIRAR   | 1000.00 | 1995-09-05 |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

14. SQL "OR" example with SQL UPDATE

Write a query to update the records in tables in BORROW in which name of the employee is anil, or the branch name is VRCE. For that employee, set the updated value of the amount as 10000.

Command: update borrow set AMOUNT=10000 where CNAME='ANIL' or BNAME='VRCE';

Output:

```
mysql> update borrow SET AMOUNT=10000 where CNAME='ANIL' OR BNAME='VRCE';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from borrow;
+-----+-----+-----+-----+
| LOANNO | CNAME   | BNAME    | AMOUNT  |
+-----+-----+-----+-----+
| 201    | ANIL    | VRCE     | 10000.00 |
| 206    | MEHUL   | AJNI     | 5000.00  |
| 311    | SUNIL   | DHARAMPETH | 3000.00 |
| 321    | MADHURI | ANDHERI  | 2000.00  |
| 375    | PRAMOD   | VIRAR    | 8000.00  |
| 481    | KRANTI  | NEHRU PLACE | 3000.00 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

15. Selects name and branch name of all customers whose amount is not 1000.

Command: select CNAME,BNAME from deposit where AMOUNT!=1000;

Output:

```
mysql> select CNAME,BNAME FROM deposit where AMOUNT!=1000;
+-----+-----+
| CNAME   | BNAME    |
+-----+-----+
| SUNIL   | AJNI     |
| MEHUL   | KAROLBAGH |
| MADHURI | CHANDI   |
| PRAMOD   | M.G.ROAD |
| SANDIP   | ANDHERI  |
| KRANTI  | NEHRU PLACE |
| MINU    | POWAI    |
+-----+-----+
7 rows in set (0.00 sec)
```

16. Select customers whose branch is either VIRAR or POWAI, and amount less than 6000.

Command: select CNAME from deposit where in ('VIRAR','POWAI') and AMOUNT<6000.00;

Output:

```
mysql> select CNAME from deposit where BNAME in ('VIRAR','POWAI') and AMOUNT
<6000.00;
+-----+
| CNAME   |
+-----+
| SHIVANI |
+-----+
1 row in set (0.00 sec)
```



LAB-4

Aim: To perform various data manipulation commands & aggregate functions on all created tables.

1. Create a following table: sales_master.

Col_name	Datatype	Size	Attribute
Salesman_no	varchar2	6	primary key
Sal_name	varchar2	20	not null
City	varchar2	40	not null
pincode	number	6,2	not null, cannot be 0
Sal_amt	number	6,2	not null, cannot be 0
target_to_get	number	6,2	not null, cannot be 0

Command:

```
create table sales_master(Salesman_no varchar(6),Sal_name
varchar(20)NOT NULL,City varchar(40) NOT NULL,pincode float (6,2) NOT
NULL,Sal_amt float(6,2) NOT NULL,target_to_get float (6,2) NOT NULL,
PRIMARY KEY(Salesman_no),check (pincode!=0 AND Sal_amt!=0 AND
target_to_get!=0));
```

Output:

```
mysql> describe sales_master;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Salesman_no | varchar(6) | NO | PRI | NULL | 
| Sal_name | varchar(20) | NO | NULL | NULL | 
| City | varchar(40) | NO | NULL | NULL | 
| pincode | float(6,2) | NO | NULL | NULL | 
| Sal_amt | float(6,2) | NO | NULL | NULL | 
| target_to_get | float(6,2) | NO | NULL | NULL | 
+-----+-----+-----+-----+-----+
6 rows in set (0.06 sec)
```

2. Create a following table: sales_order

Col_name	Datatype	Size	Attribute
S_order_no	varchar2	6	primary key
S_rder_date	date		not null
Salesman_no	number	40	Foreign key reference of salesman_no of sales_master table
Order_status	varchar	30	Status like in-process, fulfilled, cancelled
Dely_type	varchar	20	Part, full, defult-f
S_order_qunity	number		cannot be less than 2

Command:

```
create table sales_order(S_order_no varchar(6),S_rder_date date NOT
NULL,Salesman_no varchar(6),Order_status varchar(30),Dely_type
varchar(20) default 'f',S_order_qunity int,PRIMARY KEY
(S_order_no),check(Order_status='in-process' or Order_status='fulfilled'
or Order_status='cancelled'), check(Dely_type='Part' or Dely_type='full'),
check(S_order_qunity>=2));
```

Output:

```
mysql> describe sales_order;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| S_order_no | varchar(6) | NO | PRI | NULL |       |
| S_rder_date | date | NO |     | NULL |       |
| Salesman_no | varchar(6) | YES |     | NULL |       |
| Order_status | varchar(30) | YES |     | NULL |       |
| Dely_type | varchar(20) | YES |     | f    |       |
| S_order_qunity | int | YES |     | NULL |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

Insert Data in first table (sales_master):

```
insert into sales_master value
(409,'Kaushal','vadodra',391410,5000,7000);

insert into sales_master value
(404,'Harsh','vadodra',1410,5000,7000);

insert into sales_master value
(407,'Poojit','vadodra',1410,5000,7000)
```

Output:

```
mysql> select * from sales_master;
+-----+-----+-----+-----+-----+-----+
| Salesman_no | Sal_name | City     | pincode | Sal_amt | target_to_get |
+-----+-----+-----+-----+-----+-----+
| 404         | Harsh    | vadodra | 1410.00 | 5000.00 | 7000.00      |
| 407         | Poojit   | vadodra | 1410.00 | 5000.00 | 7000.00      |
| 409         | Kaushal  | vadodra | 1410.00 | 5000.00 | 7000.00      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Insert Data in Second table (sales_order):

```
insert into sales_order value('001','2023-12-23','01','fulfilled','full',5);

insert into sales_order value('019','2024-01-03','07','in-process','Part',5);

insert into sales_order value('056','2024-01-22','09','fulfilled','Part',5);
```

Output:

```
mysql> select * from sales_order;
+-----+-----+-----+-----+-----+-----+
| S_order_no | S_order_date | Salesman_no | Order_status | Dely_type | S_order_qunity |
+-----+-----+-----+-----+-----+-----+
| 001         | 2023-12-23  | 01          | fulfilled    | full      | 5             |
| 019         | 2024-01-03  | 07          | in-process   | Part      | 5             |
| 056         | 2024-01-22  | 09          | fulfilled    | Part      | 5             |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

From the Lab-1 given tables perform the following queries:

(1) List total deposit from deposit.

Command:

```
select sum(amount) from deposit;
```

Output:

```
mysql> select sum(amount) from deposit;
+-----+
| sum(amount) |
+-----+
|    28700.00 |
+-----+
1 row in set (0.06 sec)
```

(2) List total loan from karolbagh branch.

Command:

```
select sum(amount) from deposit where BNAME='KAROLBAGH';
```

Output:

```
mysql> select sum(amount) from deposit where BNAME='KAROLBAGH';
+-----+
| sum(amount) |
+-----+
|    3500.00 |
+-----+
1 row in set (0.01 sec)
```

(3) Give maximum loan from branch vrce.**Command:**

```
select max(amount) from borrow where bname='vrce';
```

Output:

```
mysql> select max(amount) from borrow where bname='vrce';
+-----+
| max(amount) |
+-----+
|    10000.00 |
+-----+
1 row in set (0.04 sec)
```

(4) Count total number of customers.**Command:**

```
select count(CNAME) from customers;
```

Output:

```
mysql> select count(CNAME) from customers;
+-----+
| count(CNAME) |
+-----+
|          9 |
+-----+
1 row in set (0.02 sec)
```

(5) Count total number of customer's cities.**Command:**

```
select count(CITY) from customers;
```

Output:

```
mysql> select count(CITY) from customers;
+-----+
| count(CITY) |
+-----+
|      9      |
+-----+
1 row in set (0.00 sec)
```

(6) Create table supplier from employee with all the columns.**Command:**

Create table:

```
create table employee(emp_id int,emp_f_name varchar(40),emp_l_name  
varchar(40),emp_email varchar(60));
```

Insert data into table:

```
insert into employee value (001,'Harsh','Patel','harsh@1gmail.com'),  
-> (002,'Poojit','Luhar','poojit02@gmail.com'),  
-> (003,'Kaushal','Ramoliya','kaushal03@gmail.com'),  
-> (004,'Alay','Patel','alay04@gmail.com');
```

Display table:

```
select*from employee;
```

```
mysql> select * from employee;
+-----+-----+-----+-----+
| emp_id | emp_f_name | emp_l_name | emp_email |
+-----+-----+-----+-----+
|     1 | Harsh      | Patel       | harsh@gmail.com |
|     2 | Poojit      | Luhar       | poojit02@gmail.com |
|     3 | Kaushal     | Ramoliya   | kaushal03@gmail.com |
|     4 | Alay        | Patel       | alay04@gmail.com |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Create supplier tabel:

```
create table supplier as select
emp_id,emp_f_name,emp_l_name,emp_email from employee;
```

Output:

```
mysql> select * from supplier;
+-----+-----+-----+-----+
| emp_id | emp_f_name | emp_l_name | emp_email |
+-----+-----+-----+-----+
|     1 | Harsh      | Patel       | harsh@gmail.com |
|     2 | Poojit      | Luhar       | poojit02@gmail.com |
|     3 | Kaushal     | Ramoliya   | kaushal03@gmail.com |
|     4 | Alay        | Patel       | alay04@gmail.com |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

(7) Create table sup1 from employee with first two columns.

Command:

```
create table sup_1 as select emp_id,emp_f_name from employee;
```

Output:

```
mysql> select * from sup_1;
+-----+-----+
| emp_id | emp_f_name |
+-----+-----+
|     1 | Harsh      |
|     2 | Poojit      |
|     3 | Kaushal     |
|     4 | Alay        |
+-----+-----+
4 rows in set (0.00 sec)
```

(8) Create table sup2 from employee with no data.**Command:**

```
create table sup_2 as select * from employee where 1=2;
```

```
select * from sup_2;
```

```
Desc sup_2;
```

Output:

```
mysql> select * from sup_2;
Empty set (0.00 sec)
```

```
mysql> desc sup_2;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_id | int    | YES  |     | NULL    |       |
| emp_f_name | varchar(40) | YES  |     | NULL    |       |
| emp_l_name | varchar(40) | YES  |     | NULL    |       |
| emp_email | varchar(60) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.06 sec)
```

(9) Insert the data into sup2 from employee whose second character should be 'n' and string should be 5 characters long in employee name field.**Command:**

```
Insetct into sup_2 select * from employee where =
emp_f_name='n__';
```

Output:

```
mysql> INSERT INTO sup2 SELECT * FROM employee where empFname like '_n___';
Query OK, 1 row affected (0.03 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> select * from sup2;
+-----+-----+-----+-----+
| empId | empFname | empLname | empEmail    |
+-----+-----+-----+-----+
| 410  | Ansha    | Patel    | xyz@nuv.ac.in |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

(10) Delete all the rows from sup1.**Command:**

```
delete from sup_1;
```

Output:

```
mysql> select * from sup_1;
Empty set (0.00 sec)
```

(11) Delete the detail of supplier whose sup_no is 103.**Command:**

```
delete from supplier where emp_id=003;
```

Output:

```
mysql> select * from supplier;
+-----+-----+-----+-----+
| emp_id | emp_f_name | emp_l_name | emp_email      |
+-----+-----+-----+-----+
|     1 | Harsh       | Patel       | harsh@1gmail.com |
|     2 | Poojita     | Luhar       | poojita02@gmail.com |
|     4 | Alay        | Patel       | alay04@gmail.com  |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

(12) Rename the table sup2**Command:**

```
alter table supplier rename to supplier_new;
```

Output:

```
mysql> select * from supplier_new;
+-----+-----+-----+-----+
| emp_id | emp_f_name | emp_l_name | emp_email |
+-----+-----+-----+-----+
|     1 | Harsh      | Patel       | harsh@1gmail.com |
|     2 | Poojit      | Luhar       | poojit02@gmail.com |
|     4 | Alay        | Patel       | alay04@gmail.com  |
+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

(13) Destroy table sup1 with all the data.**Command:**

```
drop table sup_1;
```

Output:

```
mysql> drop table sup_1;
Query OK, 0 rows affected (0.04 sec)
```

(14) Update the value dept_no to 10 where second character of emp. name is 'm'.**Command:**

```
update employee set emp_id = 10 where emp_f_name like '_l%';
```

Output:

```
mysql> select * from employee;
+-----+-----+-----+-----+
| emp_id | emp_f_name | emp_l_name | emp_email |
+-----+-----+-----+-----+
|     1 | Harsh      | Patel       | harsh@1gmail.com |
|     2 | Poojit      | Luhar       | poojit02@gmail.com |
|     3 | KAUSHAL    | Ramoliya   | kaushal03@gmail.com |
|    10 | Alay        | Patel       | alay04@gmail.com  |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

(15) Update the value of employee name whose employee number is 103

Command:

```
update employee set emp_f_name='KAUSHAL' where emp_id=003;
```

Output:

```
mysql> select * from employee;
+-----+-----+-----+-----+
| emp_id | emp_f_name | emp_l_name | emp_email |
+-----+-----+-----+-----+
|     1 | Harsh      | Patel       | harsh@1gmail.com |
|     2 | Poojita    | Luhar       | poojita02@gmail.com |
|     3 | KAUSHAL    | Ramoliya   | kaushal03@gmail.com |
|     4 | Alay        | Patel       | alay04@gmail.com  |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```



LAB-5

Aim: W3School SQL Queries

Create Database : create database w3query;

Use Database : use w3query;

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina	Berguvsvägen 8	Luleå	S-958 22	Sweden

Create Table:

Command:

```
mysql> CREATE TABLE Customers01
```

```
-> (
->   CustomerID INT,
->   CustomerName VARCHAR(50),
->   ContactName VARCHAR(50),
->   Address VARCHAR(50),
->   City VARCHAR(20),
->   PostalCode VARCHAR(10),
->   Country VARCHAR(15)
-> );
```

Insert data:**Command:**

```
mysql> INSERT INTO Customers01 VALUES
```

```
-> (1,'Alfreds Futterkiste','Maria Anders','Obere Str.  
57','Berlin','12209','Germany'),  
  
-> (2,'Ana Trujillo Emparedados y helados','Ana Trujillo','Avda. de la  
Constitución 2222','México D.F.','5021','Mexico'),  
  
-> (3,'Antonio Moreno Taquería','Antonio Moreno','Mataderos  
2312','México D.F.','5023','Mexico'),  
  
-> (4,'Around the Horn','Thomas Hardy','120 Hanover  
Sq.','London','WA1 1DP','UK'),  
  
-> (5,'Berglunds snabbköp','Christina Berglund','Berguvsvägen  
8','Luleå','S-958 22','Sweden');
```

Output:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	5021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	5023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

5 rows in set (0.00 sec)

CategoryID	CategoryName	Description
1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal

Create Table:**Command:**

```
mysql> CREATE TABLE Categories01
```

```
-> (
->   CategoryID INT,
->   CategoryName VARCHAR(25),
->   Description VARCHAR(255)
-> );
```

Insert data:**Command:**

```
mysql> INSERT INTO Categories01 VALUES
```

```
-> (1,'Beverages','Soft drinks, coffees, teas, beers, and ales'),
-> (2,'Condiments','Sweet and savory sauces, relishes, spreads, and
seasonings'),
-> (3,'Confections','Desserts, candies, and sweet breads'),
-> (4,'Dairy Products','Cheeses'),
-> (5,'Grains/Cereals','Breads, crackers, pasta, and cereal');
```

Output:

```
mysql> select * from Categories01;
+-----+-----+-----+
| CategoryID | CategoryName | Description |
+-----+-----+-----+
| 1 | Beverages | Soft drinks, coffees, teas, beers, and ales |
| 2 | Condiments | Sweet and savory sauces, relishes, spreads, and seasonings |
| 3 | Confections | Desserts, candies, and sweet breads |
| 4 | Dairy Products | Cheeses |
| 5 | Grains/Cereals | Breads, crackers, pasta, and cereal |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

EmployeeID	LastName	FirstName	BirthDate	Photo	Notes
1	Davolio	Nancy	12/8/1968	EmpID1.pic	Education includes a BA in psychology from Colorado State University. She also completed (The Art of the Cold Call). Nancy is a member of 'Toastmasters International'.
2	Fuller	Andrew	2/19/1952	EmpID2.pic	Andrew received his BTS commercial and a Ph.D. in international marketing from the University of Dallas. He is fluent in French and Italian and reads German. He joined the company as a sales representative, was promoted to sales manager and was then named vice president of sales. Andrew is a member of the Sales Management Roundtable, the Seattle Chamber of Commerce, and the Pacific Rim Importers Association.

Create Table:**Command:**

```
mysql> CREATE TABLE Employees01
```

```
-> (
->   EmployeeID INT,
->   LastName VARCHAR(15),
->   FirstName VARCHAR(15),
->   BirthDate DATETIME,
->   Photo VARCHAR(25),
->   Notes VARCHAR(1024)
-> );
```

Insert data:**Command:**

```
mysql> INSERT INTO Employees01 VALUES
```

```
-> (1,'Davolio','Nancy','1968-12-08','EmplID1.pic','Education includes a  
BA in psychology from Colorado State University.'),
```

```
-> (2,'Fuller','Andrew','1952-02-19','EmplID2.pic','Andrew received his  
BTS commercial and a Ph.D.');
```

Output:

```
mysql> select * from Employees01;
+-----+-----+-----+-----+-----+-----+
| EmployeeID | LastName | FirstName | BirthDate | Photo | Notes
+-----+-----+-----+-----+-----+-----+
| 1 | Davolio | Nancy | 1968-12-08 00:00:00 | EmplID1.pic | Education includes a BA in psychology from Colorado State University. |
| 2 | Fuller | Andrew | 1952-02-19 00:00:00 | EmplID2.pic | Andrew received his BTS commercial and a Ph.D. |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40

Create Table:**Command:**

```
mysql> CREATE TABLE OrderDetails(
```

```
-> OrderDetailID INT,  
-> OrderID INT,  
-> ProductID INT,  
-> Quantity INT);
```

Insert data:**Command:**

```
mysql> INSERT INTO OrderDetails VALUES
```

```
-> (1,10248,11,12),
-> (2,10248,42,10),
-> (3,10248,72,5),
-> (4,10249,14,9),
-> (5,10249,14,9);
```

Output:

```
mysql> select * from OrderDetails;
```

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	14	9

5 rows in set (0.00 sec)

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35

Create Table:**Command:**

```
mysql> CREATE TABLE Products01(
```

```
-> ProductID INT,
-> ProductName VARCHAR(50),
```

```

-> SupplierID INT,
-> CategoryID INT,
-> Unit VARCHAR(25),
-> Price INT
-> );

```

Insert data:**Command:**

```
mysql> INSERT INTO Products01 VALUES
```

```

-> (1,'Chais',1,1,'10 boxes x 20 bags',18),
-> (2,'Chang',1,1,'24 - 12 oz bottles',19),
-> (3,'Aniseed Syrup',1,2,'12 - 550 ml bottles',10),
-> (4,'Chef Anton''s Cajun Seasoning',2,2,'48 - 6 oz jars',22),
-> (5,'Chef Anton''s Gumbo Mix',2,2,'36 boxes',21.35);

```

Output:

```
mysql> select * from Products01;
+-----+-----+-----+-----+-----+-----+
| ProductID | ProductName | SupplierID | CategoryID | Unit | Price |
+-----+-----+-----+-----+-----+-----+
| 1 | Chais | 1 | 1 | 10 boxes x 20 bags | 18 |
| 2 | Chang | 1 | 1 | 24 - 12 oz bottles | 19 |
| 3 | Aniseed Syrup | 1 | 2 | 12 - 550 ml bottles | 10 |
| 4 | Chef Anton's Cajun Seasoning | 2 | 2 | 48 - 6 oz jars | 22 |
| 5 | Chef Anton's Gumbo Mix | 2 | 2 | 36 boxes | 21 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35

Create Table:**Command:**

```
mysql> CREATE TABLE Shippers01(
```

```
->     ShipperID INT,  
->     ShipperName VARCHAR(25),  
->     Phone VARCHAR(15)  
-> );
```

Insert data:**Command:**

```
mysql> INSERT INTO Shippers01 VALUES
```

```
-> (1,'Speedy Express', '(503) 555-9831'),  
-> (2,'United Package', '(503) 555-3199'),  
-> (3,'Federal Shipping', '(503) 555-9931');
```

Output:

```
mysql> select * from Shippers01;  
+-----+-----+-----+  
| ShipperID | ShipperName | Phone |  
+-----+-----+-----+  
| 1 | Speedy Express | (503) 555-9831 |  
| 2 | United Package | (503) 555-3199 |  
| 3 | Federal Shipping | (503) 555-9931 |  
+-----+-----+-----+  
3 rows in set (0.00 sec)
```

SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country	Phone
1	Exotic Liquid	Charlotte Cooper	49 Gilbert St.	Londona	EC1 4SD	UK	(171) 555-2222
2	New Orleans Cajun Delights	Shelley Burke	P.O. Box 78934	New Orleans	70117	USA	(100) 555-4822
3	Grandma Kelly's Homestead	Regina Murphy	707 Oxford Rd.	Ann Arbor	48104	USA	(313) 555-5735
4	Tokyo Traders	Yoshi Nagase	9-8 Sekimai Musashino-shi	Tokyo	100	Japan	(03) 3555-5011

Create Table:**Command:**

```
mysql> CREATE TABLE Suppliers02(  
->     SupplierID INT,  
->     SupplierName VARCHAR(50),  
->     ContactName VARCHAR(50),  
->     Address VARCHAR(50),  
->     City VARCHAR(20),  
->     PostalCode VARCHAR(10),  
->     Country VARCHAR(15),  
->     Phone VARCHAR(15)  
-> );
```

Insert data:**Command:**

```
mysql> INSERT INTO Suppliers02 VALUES  
-> (1,'Exotic Liquid','Charlotte Cooper','49 Gilbert St.','Londona','EC1  
4SD','UK','(171) 555-2222'),  
-> (2,'New Orleans Cajun Delights','Shelley Burke','P.O. Box 78934','New  
Orleans','70117','USA','(100) 555-4822'),  
-> (3,'Grandma Kelly''s Homestead','Regina Murphy','707 Oxford  
Rd.','Ann Arbor','48104','USA','(313) 555-5735'),  
-> (4,'Tokyo Traders','Yoshi Nagase','9-8 Sekimai Musashino-  
shi','Tokyo','100','Japan','(03) 3555-5011');
```

Output:

```
mysql> select * from Suppliers02;
+-----+-----+-----+-----+-----+-----+-----+
| SupplierID | SupplierName | ContactName | Address | City | PostalCode | Country | Phone |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Exotic Liquid | Charlotte Cooper | 49 Gilbert St. | Londona | EC1 4SD | UK | (171) 555-2222 |
| 2 | New Orleans Cajun Delights | Shelley Burke | P.O. Box 78934 | New Orleans | 70117 | USA | (100) 555-4822 |
| 3 | Grandma Kelly's Homestead | Regina Murphy | 787 Oxford Rd. | Ann Arbor | 48104 | USA | (313) 555-5735 |
| 4 | Tokyo Traders | Yoshi Nagase | 9-8 Sekimai Musashino-shi | Tokyo | 100 | Japan | (03) 3555-5011 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```



LAB-6

Aim: W3School Tables Creation

Your Database:

Tablenames	Records
<u>Customers</u>	91
<u>Categories</u>	8
<u>Employees</u>	10
<u>OrderDetails</u>	518
<u>Orders</u>	196
<u>Products</u>	77
<u>Shippers</u>	3
<u>Suppliers</u>	29

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
6	Blauer See	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany

SQL_query

Select Statement:

Q.1 select customerName,city from Customers01;

```
mysql> select customerName,city from Customers01;
+-----+-----+
| customerName | city
+-----+-----+
| Alfreds Futterkiste | Berlin
| Ana Trujillo Emparedados y helados | México D.F.
| Antonio Moreno Taquería | México D.F.
| Around the Horn | London
| Berglunds snabbköp | Luleå
+-----+-----+
5 rows in set (0.12 sec)
```

Q.2 select * from Customers01;

```
mysql> select * from Customers01;
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
| 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
| 5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Q.3 select distinct Country from Customers01;

```
mysql> select distinct Country from Customers01;
+-----+
| Country |
+-----+
| Germany |
| Mexico |
| UK |
| Sweden |
+-----+
4 rows in set (0.01 sec)
```

Q.4 select Country from Customers01;

```
mysql> select Country from Customers01;
+-----+
| Country |
+-----+
| Germany |
| Mexico  |
| Mexico  |
| UK      |
| Sweden  |
+-----+
5 rows in set (0.00 sec)
```

Q.5 select count(distinct Country) from Customers01;

```
mysql> select count(distinct Country) from Customers01;
+-----+
| count(distinct Country) |
+-----+
|                      4 |
+-----+
1 row in set (0.01 sec)
```

Q.6 select * from Customers01 where Country = 'Mexico';

```
mysql> select * from Customers01 where Country = 'Mexico';
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
|          2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
|          3 | Antonio Moreno Taqueria | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

Q.7 select * from Customers01 where CustomerID=1;

```
mysql> select * from Customers01 where CustomerID=1;
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
|          1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

Q.8 select * from Customers01 where CustomerID>1;

```
mysql> select * from Customers01 where CustomerID>1;
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
| 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
| 5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Q.9 select * from Customers01 where Country = 'Mexico' AND CustomerID = 2;

```
mysql> select * from Customers01 where Country = 'Mexico' AND CustomerID = 2;
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Q.10 mysql> SELECT * FROM Customers01 WHERE Country = 'Germany' AND City = 'Berlin' AND PostalCode > 1200;

```
mysql> SELECT * FROM Customers01 WHERE Country = 'Germany' AND City = 'Berlin' AND PostalCode > 1200;
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
| 1 | Alfs Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.04 sec)
```

Q.11SELECT * FROM Customers WHERE Country = 'Spain' AND (CustomerName LIKE 'G%' OR CustomerName LIKE 'R%');

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
29	Galeria del gastrónomo	Eduardo Saavedra	Rambla de Cataluña, 23	Barcelona	08022	Spain
30	Godos Cocina Típica	José Pedro Freyre	C/ Romero, 33	Sevilla	41101	Spain
69	Romero y tomillo	Alejandra Camino	Gran Vía, 1	Madrid	28001	Spain

Q.12 SELECT * FROM Customers WHERE Country = 'Spain' AND CustomerName LIKE 'G%' OR CustomerName LIKE 'R%';

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
29	Galeria del gastrónomo	Eduardo Saavedra	Rambla de Cataluña, 23	Barcelona	08022	Spain
30	Godos Cocina Típica	José Pedro Freyre	C/ Romero, 33	Sevilla	41101	Spain
64	Rancho grande	Sergio Gutiérrez	Av. del Libertador 900	Buenos Aires	1010	Argentina
65	Rattlesnake Canyon Grocery	Paula Wilson	2817 Milton Dr.	Albuquerque	87110	USA
66	Reggiani Caseifici	Maurizio Moroni	Strada Provinciale 124	Reggio Emilia	42100	Italy
67	Ricardo Adocicados	Janete Limeira	Av. Copacabana, 267	Rio de Janeiro	02389-890	Brazil
68	Richter Supermarkt	Michael Holz	Grenzacherweg 237	Genève	1203	Switzerland
69	Romero y tomillo	Alejandra Camino	Gran Vía, 1	Madrid	28001	Spain

Q.13 select * from Customers01 where Country = 'Mexico' OR Country = 'Germany';

```
mysql> select * from Customers01 where Country = 'Mexico' OR Country = 'Germany';
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
| 1 | Alfréd Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Q.14 SELECT * FROM Customers WHERE City = 'Berlin' OR CustomerName LIKE 'G%' OR Country = 'Norway';

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
29	Galería del gastrónomo	Eduardo Saavedra	Rambla de Cataluña, 23	Barcelona	08022	Spain
30	Godos Cocina Típica	José Pedro Freyre	C/ Romero, 33	Sevilla	41101	Spain
31	Gourmet Lanchonetes	André Fonseca	Av. Brasil, 442	Campinas	04876-786	Brazil
32	Great Lakes Food Market	Howard Snyder	2732 Baker Blvd.	Eugene	97403	USA
33	GROSELLA-Restaurante	Manuel Pereira	5ª Ave. Los Palos Grandes	Caracas	1081	Venezuela
70	Santé Gourmet	Jonas Bergulfsen	Erling Skakkes gate 78	Stavern	4110	Norway

Q.15 SELECT * FROM Customers WHERE Country = 'Spain' AND (CustomerName LIKE 'G%' OR CustomerName LIKE 'R%');

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
29	Galería del gastrónomo	Eduardo Saavedra	Rambla de Cataluña, 23	Barcelona	08022	Spain
30	Godos Cocina Típica	José Pedro Freyre	C/ Romero, 33	Sevilla	41101	Spain
69	Romero y tomillo	Alejandra Camino	Gran Vía, 1	Madrid	28001	Spain

Q.16SELECT * FROM Customers WHERE Country = 'Spain' AND CustomerName LIKE 'G%' OR CustomerName LIKE 'R%';

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
29	Galería del gastrónomo	Eduardo Saavedra	Rambla de Cataluña, 23	Barcelona	08022	Spain
30	Godos Cocina Típica	José Pedro Freyre	C/ Romero, 33	Sevilla	41101	Spain
64	Rancho grande	Sergio Gutiérrez	Av. del Libertador 900	Buenos Aires	1010	Argentina
65	Rattlesnake Canyon Grocery	Paula Wilson	2817 Milton Dr.	Albuquerque	87110	USA
66	Reggiani Caseifaci	Maurizio Moroni	Strada Provinciale 124	Reggio Emilia	42100	Italy
67	Ricardo Adocicados	Janete Limeira	Av. Copacabana, 267	Rio de Janeiro	02389-890	Brazil
68	Richter Supermarkt	Michael Holz	Grenzacherweg 237	Genève	1203	Switzerland
69	Romero y tomillo	Alejandra Camino	Gran Vía, 1	Madrid	28001	Spain

Q.17 select * from Customers01 where NOT Country = 'Mexico' ;

```
mysql> select * from Customers01 where NOT Country = 'Mexico' ;
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
|      1 | Al�reds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
|      4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
|      5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Q.18 select * from Customers01 where CustomerName NOT LIKE 'A%';

```
mysql> select * from Customers01 where CustomerName NOT LIKE 'A%';
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
|      5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

Q.19 select * from Customers01 where CustomerID NOT BETWEEN 3 and 5;

```
mysql> select * from Customers01 where CustomerID NOT BETWEEN 3 and 5;
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
|      1 | Al�reds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
|      2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

Q.20 select * from Customers01 where Country NOT IN ('Germany', 'UK');

```
mysql> select * from Customers01 where Country NOT IN ('Germany', 'UK');
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
|      2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
|      3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
|      5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Q.21 select * from Customers01 where NOT CustomerID >3;

```
mysql> select * from Customers01 where NOT CustomerID >3;
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
|      1 | Al�reds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
|      2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
|      3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Q.22 mysql> select * from Customers01 where NOT CustomerID <3;

```
mysql> select * from Customers01 where NOT CustomerID <3;
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
|      3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
|      4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
|      5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Q.23
INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country) VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');

Q.24
INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES

('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway'),
 ('Greasy Burger', 'Per Olsen', 'Gateveien 15', 'Sandnes', '4306', 'Norway'),
 ('Tasty Tee', 'Finn Egan', 'Streetroad 19B', 'Liverpool', 'L1 0AA', 'UK');

Q.25
SELECT CustomerName, ContactName, Address FROM Customers WHERE Address IS NULL;

Q.26
SELECT CustomerName, ContactName, Address FROM Customers WHERE Address IS NOT NULL;

CustomerName	ContactName	Address
Alfreds Futterkiste	Maria Anders	Obere Str. 57
Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222
Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312
Around the Horn	Thomas Hardy	120 Hanover Sq.
Berglunds snabbköp	Christina Berglund	Berguvsvägen 8
Blauer See Delikatessen	Hanna Moos	Forsterstr. 57
Blondel père et fils	Frédérique Citeaux	24, place Kléber
Bólido Comidas preparadas	Martín Sommer	C/ Araquil, 67

Q.27
DELETE FROM table_name WHERE condition;

Q.28
DELETE FROM table_name;

Q.29
DELETE FROM Customers;

Q.30
DROP TABLE Customers;

Q.31 SELECT TOP 3 * FROM Customers;

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

Q.32 SELECT * FROM Customers LIMIT 3;

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

Q.33 SELECT MIN(Price) FROM Products;

Expr1000

2.5

Q.34 SELECT MAX(Price) FROM Products;

Expr1000

263.5

Q.35 SELECT MIN(Price) AS SmallestPrice FROM Products;

SmallestPrice

2.5

Q.36 SELECT COUNT(*) FROM Products;

Expr1000

77

Q.37 SELECT COUNT(ProductID) FROM Products WHERE price > 20;

Expr1000

37

Q.38 SELECT COUNT(ProductName) FROM Products;

Expr1000

77

Q.39 SELECT COUNT(DISTINCT Price) FROM Products;

COUNT(DISTINCT Price)

62

Q.40 SELECT COUNT(*) AS [number of records] FROM Products;

number of records

77

Q.41 SELECT SUM(Quantity) FROM OrderDetails;

Expr1000

12743

Q.42 SELECT SUM(Quantity) FROM OrderDetails WHERE ProductID = 11;

Expr1000

182

Q.43 SELECT SUM(Quantity) AS total FROM OrderDetails;

total

12743

Q.44 SELECT SUM(Quantity * 10) FROM OrderDetails;

Expr1000

127430

Q.45 SELECT SUM(Price * Quantity) FROM OrderDetails LEFT JOIN Products ON OrderDetails.ProductID = Products.ProductID;

Expr1000

386424.23

Q.46 SELECT AVG(Price) FROM Products;

Expr1000

28.8664

Q.47 SELECT AVG(Price) FROM Products WHERE CategoryID = 1;

Expr1000

37.9792

Q.48 SELECT AVG(Price) AS [average price] FROM Products;

average price

28.8664

Q.49 SELECT * FROM Products WHERE price > (SELECT AVG(price) FROM Products);

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	30
8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40
9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97
10	Ikura	4	8	12 - 200 ml jars	31

Q.50 select * from Products01 where price BETWEEN 10 AND 20;

```
mysql> select * from Products01 where price BETWEEN 10 AND 20;
+-----+-----+-----+-----+-----+
| ProductID | ProductName | SupplierID | CategoryID | Unit
| Price |
+-----+-----+-----+-----+-----+
|      1 | Chais        |      1 |          1 | 10 boxes x 20 bags |    18 |
|      2 | Chang        |      1 |          1 | 24 - 12 oz bottles |    19 |
|      3 | Aniseed Syrup |      1 |          2 | 12 - 550 ml bottles |    10 |
+-----+-----+-----+-----+-----+
3 rows in set (0.04 sec)
```

Q.51select * from Products01 where price NOT BETWEEN 10 AND 20;

```
mysql> select * from Products01 where price NOT BETWEEN 10 AND 20;
+-----+-----+-----+-----+-----+
| ProductID | ProductName | SupplierID | CategoryID | Unit | Price |
+-----+-----+-----+-----+-----+
|        4 | Chef Anton's Cajun Seasoning |        2 |        2 | 48 - 6 oz jars |    22 |
|        5 | Chef Anton's Gumbo Mix |        2 |        2 | 36 boxes |    21 |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Q.52select * from orders01 where OrderDate '07/01/1996' AND '07/31/1996';

Q.53select * from Customers01 where Country IN ('Germany','France','UK');

```
mysql> select * from Customers01 where Country IN ('Germany', 'France', 'UK');
+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+
|        1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
|        4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Q.54select * from Customers01 where Country NOT IN ('Germany','France','UK');

```
mysql> select * from Customers01 where Country NOT IN ('Germany', 'France', 'UK');
+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+
|        2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
|        3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
|        5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Q.55select * from Products01 where Price BETWEEN 10 AND 20 AND CategoryID IN (1,2,3);

```
mysql> select * from Products01 where Price BETWEEN 10 AND 20 AND CategoryID IN (1,2,3);
+-----+-----+-----+-----+-----+
| ProductID | ProductName | SupplierID | CategoryID | Unit | Price |
+-----+-----+-----+-----+-----+
|        1 | Chais |        1 |        1 | 10 boxes x 20 bags |    18 |
|        2 | Chang |        1 |        1 | 24 - 12 oz bottles |    19 |
|        3 | Aniseed Syrup |        1 |        2 | 12 - 550 ml bottles |    10 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Q.56select * from Customers01 where CustomerName LIKE 'A%';

```
mysql> select * from Customers01 where CustomerName LIKE 'A%';
+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+
| 1 | Al�reds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
| 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Q.57 mysql> select * from Customers01 where Country LIKE 'G_rm__';

```
mysql> select * from Customers01 where Country LIKE 'G_rm__';
+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+
| 1 | Al�reds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Q.58 select * from Customers01 where Country LIKE '%ny';

```
mysql> select * from Customers01 where Country LIKE '%ny';
+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+
| 1 | Al�reds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Q.59 select * from Customers01 where CustomerName LIKE '%ste%';

```
mysql> select * from Customers01 where CustomerName LIKE '%ste%';
+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+
| 1 | Al�reds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Q.60 select * from Customers01 where CustomerName LIKE 'Al%';

```
mysql> select * from Customers01 where CustomerName LIKE 'Al%';
+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+
| 1 | Al�reds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Q.61 select * from Customers01 where CustomerName LIKE 'a%' OR CustomerName LIKE 'b%';

```
mysql> select * from Customers01 where CustomerName LIKE 'a%' OR CustomerName LIKE 'b%';
+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+
| 1 | Al�reds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
| 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
| 5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Q.62 select * from Customers01 where CustomerName LIKE '%a';

```
mysql> select * from Customers01 where CustomerName LIKE '%a';
+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+
| 3 | Antonio Moreno Taqueria | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Q.63 select * from Customers01 where CustomerName LIKE 'a%e';

```
mysql> select * from Customers01 where CustomerName LIKE 'a%e';
+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+
| 1 | Al�eds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Q.64 select * from Customers01 where CustomerName LIKE 'a__%';

```
mysql> select * from Customers01 where CustomerName LIKE 'a__%';
+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+
| 1 | Al�eds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
| 3 | Antonio Moreno Taqueria | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
| 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Q.65 select * from Customers01 where CustomerName LIKE '_r%';

```
mysql> select * from Customers01 where CustomerName LIKE '_r%';
+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+
| 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Q.66 select * from Customers01 where CustomerName LIKE '[a-f]%';

Q.67 select * from Products01 ORDER BY price;

```
mysql> select * from Products01 ORDER BY price;
+-----+-----+-----+-----+-----+
| ProductID | ProductName | SupplierID | CategoryID | Unit | Price |
+-----+-----+-----+-----+-----+
| 3 | Aniseed Syrup | 1 | 2 | 12 - 550 ml bottles | 10 |
| 1 | Chais | 1 | 1 | 10 boxes x 20 bags | 18 |
| 2 | Chang | 1 | 1 | 24 - 12 oz bottles | 19 |
| 5 | Chef Anton's Gumbo Mix | 2 | 2 | 36 boxes | 21 |
| 4 | Chef Anton's Cajun Seasoning | 2 | 2 | 48 - 6 oz jars | 22 |
+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Q.68 mysql> select * from Products01 ORDER BY price ASC;

```
mysql> select * from Products01 ORDER BY price ASC;
+-----+-----+-----+-----+-----+
| ProductID | ProductName | SupplierID | CategoryID | Unit | Price |
+-----+-----+-----+-----+-----+
| 3 | Aniseed Syrup | 1 | 2 | 12 - 550 ml bottles | 10 |
| 1 | Chais | 1 | 1 | 10 boxes x 20 bags | 18 |
| 2 | Chang | 1 | 1 | 24 - 12 oz bottles | 19 |
| 5 | Chef Anton's Gumbo Mix | 2 | 2 | 36 boxes | 21 |
| 4 | Chef Anton's Cajun Seasoning | 2 | 2 | 48 - 6 oz jars | 22 |
+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Q.69 select * from Products01 ORDER BY ProductName;

```
mysql> select * from Products01 ORDER BY ProductName;
+-----+-----+-----+-----+-----+
| ProductID | ProductName | SupplierID | CategoryID | Unit | Price |
+-----+-----+-----+-----+-----+
| 3 | Aniseed Syrup | 1 | 2 | 12 - 550 ml bottles | 10 |
| 1 | Chais | 1 | 1 | 10 boxes x 20 bags | 18 |
| 2 | Chang | 1 | 1 | 24 - 12 oz bottles | 19 |
| 4 | Chef Anton's Cajun Seasoning | 2 | 2 | 48 - 6 oz jars | 22 |
| 5 | Chef Anton's Gumbo Mix | 2 | 2 | 36 boxes | 21 |
+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Q.70 select * from Products01 ORDER BY ProductName desc;

```
mysql> select * from Products01 ORDER BY ProductName desc;
+-----+-----+-----+-----+-----+
| ProductID | ProductName | SupplierID | CategoryID | Unit | Price |
+-----+-----+-----+-----+-----+
| 5 | Chef Anton's Gumbo Mix | 2 | 2 | 36 boxes | 21 |
| 4 | Chef Anton's Cajun Seasoning | 2 | 2 | 48 - 6 oz jars | 22 |
| 2 | Chang | 1 | 1 | 24 - 12 oz bottles | 19 |
| 1 | Chais | 1 | 1 | 10 boxes x 20 bags | 18 |
| 3 | Aniseed Syrup | 1 | 2 | 12 - 550 ml bottles | 10 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Q.71 select * from Customers01 ORDER BY Country, CustomerName;

```
mysql> select * from Customers01 ORDER BY Country, CustomerName;
+-----+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+-----+
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
| 5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden |
| 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Q.72 select * from Customers01 ORDER BY Country asc, CustomerName desc;

```
mysql> select * from Customers01 ORDER BY Country asc, CustomerName desc;
+-----+-----+-----+-----+-----+
| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
+-----+-----+-----+-----+-----+
| 1 | Alfs Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 5023 | Mexico |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 5021 | Mexico |
| 5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden |
| 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```



LAB-7

Aim: W3School SQL Practice.

GROUP BY

Command:

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;
```

Output:

Number of Records: 21

Expr1000	Country
3	Argentina
2	Austria
2	Belgium
9	Brazil
3	Canada
2	Denmark
2	Finland
11	France

Command:

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
ORDER BY COUNT(CustomerID) DESC;
```

Output:

Number of Records: 21

Expr1000	Country
13	USA
11	France
11	Germany
9	Brazil
7	UK
5	Mexico
5	Spain
4	Venezuela

Command:

```
SELECT Shippers.ShipperName,COUNT(Orders.OrderID) AS
NumberOfOrders FROM Orders
LEFT JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID
GROUP BY ShipperName;
```

Output:

Number of Records: 3

ShipperName	NumberOfOrders
Federal Shipping	68
Speedy Express	54
United Package	74

HAVING**Command:**

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5;
```

Output:

Number of Records: 5

Expr1000	Country
9	Brazil
11	France
11	Germany
7	UK
13	USA

Command:

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
HAVING COUNT(CustomerID) > 5  
ORDER BY COUNT(CustomerID) DESC;
```

Output:

Number of Records: 5

Expr1000	Country
9	Brazil
11	France
11	Germany
7	UK
13	USA

Command:

```
SELECT Employees.LastName, COUNT(Orders.OrderID) AS  
NumberOfOrders  
FROM (Orders  
INNER JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID)  
GROUP BY LastName  
HAVING COUNT(Orders.OrderID) > 10;
```

Output:

Number of Records: 8

Last Name	Number of Orders
Buchanan	11
Callahan	27
Davolio	29
Fuller	20
King	14
Leverling	31
Peacock	40
Suyama	18

Command:

```

SELECT Employees.LastName, COUNT(Orders.OrderID) AS
NumberOfOrders
FROM Orders
INNER JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
WHERE LastName = 'Davolio' OR LastName = 'Fuller'
GROUP BY LastName
HAVING COUNT(Orders.OrderID) > 25;

```

Output:

Number of Records: 1

Last Name	Number of Orders
Davolio	29

ANY and ALL

Command:

```
SELECT ProductName  
FROM Products  
WHERE ProductID = ANY (SELECT ProductID FROM OrderDetails WHERE  
Quantity = 10);
```

Output:

Number of Records: 31

ProductName
Chais
Chang
Chef Anton's Cajun Seasoning
Uncle Bob's Organic Dried Pears
Konbu

Command:

```
SELECT ProductName  
FROM Products  
WHERE ProductID = ANY (SELECT ProductID FROM OrderDetails WHERE  
Quantity > 99);
```

Output:

Number of Records: 2

ProductName
Steeleye Stout
Pâté chinois

Command:

```
SELECT ProductName  
FROM Products  
WHERE ProductID = ANY (SELECT ProductID FROM OrderDetails WHERE  
Quantity > 1000);
```

Output:

Number of Records: 0

ProductName

Command:

```
SELECT ALL ProductName  
FROM Products  
WHERE TRUE;
```

Output:

Number of Records: 77

ProductName
Chais
Chang
Aniseed Syrup
Chef Anton's Cajun Seasoning
Chef Anton's Gumbo Mix
Grandma's Boysenberry Spread
Uncle Bob's Organic Dried Pears

Command:

```
SELECT ProductName  
FROM Products  
WHERE ProductID = ALL (SELECT ProductID FROM OrderDetails WHERE  
Quantity = 10);
```

Output:

Number of Records: 0

ProductName

SQL DATABASE

CREATE DATABASE

Command:

```
CREATE DATABASE databasename;
```

DROP DATABASE

Command:

```
DROP DATABASE databasename;
```

CREATE TABLE

Syntax:

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
    ....
);
```

Command:

```
CREATE TABLE Persons (
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

Command:

```
CREATE TABLE TestTable AS
SELECT customername, contactname
FROM customers;
```

DROP TABLE

Syntax:

```
DROP TABLE table_name;
```

ALTER TABLE

Syntax:

```
ALTER TABLE table_name  
ADD column_name datatype;
```

Command:

```
ALTER TABLE Customers  
ADD Email varchar(255);
```

Command:

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

Command:

```
ALTER TABLE Customers  
DROP COLUMN Email;
```

Command:

```
ALTER TABLE table_name  
RENAME COLUMN old_name to new_name;
```

Command:

```
ALTER TABLE Persons  
ADD DateOfBirth date;
```

Command:

```
ALTER TABLE Persons  
ALTER COLUMN DateOfBirth year;
```

Command:

```
ALTER TABLE Persons  
DROP COLUMN DateOfBirth;
```

Constraints

NOT NULL

Command:

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    Age int  
)
```

Command:

```
ALTER TABLE Persons  
ALTER COLUMN Age int NOT NULL;
```

UNIQUE

Command:

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,
```

```
FirstName varchar(255),  
Age int,  
UNIQUE (ID)  
);
```

Command:

```
ALTER TABLE Persons  
ADD UNIQUE (ID);
```

Command:

```
ALTER TABLE Persons  
DROP INDEX UC_Person;
```

PRIMARY KEY

Command:

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY (ID)  
);
```

Command:

```
ALTER TABLE Persons  
ADD PRIMARY KEY (ID);
```

Command:

```
ALTER TABLE Persons  
DROP PRIMARY KEY;
```

FOREIGN KEY

Command:

```
CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);
```

Command:

```
ALTER TABLE Orders  
ADD FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);
```

Command:

```
ALTER TABLE Orders  
DROP FOREIGN KEY FK_PersonOrder;
```

CHECK

Command:

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

Command:

```
ALTER TABLE Persons
ADD CHECK (Age>=18);
```

Command:

```
ALTER TABLE Persons
DROP CONSTRAINT CHK_PersonAge;
```

DEFAULT

Command:

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    City varchar(255) DEFAULT 'Sandnes'
);
```

Command:

```
CREATE TABLE Orders (
    ID int NOT NULL,
    OrderNumber int NOT NULL,
    OrderDate date DEFAULT GETDATE()
);
```

Command:

```
ALTER TABLE Persons
ALTER City SET DEFAULT 'Sandnes';
```

Command:

```
ALTER TABLE Persons
ALTER City DROP DEFAULT;
```

Dates

Command:

```
SELECT * FROM Orders WHERE OrderDate='2008-11-11'
```



LAB-8

Aim: SQL Practice.

Create Tables:

Table-1:

Command:

```
create table salesman1(salesman_id int,name varchar(20),city  
varchar(20),commission float);
```

```
mysql> insert into salesman1 value(5001,'James Hooq','New York',0.15),  
-> (5002,'Nail Knite','Paris',0.13),  
-> (5005,'Pit Alex','London',0.11),  
-> (5006,'Mc Lyon','Paris',0.14),  
-> (5003,'Lauson Hen','','0.12),  
-> (5007,'Paul Adam','Rome',0.13);
```

```
mysql> select * from salesman1;  
+-----+-----+-----+-----+  
| salesman_id | name      | city     | commission |  
+-----+-----+-----+-----+  
|      5001  | James Hooq  | New York |      0.15  |  
|      5002  | Nail Knite | Paris    |      0.13  |  
|      5005  | Pit Alex   | London   |      0.11  |  
|      5006  | Mc Lyon   | Paris    |      0.14  |  
|      5003  | Lauson Hen |          |      0.12  |  
|      5007  | Paul Adam  | Rome    |      0.13  |  
+-----+-----+-----+-----+  
6 rows in set (0.01 sec)
```

Table-2:**Command:**

```
create table customer1(customer_id int, customer_name varchar(30), city
varchar(20), grade int, salesman_id int, primary
key(customer_id), FOREIGN KEY (salesman_id) REFERENCES
salesman1(salesman_id));
```

Query OK, 0 rows affected (0.05 sec)

```
mysql> insert into customer1 value(3002,'Nick Rimando','New
York',100,5001),
```

```
-> (3005,'Graham Zusi','California',200,5002),
-> (3001,'Brad Guzan','London',NULL,NULL),
-> (3004,'Fabian Johns','Paris',300,5006),
-> (3007,'Brad Davis','New York',200,5001),
-> (3009,'Geoff Camero','Berlin',100,NULL),
-> (3008,'Julian Green','London',300,5002),
-> (3003,'Jozy Altidor','Moncow',200,5007);
```

Query OK, 8 rows affected (0.01 sec)

```
mysql> select * from customer1;
+-----+-----+-----+-----+-----+
| customer_id | customer_name | city | grade | salesman_id |
+-----+-----+-----+-----+-----+
| 3001 | Brad Guzan | London | NULL | NULL |
| 3002 | Nick Rimando | New York | 100 | 5001 |
| 3003 | Jozy Altidor | Moncow | 200 | 5007 |
| 3004 | Fabian Johns | Paris | 300 | 5006 |
| 3005 | Graham Zusi | California | 200 | 5002 |
| 3007 | Brad Davis | New York | 200 | 5001 |
| 3008 | Julian Green | London | 300 | 5002 |
| 3009 | Geoff Camero | Berlin | 100 | NULL |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Table-3:**Command:**

```
create table order1(order_no int Primary key,purch_amt  
float(6,2),order_date Date,customer_id int, salesman_id int);
```

Query OK, 0 rows affected, 1 warning (0.04 sec)

```
alter table orders1 add foreign key(customer_id) references  
customer1(customer_id);
```

Query OK, 0 rows affected (0.09 sec)

```
alter table orders1 add foreign key(salesman_id) references  
salesman1(salesman_id);;
```

Query OK, 0 rows affected (0.09 sec)

```
insert into orders1 value(70009,270.65,'2016-09-10',3001,NULL),  
-> (70002,65.26,'2016-10-05',3002,5001),  
-> (70004,110.5,'2016-08-17',3009,NULL),  
-> (70007,948.5,'2016-09-10',3005,5002),  
-> (70005,2400.6,'2016-07-27',3007,5001),  
-> (70008,5760,'2016-09-10',3002,5001),  
-> (70010,1983.43,'2016-10-10',3004,5006),  
-> (70003,2480.4,'2016-10-10',3009,NULL),  
-> (70012,250.45,'2016-06-27',3008,5002),  
-> (70011,75.29,'2016-08-17',3003,5007);
```

Query OK, 10 rows affected (0.01 sec)

```
mysql> select * from orders1;
+-----+-----+-----+-----+-----+
| order_no | purch_amt | order_date | customer_id | salesman_id |
+-----+-----+-----+-----+-----+
| 70002 | 65.26 | 2016-10-05 | 3002 | 5001 |
| 70003 | 2480.40 | 2016-10-10 | 3009 | NULL |
| 70004 | 110.50 | 2016-08-17 | 3009 | NULL |
| 70005 | 2400.60 | 2016-07-27 | 3007 | 5001 |
| 70007 | 948.50 | 2016-09-10 | 3005 | 5002 |
| 70008 | 5760.00 | 2016-09-10 | 3002 | 5001 |
| 70009 | 270.65 | 2016-09-10 | 3001 | NULL |
| 70010 | 1983.43 | 2016-10-10 | 3004 | 5006 |
| 70011 | 75.29 | 2016-08-17 | 3003 | 5007 |
| 70012 | 250.45 | 2016-06-27 | 3008 | 5002 |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

queries

Query-1: Display name and commission of all the salesmen.

Command:

```
select name,commission from salesman1;
```

Output:

```
mysql> select name,commission from salesman1;
+-----+-----+
| name | commission |
+-----+-----+
| James Hooq | 0.15 |
| Nail Knite | 0.13 |
| Lauson Hen | 0.12 |
| Pit Alex | 0.11 |
| Mc Lyon | 0.14 |
| Paul Adam | 0.13 |
+-----+-----+
6 rows in set (0.07 sec)
```

Query-2: Retrieve salesman id of all salesmen from orders table without any repeats.

Command:

```
select distinct salesman_id from orders1;
```

Output:

```
mysql> select distinct salesman_id from orders1;
+-----+
| salesman_id |
+-----+
|      NULL  |
|      5001   |
|      5002   |
|      5006   |
|      5007   |
+-----+
5 rows in set (0.01 sec)
```

Query 3: Display names and city of salesman, who belongs to the city of Paris.

Command:

```
select name,city from salesman1 where city='Paris';
```

Output:

```
mysql> select name,city from salesman1 where city='Paris';
+-----+-----+
| name    | city   |
+-----+-----+
| Nail Knite | Paris |
| Mc Lyon    | Paris |
+-----+-----+
2 rows in set (0.01 sec)
```

Query 4: Display all the information for those customers with a grade of 200.

Command:

```
select * from customer1 where grade=200;
```

Output:

```
mysql> select * from customer1 where grade=200;
+-----+-----+-----+-----+
| customer_id | customer_name | city      | grade | salesman_id |
+-----+-----+-----+-----+
|     3003 | Jozy Altidor   | Moncow    | 200   |      5007 |
|     3005 | Graham Zusie  | California | 200   |      5002 |
|     3007 | Brad Davis     | New York  | 200   |      5001 |
+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Query 5: Display the order number, order date and the purchase amount for order(s) which will be delivered by the salesman with ID 5001.

Command:

```
select order_no, order_date, purch_amt from orders1 where salesman_id = 5001;
```

Output:

```
mysql> select order_no, order_date, purch_amt from orders1 where salesman_id = 5001;
+-----+-----+-----+
| order_no | order_date | purch_amt |
+-----+-----+-----+
|    70002 | 2016-10-05 |    65.26 |
|    70005 | 2016-07-27 | 2400.60 |
|    70008 | 2016-09-10 | 5760.00 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Create table nobel_win

```
mysql> create table nobel_win( year int, subject varchar(20), winner varchar(30), country varchar(15), category varchar(15));
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> desc nobel_win;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| year  | int    | YES  |     | NULL    |       |
| subject | varchar(20) | YES  |     | NULL    |       |
| winner | varchar(30) | YES  |     | NULL    |       |
| country | varchar(15) | YES  |     | NULL    |       |
| category | varchar(15) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.04 sec)
```

Insert data into nobel_win

Command:

```
insert into nobel_win value(1994,'Literature','Kenzaburo Oe','Japan','Linguist'),  

(1994,'Economics','Reinhard Selten','Germany','Economist'),  

(1987,'Chemistry','Donald J. Cram','USA','Scientist'),  

(1987,'Chemistry','Jean-Marie Lehn','France','Scientist'),  

(1987,'Literature','Joseph Brodsky','Russia','Linguist'),  

(1987,'Economics','Robert Solow','USA','Economist'),  

(1971,'Chemistry','Gerhard Herzberg','Germany','Scientist'),  

(1971,'Literature','Pablo Neruda','Chile','Linguist'),  

(1971,'Economics','Simon Kuznets','Russia','Economist'),  

(1970,'Literature','Aleksandr Solzhenitsyn','Russia','Linguist'),  

(1970,'Chemistry','Luis Federico Leloir','France','Scientist'),  

(1970,'Economics','Paul Samuelson','USA','Economist');
```

```
mysql> insert into nobel_win value(1994,'Literature','Kenzaburo Oe','Japan','Linguist')
->
-> (1994,'Economics','Reinhard Selten','Germany','Economist'),
->
-> (1987,'Chemistry','Donald J. Cram','USA','Scientist'),
->
-> (1987,'Chemistry','Jean-Marie Lehn','France','Scientist'),
->
-> (1987,'Literature','Joseph Brodsky','Russia','Linguist'),
->
-> (1987,'Economics','Robert Solow','USA','Economist'),
->
-> (1971,'Chemistry','Gerhard Herzberg','Germany','Scientist'),
->
-> (1971,'Literature','Pablo Neruda','Chile','Linguist'),
->
-> (1971,'Economics','Simon Kuznets','Russia','Economist'),
->
-> (1970,'Literature','Aleksandr Solzhenitsyn','Russia','Linguist'),
->
-> (1970,'Chemistry','Luis Federico Leloir','France','Scientist'),
->
-> (1970,'Economics','Paul Samuelson','USA','Economist');
Query OK, 12 rows affected (0.02 sec)
Records: 12  Duplicates: 0  Warnings: 0
```

Output:

```
mysql> select * from nobel_win;
+---+---+-----+-----+-----+
| year | subject | winner | country | category |
+---+---+-----+-----+-----+
| 1994 | Literature | Kenzaburo Oe | Japan | Linguist |
| 1994 | Economics | Reinhard Selten | Germany | Economist |
| 1987 | Chemistry | Donald J. Cram | USA | Scientist |
| 1987 | Chemistry | Jean-Marie Lehn | France | Scientist |
| 1987 | Literature | Joseph Brodsky | Russia | Linguist |
| 1987 | Economics | Robert Solow | USA | Economist |
| 1971 | Chemistry | Gerhard Herzberg | Germany | Scientist |
| 1971 | Literature | Pablo Neruda | Chile | Linguist |
| 1971 | Economics | Simon Kuznets | Russia | Economist |
| 1970 | Literature | Aleksandr Solzhenitsyn | Russia | Linguist |
| 1970 | Chemistry | Luis Federico Leloir | France | Scientist |
| 1970 | Economics | Paul Samuelson | USA | Economist |
+---+---+-----+-----+-----+
12 rows in set (0.00 sec)
```

Query 6: Show the winner of the 1971 prize for Literature.**Command:**

```
select winner from nobel_win where year = 1971 AND subject =
'Literature';
```

Output:

```
mysql> select winner from nobel_win where year = 1971 AND subject = 'Literature';
+-----+
| winner |
+-----+
| Pablo Neruda |
+-----+
1 row in set (0.00 sec)
```

Query 7: Show all the details of the winners with first name Louis.**Command:**

```
select* from nobel_win where winner LIKE 'Luis%';
```

Output:

```
mysql> select* from nobel_win where winner LIKE 'Luis%';
+-----+-----+-----+-----+
| year | subject | winner | country | category |
+-----+-----+-----+-----+
| 1970 | Chemistry | Luis Federico Leloir | France | Scientist |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

Query 8: Show all the winners in Physics for 1970 together with the winner of Economics for 1971.**Command:**

```
select * from nobel_win where (subject = 'Physics' AND year = 1970)
UNION (select * from nobel_win where (subject = 'Economics' AND year = 1971));
```

Output:

```
mysql> select * from nobel_win where (subject = 'Physics' AND year = 1970) UNION (select * from nobel_win where (subject = 'Economics' AND year = 1971));
+-----+-----+-----+-----+
| year | subject | winner | country | category |
+-----+-----+-----+-----+
| 1971 | Economics | Simon Kuznets | Russia | Economist |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

Query 9: Show all the winners of Nobel prize in the year 1970 except the subject Physiology and Economics.**Command:**

```
select * from nobel_win where year = 1970 AND subject NOT IN ('Physiology','Economics');
```

Output:

```
mysql> select * from nobel_win where year = 1970 AND subject NOT IN ('Physiology','Economics');
+-----+-----+-----+-----+
| year | subject | winner | country | category |
+-----+-----+-----+-----+
| 1970 | Literature | Aleksandr Solzhenitsyn | Russia | Linguist |
| 1970 | Chemistry | Luis Federico Leloir | France | Scientist |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Query 10 • Find all the details of the Nobel winners for the subject not started with the letter 'P' and arranged the list as the most recent comes first, then by name in order.

Command:

```
select * from nobel_win where subject NOT LIKE 'P%' ORDER BY year DESC, winner;
```

Output:

```
mysql> select * from nobel_win where subject NOT LIKE 'P%' ORDER BY year DESC, winner;
+-----+-----+-----+-----+
| year | subject | winner | country | category |
+-----+-----+-----+-----+
| 1994 | Literature | Kenzaburo Oe | Japan | Linguist |
| 1994 | Economics | Reinhard Selten | Germany | Economist |
| 1987 | Chemistry | Donald J. Cram | USA | Scientist |
| 1987 | Chemistry | Jean-Marie Lehn | France | Scientist |
| 1987 | Literature | Joseph Brodsky | Russia | Linguist |
| 1987 | Economics | Robert Solow | USA | Economist |
| 1971 | Chemistry | Gerhard Herzberg | Germany | Scientist |
| 1971 | Literature | Pablo Neruda | Chile | Linguist |
| 1971 | Economics | Simon Kuznets | Russia | Economist |
| 1970 | Literature | Aleksandr Solzhenitsyn | Russia | Linguist |
| 1970 | Chemistry | Luis Federico Leloir | France | Scientist |
| 1970 | Economics | Paul Samuelson | USA | Economist |
+-----+-----+-----+-----+
12 rows in set (0.01 sec)
```

Query 11: Find the name and price of the cheapest item(s).

Query 12: Display all the customers, who are either belongs to the city New York or not had a grade above 100.

Command:

```
select * from customer1 where city = 'New York' OR NOT grade > 100;
```

Output:

```
mysql> select * from customer1 where city = 'New York' OR NOT grade > 100;
+-----+-----+-----+-----+-----+
| customer_id | customer_name | city      | grade | salesman_id |
+-----+-----+-----+-----+-----+
|     3002 | Nick Rimando | New York | 100   |      5001 |
|     3007 | Brad Davis    | New York | 200   |      5001 |
|     3009 | Geoff Camero  | Berlin    | 100   |        NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Query 13:Find those salesmen with all information who gets the commission within a range of 0.12 and 0.14.

Command:

```
select salesman_id, name, city, commission from salesman1 where commission between 0.10 AND 0.12;
```

Output:

```
mysql> select salesman_id, name, city, commission from salesman1 where commission between 0.10 AND 0.12;
+-----+-----+-----+-----+
| salesman_id | name      | city      | commission |
+-----+-----+-----+-----+
|      5003 | Lauson Hen | London    |      0.12 |
|      5005 | Pit Alex   | London    |      0.11 |
+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

Query 14: Find all those customers with all information whose names are ending with the letter 'n'.

Command:

```
select * from customer1 where customer_name LIKE '%n';
```

Output:

```
mysql> select * from customer1 where customer_name LIKE '%n';
+-----+-----+-----+-----+
| customer_id | customer_name | city | grade | salesman_id |
+-----+-----+-----+-----+
|      3001 | Brad Guzan | London | NULL |        NULL |
|      3008 | Julian Green | London |   300 |      5002 |
+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

Query 15: Find those salesmen with all information whose name containing the 1st character is 'N' and the 4th character is 'l' and rests may be any character.

Command:

```
select * from salesman1 where name LIKE 'N__l%';
```

Output:

```
mysql> select * from salesman1 where name like 'N__l%';
+-----+-----+-----+
| salesman_id | name | city | commission |
+-----+-----+-----+
|      5002 | Nail Knite | Paris |      0.13 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Query 16: Find that customer with all information who does not get any grade except NULL.

Command:

```
select * from customer1 where grade IS NULL;
```

Output:

```
mysql> select * from customer1 where grade IS NULL;
+-----+-----+-----+-----+
| customer_id | customer_name | city | grade | salesman_id |
+-----+-----+-----+-----+
|      3001 | Brad Guzan | London | NULL |        NULL |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Query 17: Find the total purchase amount of all orders.

Command:

```
select SUM(purch_amt) from orders1;
```

Output:

```
mysql> select SUM(purch_amt) from orders1;
+-----+
| SUM(purch_amt) |
+-----+
|      14345.08 |
+-----+
1 row in set (0.00 sec)
```

Query 18: Find the number of salesman currently listing for all of their customers.

Command:

```
select COUNT(salesman_id) from orders1;
```

Output:

```
mysql> select COUNT(salesman_id) from orders1;
+-----+
| COUNT(salesman_id) |
+-----+
|          7 |
+-----+
1 row in set (0.00 sec)
```

Query 19:Find the highest grade for each of the cities of the customers.

Command:

```
select city, MAX(grade) from customer1 GROUP BY city;
```

Output:

```
mysql> select city, MAX(grade) from customer1 GROUP BY city;
+-----+-----+
| city | MAX(grade) |
+-----+-----+
| London |      300 |
| New York |    200 |
| Moncow |    200 |
| Paris |    300 |
| California | 200 |
| Berlin |    100 |
+-----+-----+
6 rows in set (0.01 sec)
```

Query 20: Find the highest purchase amount ordered by the each customer with their ID and highest purchase amount.

Command:

```
select customer_id, MAX(purch_amt) from orders1 GROUP BY
customer_id;
```

Output:

```
mysql> select customer_id, MAX(purch_amt) from orders1 GROUP BY customer_id;
+-----+-----+
| customer_id | MAX(purch_amt) |
+-----+-----+
| 3001 |    270.65 |
| 3002 |  5760.00 |
| 3003 |    75.29 |
| 3004 | 1983.43 |
| 3005 |    948.50 |
| 3007 | 2400.60 |
| 3008 |    250.45 |
| 3009 | 2480.40 |
+-----+-----+
8 rows in set (0.01 sec)
```

Query 21: Find the highest purchase amount ordered by the each customer on a particular date with their ID, order date and highest purchase amount.

Command:

```
select customer_id, order_date, MAX(purch_amt) from orders1 GROUP
BY customer_id, order_date;
```

Output:

```
mysql> select customer_id, order_date, MAX(purch_amt) from orders1 GROUP BY customer_id, order_date;
+-----+-----+-----+
| customer_id | order_date | MAX(purch_amt) |
+-----+-----+-----+
| 3002 | 2016-10-05 | 65.26 |
| 3009 | 2016-10-10 | 2480.40 |
| 3009 | 2016-08-17 | 110.50 |
| 3007 | 2016-07-27 | 2400.60 |
| 3005 | 2016-09-10 | 948.50 |
| 3002 | 2016-09-10 | 5760.00 |
| 3001 | 2016-09-10 | 270.65 |
| 3004 | 2016-10-10 | 1983.43 |
| 3003 | 2016-08-17 | 75.29 |
| 3008 | 2016-06-27 | 250.45 |
+-----+-----+-----+
10 rows in set (0.00 sec)
```

Query 22: Find the highest purchase amount on a date '2012-08-17' for each salesman with their ID.

Command:

```
select salesman_id, MAX(purch_amt) from orders1 where order_date = '2012-08-17' GROUP BY salesman_id;
```

Output:

```
mysql> select salesman_id, MAX(purch_amt) from orders1 where order_date = '2012-08-17' GROUP BY salesman_id;
Empty set (0.01 sec)
```

Query 23: Find the highest purchase amount with their customer ID and order date, for only those customers who have the highest purchase amount in a day is more than 2000.

Command:

```
select customer_id, order_date, MAX(purch_amt) from orders1 GROUP BY customer_id, order_date HAVING MAX(purch_amt) > 2000.00;
```

Output:

```
mysql> select customer_id, order_date, MAX(purch_amt) from orders1 GROUP BY customer_id, order_date HAVING MAX(purch_amt) > 2000.00;
+-----+-----+-----+
| customer_id | order_date | MAX(purch_amt) |
+-----+-----+-----+
| 3009 | 2016-10-10 | 2480.40 |
| 3007 | 2016-07-27 | 2400.60 |
| 3002 | 2016-09-10 | 5760.00 |
+-----+-----+-----+
3 rows in set (0.01 sec)
```

Query 24: Write a SQL statement that counts all orders for a date August 17th, 2012.

Command:

```
select COUNT(*) from orders1 where order_date = '2016-08-17';
```

Output:

```
mysql> select COUNT(*) from orders1 where order_date = '2016-08-17';
+-----+
| COUNT(*) |
+-----+
|      2   |
+-----+
1 row in set (0.00 sec)
```



LAB-9

Aim: Use the concept of PL/SQL Control Structure.

Program to be performed:

1. Write a program to find out Minimum & Maximum Value using the concept of PL/SQLControl Structure.

Code:

```
DECLARE
  a number(5) := 8;
BEGIN
  IF( a < 20 )
    THEN
      dbms_output.put_line('a is less than 20 ');
    ELSE
      dbms_output.put_line('a is not less than 20 ');
    END IF;
  dbms_output.put_line('value of a is : ' || a);
END;
```

Output:**SQL Worksheet**

```
1 v DECLARE
2 a number(5) := 8;
3 v BEGIN
4 IF( a < 20 )
5 THEN
6 dbms_output.put_line('a is less than 20 ' );
7 v ELSE
8 dbms_output.put_line('a is not less than 20 ' );
9 END IF;
10 dbms_output.put_line('value of a is : ' || a);
11 END;
```

Statement processed.
a is less than 20
value of a is : 8

2. Write a program to find out whether a number is even or odd using if-else statement in PL/SQL Control Structure.**Code:**

```
DECLARE
a number(5) := 71;
BEGIN
IF MOD(a,2) = 0 THEN
DBMS_OUTPUT.PUT_LINE ('The number. '| |a| |' is even number');
ELSE
DBMS_OUTPUT.PUT_LINE ('The number '| |a| |' is odd number.');
```

```
END IF;  
DBMS_OUTPUT.PUT_LINE ('Done Successfully');  
END;
```

Output:**SQL Worksheet**

```
1 v DECLARE  
2 a number(5) := 71;  
3 v BEGIN  
4 IF MOD(a,2) = 0 THEN  
5 DBMS_OUTPUT.PUT_LINE ('The number. '||a||' is even number');  
6 v ELSE  
7 DBMS_OUTPUT.PUT_LINE ('The number '||a||' is odd number.');//  
8 END IF;  
9 DBMS_OUTPUT.PUT_LINE ('Done Successfully');  
10 END;
```

Statement processed.
The number 71 is odd number.
Done Successfully

3. Write a program to check given year is leap or not using the concept of PL/SQL Control Structure.**Code:**

```
DECLARE
year NUMBER := 2020;
BEGIN
IF MOD(year, 4)=0
AND
MOD(year, 100)!=0
OR
MOD(year, 400)=0 THEN
dbms_output.Put_line(year || ' is leap year ');
ELSE
dbms_output.Put_line(year || ' is not leap year.');
END IF;
END;
```

Output:**SQL Worksheet**

```
1 v DECLARE
2 year NUMBER := 2020;
3 v BEGIN
4 IF MOD(year, 4)=0
5 AND
6 MOD(year, 100)!=0
7 OR
8 MOD(year, 400)=0 THEN
9 dbms_output.Put_line(year || ' is leap year ');
10 v ELSE
11 dbms_output.Put_line(year || ' is not leap year.');
12 END IF;
13 END;
```

Statement processed.
2020 is leap year

4. Write a program to find out Grade based on the character entered using the concept of PL/SQL Control Structure.**Code:**

```
DECLARE
grade char(1) := 'A';
BEGIN
CASE grade
when 'A' then dbms_output.put_line('Excellent');
when 'B' then dbms_output.put_line('Very good');
when 'C' then dbms_output.put_line('Good');
```

```
when 'D' then dbms_output.put_line('Average');
when 'F' then dbms_output.put_line('Passed with Grace');
else dbms_output.put_line('Failed');
END CASE;
END;
```

Output:**SQL Worksheet**

```
1 v DECLARE
2   grade char(1) := 'A';
3 v BEGIN
4   CASE grade
5     when 'A' then dbms_output.put_line('Excellent');
6     when 'B' then dbms_output.put_line('Very good');
7     when 'C' then dbms_output.put_line('Good');
8     when 'D' then dbms_output.put_line('Average');
9     when 'F' then dbms_output.put_line('Passed with Grace');
10    else dbms_output.put_line('Failed');
11  END CASE;
12 END;
```

Statement processed.
Excellent

5. Write a program to find out Maximum number of the three using the concept of PL/SQL Control Structure.

Code:

```
DECLARE
    num1 NUMBER(10);
    num2 NUMBER(10);
    num3 NUMBER(10);
    max_num NUMBER(10);

BEGIN
    num1 := 35;
    num2 := 69;
    num3 := 51;
    max_num := num1;

    IF num2 > max_num THEN
        max_num := num2;
    END IF;

    IF num3 > max_num THEN
        max_num := num3;
    END IF;

    DBMS_OUTPUT.PUT_LINE('The maximum number is: ' || max_num);
END;
```

Output:**SQL Worksheet**

```
4 num3 NUMBER(10);
5 max_num NUMBER(10);
6 v BEGIN
7 num1 := 35;
8 num2 := 69;
9 num3 := 51;
10 max_num := num1;
11 v IF num2 > max_num THEN
12 max_num := num2;
13 END IF;
14 v IF num3 > max_num THEN
15 max_num := num3;
16 END IF;
17 DBMS_OUTPUT.PUT_LINE('The maximum number is: ' || max_num);
18 END;
```

Statement processed.
The maximum number is: 69



LAB-10

Aim: Write a program to Print Number using PL/SQL Loop.

Program to be performed:

1. Write a program to Print 1 to 10 Number using PL/SQL Loop.

Code:

```
DECLARE
  i NUMBER := 1;
BEGIN
  LOOP
    EXIT WHEN i>10;
    DBMS_OUTPUT.PUT_LINE(i);
    i := i+1;
  END LOOP;
END;
```

Output:**SQL Worksheet**

```
1 v DECLARE
2 i NUMBER := 1;
3 v BEGIN
4 LOOP
5 EXIT WHEN i>10;
6 DBMS_OUTPUT.PUT_LINE(i);
7 i := i+1;
8 END LOOP;
9 END;
```

10

statement processed.

1
2
3
4
5
6
7
8
9
10

2. Write a program to find multiplication of two number using PL/SQL Loop.**Code:**

```
DECLARE
    num1 NUMBER := 5;
    num2 NUMBER := 3;
    result NUMBER := 0;
BEGIN
    FOR i IN 1..num2 LOOP
        result := result + num1;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Multiplication of ' || num1 || ' and ' || num2
    || ' is ' || result);
END;
```

Output:**SQL Worksheet**

```
1 v DECLARE
2 num1 NUMBER := 5;
3 num2 NUMBER := 3;
4 result NUMBER := 0;
5 v BEGIN
6 FOR i IN 1..num2 LOOP
7 result := result + num1;
8 END LOOP;
9 DBMS_OUTPUT.PUT_LINE('Multiplication of ' || num1 || ' and ' || num2 || ' is ' || result);
10 END;
11
```

Statement processed.
Multiplication of 5 and 3 is 15

3. Write a Program to print table for the given number using while loop in PL/SQL.**Code:**

```
DECLARE
I NUMBER:=1;
N NUMBER;
BEGIN
N:=10;
WHILE(I<=10)
LOOP
DBMS_OUTPUT.PUT_LINE( N || '*' || I || '=' || N*I);
I:=I+1;
END LOOP;
END;
```

Output:**SQL Worksheet**

```
1 v DECLARE
2 I NUMBER:=1;
3 N NUMBER;
4 v BEGIN
5 N:=10;
6 v WHILE(I<=10)
7 LOOP
8 DBMS_OUTPUT.PUT_LINE( N || ' * ' || I || ' = ' || N*I);
9 I:=I+1;
10 END LOOP;
11 END;
```

Statement processed.

```
10 * 1 = 10
10 * 2 = 20
10 * 3 = 30
10 * 4 = 40
10 * 5 = 50
10 * 6 = 60
10 * 7 = 70
10 * 8 = 80
10 * 9 = 90
10 * 10 = 100
```

4. Write a Program to print factorial of given number using for loop in PL/SQL.**Code:**

```
DECLARE
  N NUMBER;
  FAC NUMBER:=1;
  I NUMBER;
BEGIN
  N:=5;
  FOR I IN 1..N
  LOOP
    FAC:=FAC*I;
  END LOOP;
  DBMS_OUTPUT.PUT_LINE('FACTORIAL=' || FAC);
END;
```

Output:**SQL Worksheet**

```
1 v DECLARE
2 N NUMBER;
3 FAC NUMBER:=1;
4 I NUMBER;
5 v BEGIN
6 N:=5;
7 v FOR I IN 1..N
8 LOOP
9 FAC:=FAC*I;
10 END LOOP;
11 DBMS_OUTPUT.PUT_LINE('FACTORIAL=' || FAC);
12 END;
```

Statement processed.
FACTORIAL=120

5. Write a program to print reverse number using PL/SQL for Loop.**Code:**

```
DECLARE
VAR1 NUMBER;
BEGIN
VAR1:=10;
FOR VAR2 IN REVERSE 1..5
LOOP
DBMS_OUTPUT.PUT_LINE (VAR1*VAR2);
END LOOP;
```

END;

Output:

SQL Worksheet

```
1 v DECLARE
2   VAR1 NUMBER;
3 v BEGIN
4   VAR1:=10;
5 v FOR VAR2 IN REVERSE 1..5
6   LOOP
7   DBMS_OUTPUT.PUT_LINE (VAR1*VAR2);
8 END LOOP;
9 END;
```

Statement processed.

50
40
30
20
10



LAB-11

Aim: To study TCL Commands.

The following commands are used to control transactions.

- **COMMIT** – to save the changes.
- **ROLLBACK** – to roll back the changes.
- **SAVEPOINT** – creates points within the groups of transactions in which to
- **ROLLBACK.**

Create table Employee: -

Code:

- create database practical7;
- use practical7;
- create table Employee(EMP_ID int,EMP_NAME varchar(30),EMP_LOC varchar(30));
- insert into Employee values(1000,'Amit','Pune'),
-> (2000,'Raju','mumbai'),
-> (3000,'Ram','Chennai');
- select * from Employee;

Output:

```
mysql> create database practical7;
Query OK, 1 row affected (0.06 sec)

mysql> use practical7;
Database changed
mysql> create table Employee(EMP_ID int,EMP_NAME varchar (30),EMP_LOC varchar(30));
Query OK, 0 rows affected (0.11 sec)

mysql> insert into Employee values(1000,'Amit','Pune'),
-> (2000,'Raju','mumbai'),
-> (3000,'Ram','Chennai');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> select * from Employee;
+-----+-----+-----+
| EMP_ID | EMP_NAME | EMP_LOC |
+-----+-----+-----+
| 1000  | Amit      | Pune    |
| 2000  | Raju      | mumbai  |
| 3000  | Ram       | Chennai |
+-----+-----+-----+
3 rows in set (0.01 sec)
```

Queries to be performed:**1. update the EMP_ LOC for Amit.**

Query: update Employee set EMP_LOC = 'Rajkot' where EMP_NAME = 'Amit';Commit;

Output:

```
mysql> update Employee set EMP_LOC = 'Rajkot' where EMP_NAME = 'Amit';Commit;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from Employee;
+-----+-----+-----+
| EMP_ID | EMP_NAME | EMP_LOC |
+-----+-----+-----+
| 1000  | Amit     | Rajkot   |
| 2000  | Raju     | mumbai   |
| 3000  | Ram      | Chennai  |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

2. Let us consider that we have updated EMP_LOC for Raju to Bangalore later and realize that the update was done mistakenly as below. Then we can restore the EMP_LOC for 'Raju' to Hyderabad again by using the Rollback command as below.

Query:

- update Employee set EMP_LOC = 'Bangalore' where EMP_NAME = 'Raju';Rollback;

- update Employee set EMP_LOC = 'Hyderabad' where EMP_NAME = 'Raju';Rollback;

Output:

```

mysql> update Employee set EMP_LOC = 'Bangalore' where EMP_NAME = 'Raju';Rollback;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

Query OK, 0 rows affected (0.00 sec)

mysql> select * from Employee;
+-----+-----+-----+
| EMP_ID | EMP_NAME | EMP_LOC   |
+-----+-----+-----+
| 1000  | Amit     | Rajkot    |
| 2000  | Raju     | Bangalore |
| 3000  | Ram      | Chennai   |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> update Employee set EMP_LOC = 'Hyderabad' where EMP_NAME = 'Raju';Rollback;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

Query OK, 0 rows affected (0.00 sec)

mysql> select * from Employee;
+-----+-----+-----+
| EMP_ID | EMP_NAME | EMP_LOC   |
+-----+-----+-----+
| 1000  | Amit     | Rajkot    |
| 2000  | Raju     | Hyderabad |
| 3000  | Ram      | Chennai   |
+-----+-----+-----+
3 rows in set (0.00 sec)

```

3. insert the values to the Employee table and perform the updates using savepoint.**Query:**

- update Employee set EMP_NAME = 'Harsh' where EMP_ID = 4000;savepoint;
- select * from Employee;

Output:

```

mysql> update Employee set EMP_NAME = 'Harsh' where EMP_ID = 4000;savepoint;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

```

```
mysql> select * from Employee;
+-----+-----+-----+
| EMP_ID | EMP_NAME | EMP_LOC |
+-----+-----+-----+
| 1000   | Amit     | Rajkot  |
| 2000   | Raju     | Hyderabad |
| 3000   | Ram      | Chennai |
| 4000   | Harsh    | Vadodra  |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

4. update the EMP_ LOC for Ram.

Query:

- update Employee set EMP_LOC = 'Pune' where EMP_NAME = 'Ram';Commit;
- select * from Employee;

Output:

```
mysql> update Employee set EMP_LOC = 'Pune' where EMP_NAME = 'Ram';Commit;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

Query OK, 0 rows affected (0.00 sec)

mysql> select * from Employee;
+-----+-----+-----+
| EMP_ID | EMP_NAME | EMP_LOC |
+-----+-----+-----+
| 1000   | Amit     | Rajkot  |
| 2000   | Raju     | Hyderabad |
| 3000   | Ram      | Pune    |
| 4000   | Harsh    | Vadodra  |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

5. Update the EMP_NAME Ram to Raman

Query:

```
- update Employee set EMP_NAME = 'Raman' where EMP_NAME = 'Raj';  
- select * from Employee;
```

Output:

```
mysql> update Employee set EMP_NAME = 'Raman' where EMP_NAME = 'Raj';  
Query OK, 0 rows affected (0.00 sec)  
Rows matched: 0 Changed: 0 Warnings: 0  
  
mysql> select * from Employee;  
+-----+-----+-----+  
| EMP_ID | EMP_NAME | EMP_LOC |  
+-----+-----+-----+  
| 1000 | Amit | Rajkot |  
| 2000 | Raju | Hyderabad |  
| 3000 | Ram | Pune |  
| 4000 | Harsh | Vadodra |  
+-----+-----+-----+  
4 rows in set (0.00 sec)
```



LAB-12

Aim: Create Connection Between Programming Language and Database.

Steps:-

Step-1: Install the MySQL Connector/Python:

- Open your terminal or command prompt and run the following command to install the mysql-connector-python package using pip:

```
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kaush>pip install mysql-connector-python
Requirement already satisfied: mysql-connector-python in c:\user-
-packages (8.2.0)
Requirement already satisfied: protobuf<=4.21.12,>=4.21.1 in c:\
site-packages (from mysql-connector-python) (4.21.12)
```

Step-2: Set up a MySQL Database:

- Make sure you have a MySQL database installed and running. If not, you can download and install MySQL from the official website.

Step-3: Create a Database and Table:

- Using a MySQL (like MySQL Workbench), create a database and a table inside it.
- For example:

```
CREATE DATABASE mydatabase;
USE mydatabase;

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50),
    password VARCHAR(50)
);
```

Step-4: Establish the Connection:

- Create a Python script (e.g., connect_mysql.py) and import the mysql.connector module:

```
Python
```

```
import mysql.connector
```

Step-5: Write Python Code:

- Now, you can write Python code to connect to your MySQL database.
- **host**: The hostname or IP address of your MySQL server
- **user**: Your MySQL username with access to the database.
- **password**: Your MySQL password for the specified user.
- **database**: The name of the MySQL database you want to connect to.

```
import mysql.connector

# Connect to the database
db = mysql.connector.connect(
    host="localhost",
    user="kaushal01",
    password="root",
    database="mydatabase"
)
```

Step-6: Execute Queries:

- You can execute various SQL queries using the execute() method of the cursor object.
- `cursor = con.cursor()`
- `cursor.execute("SELECT * FROM my_table")`

Step-7: Handle Exceptions:

- Always wrap your database operations in try-except blocks to handle any potential errors gracefully.

Step-8: Commit Changes:

- Now, to save changes in python you have to write:
- `con.commit()`

Step 9: Close Connection:

- Once you are finished working with the database, you need to close the connection.
- For that you have to write:
- `con.close()`

