

ASSIGNMENT MYSQL

OPERATIONS AND

KEYWORDS

Nimra Ashraf

110829

Submitted to:

Mam Sehrish Khan

05/05/2025

CREATE DATABASE, TABLES & INSERTING VALUES

Here are the syntaxes of mostly keys used here and before:

- ⇒ **create database database_name;**
- ⇒ **use database_name;**
- ⇒ **create table table_name(attribute_name datatype(size),....);**
- ⇒ **INSERT INTO table_name values('values...')**
- ⇒ **ALTER table table_name ADD FOREIGN KEY (name) REFERENCES table_name (P.K);**

```
MySQL 8.0 Command Line Client - Unicode
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.41 MySQL Community Server - GPL

Copyright (c) 2000, 2025, 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> create database company;
Query OK, 1 row affected (0.01 sec)

mysql> use company;
Database changed
mysql> create table Dept(Dept_no INT(5) PRIMARY KEY, Dept_name varchar(25));
Query OK, 0 rows affected, 1 warning (0.05 sec)

mysql> DESCRIBE Dept;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Dept_no | int | NO | PRI | NULL |  |
| Dept_name | varchar(25) | YES |  | NULL |  |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)

mysql> INSERT INTO Dept values('101','marketing');
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO Dept values('102','FIR');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Dept values('103','finance');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO Dept values('104','employee_union');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from Dept;
```

Dept_no	Dept_name
101	marketing
102	FIR
103	finance
104	employee_union

4 rows in set (0.00 sec)

```
mysql> create table Grade(grade varchar(5) PRIMARY KEY,low_salary INT(10),high_salary INT(10));
Query OK, 0 rows affected, 2 warnings (0.05 sec)
```

```
mysql> DESCRIBE Grade;
```

Field	Type	Null	Key	Default	Extra
grade	varchar(5)	NO	PRI	NULL	
low_salary	int	YES		NULL	
high_salary	int	YES		NULL	

3 rows in set (0.00 sec)

```
mysql> INSERT INTO Grade values('1','10000','17000');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO Grade values('2','12000','19000');
```

MySQL 8.0 Command Line Client - Unicode

```
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO Grade values('3','14000','20000');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO Grade values('4','15000','22000');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from Grade;
```

grade	low_salary	high_salary
1	10000	17000
2	12000	19000
3	14000	20000
4	15000	22000

4 rows in set (0.00 sec)

```
mysql> create table Employee(em_no INT(5) PRIMARY KEY,em_name varchar(20),job varchar(15),hire_date DATE,salary INT(10),commission INT(10),Dept_no INT(5),grade varchar(5));
Query OK, 0 rows affected, 4 warnings (0.05 sec)
```

```
mysql> DESCRIBE Employee;
```

Field	Type	Null	Key	Default	Extra
em_no	int	NO	PRI	NULL	
em_name	varchar(20)	YES		NULL	
job	varchar(15)	YES		NULL	
hire_date	date	YES		NULL	
salary	int	YES		NULL	
commission	int	YES		NULL	
Dept_no	int	YES		NULL	
grade	varchar(5)	YES		NULL	

8 rows in set (0.00 sec)

```
mysql> ALTER table Employee ADD FOREIGN KEY (Dept_no) REFERENCES Dept (Dept_no);
Query OK, 0 rows affected (0.13 sec)
```

```
mysql> ALTER table Employee ADD FOREIGN KEY (grade) REFERENCES Grade (grade);
Query OK, 0 rows affected (0.12 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DESCRIBE Employee;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| em_no | int | NO | PRI | NULL | |
| em_name | varchar(20) | YES | | NULL | |
| job | varchar(15) | YES | | NULL | |
| hire_date | date | YES | | NULL | |
| salary | int | YES | | NULL | |
| commission | int | YES | | NULL | |
| Dept_no | int | YES | MUL | NULL | |
| grade | varchar(5) | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> INSERT INTO Employee values('1234','Ali','marketing','2025-02-03','19000','5000','101','1');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Employee values('2341','Ahmad','HR','2025-03-06','30000','8000','102','2');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Employee values('3456','Arham','assistant','2025-01-09','15000','3000','103','3');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Employee values('6438','Rehman','manager','2025-10-11','50000','9000','104','4');
Query OK, 1 row affected (0.01 sec)

mysql> select * from Employee;
+-----+-----+-----+-----+-----+-----+-----+
| em_no | em_name | job | hire_date | salary | commission | Dept_no | grade |
+-----+-----+-----+-----+-----+-----+-----+
| 1234 | Ali | marketing | 2025-02-03 | 19000 | 5000 | 101 | 1 |
| 2341 | Ahmad | HR | 2025-03-06 | 30000 | 8000 | 102 | 2 |
| 3456 | Arham | assistant | 2025-01-09 | 15000 | 3000 | 103 | 3 |
| 6438 | Rehman | manager | 2025-10-11 | 50000 | 9000 | 104 | 4 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

SELECTING ONE COLUMN FROM TABLE

⇒ To select only one column from a table use the syntax follows:

Select attribute1_name from table_name where attribute2_name='...' ;

```
MySQL 8.0 Command Line Client - Unicode

mysql> INSERT INTO Employee values('6438','Rehman','manager','2025-10-11','50000','9000','104','4');
Query OK, 1 row affected (0.01 sec)

mysql> select * from Employee;
+-----+-----+-----+-----+-----+-----+-----+
| em_no | em_name | job | hire_date | salary | commission | Dept_no | grade |
+-----+-----+-----+-----+-----+-----+-----+
| 1234 | Ali | marketing | 2025-02-03 | 19000 | 5000 | 101 | 1 |
| 2341 | Ahmad | HR | 2025-03-06 | 30000 | 8000 | 102 | 2 |
| 3456 | Arham | assistant | 2025-01-09 | 15000 | 3000 | 103 | 3 |
| 6438 | Rehman | manager | 2025-10-11 | 50000 | 9000 | 104 | 4 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> UPDATE Employee set hire_date='2024-01-04' where em_no='2341';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from Employee;
+-----+-----+-----+-----+-----+-----+-----+
| em_no | em_name | job | hire_date | salary | commission | Dept_no | grade |
+-----+-----+-----+-----+-----+-----+-----+
| 1234 | Ali | marketing | 2025-02-03 | 19000 | 5000 | 101 | 1 |
| 2341 | Ahmad | HR | 2024-01-04 | 30000 | 8000 | 102 | 2 |
| 3456 | Arham | assistant | 2025-01-09 | 15000 | 3000 | 103 | 3 |
| 6438 | Rehman | manager | 2025-10-11 | 50000 | 9000 | 104 | 4 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select job from Employee where em_no='1234'
->
+-----+
| job |
+-----+
| marketing |
+-----+
1 row in set (0.00 sec)

mysql>
```

PERFORM ARITHMETIC & RELATIONAL OPERATIONS

To perform the arithmetic operations (+,-,*,/) and relational operations (>,<=,!=,<,>=) use the following syntaxes:

⇒ **select attribute_name + value AS annual from table_name;**

USE OF AS

Here **AS annual** keyword is used for performing operations on all values of attribute.

USE OF WHERE CLAUSE

We can use this syntax for each operator. The **where clause** is used to point that attribute or table on which we are working.

```
mysql> select salary+200 AS annual from Employee;
+-----+
| annual |
+-----+
| 19200  |
| 30200  |
| 15200  |
| 50200  |
+-----+
4 rows in set (0.00 sec)

mysql> select salary+200 AS annual from Employee where em_no='1234';
+-----+
| annual |
+-----+
| 19200  |
+-----+
1 row in set (0.00 sec)

mysql> select salary-200 AS annual from Employee where em_no='1234';
+-----+
| annual |
+-----+
| 18800  |
+-----+
1 row in set (0.00 sec)

mysql> select salary+250*12 AS annual from Employee;
+-----+
| annual |
+-----+
| 22000  |
| 33000  |
| 18000  |
| 53000  |
+-----+
4 rows in set (0.00 sec)

mysql>
```

```
MySQL 8.0 Command Line Client - Unicode
mysql> select salary*10 from Employee;
+-----+
| salary*10 |
+-----+
| 190000    |
| 300000    |
| 150000    |
| 500000    |
+-----+
4 rows in set (0.00 sec)

mysql> select em_name from Employee where salary>19000;
+-----+
| em_name |
+-----+
| Ahmad   |
| Rehman  |
+-----+
2 rows in set (0.00 sec)

mysql> select em_name from Employee where salary<19000;
+-----+
| em_name |
+-----+
| Arham   |
+-----+
1 row in set (0.00 sec)

mysql> select em_name from Employee where salary=19000;
+-----+
| em_name |
+-----+
| Ali     |
+-----+
1 row in set (0.00 sec)

mysql>
```

USE OF ORDER BY FOR SORTING

ORDER BY keyword is used to sort the values of tables in ascending or descending order. By default ascending order is set. Otherwise for sorting the records in descending order DESC order is used. For ascending **ASC** and descending **DESC** is used.

Syntaxes of above two orders are:

- ⇒ **select * from table_name ORDER BY attribute_name
DESC;**
- ⇒ **select * from table_name ORDER BY attribute_name
DESC;**

```
mysql> ORDER BY low_salary DESC;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'ORDER BY low_salary DESC' at line 1
mysql> select * from Grade ORDER BY low_salary DESC;
+-----+-----+-----+
| grade | low_salary | high_salary |
+-----+-----+-----+
| 4     | 15000     | 22000     |
| 3     | 14000     | 20000     |
| 2     | 12000     | 19000     |
| 1     | 10000     | 17000     |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from Grade ORDER BY low_salary ASC;
+-----+-----+-----+
| grade | low_salary | high_salary |
+-----+-----+-----+
| 1     | 10000     | 17000     |
| 2     | 12000     | 19000     |
| 3     | 14000     | 20000     |
| 4     | 15000     | 22000     |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select en_name from Employee ORDER BY salary;
+-----+
| en_name |
+-----+
| Arham   |
| Ali     |
| Ahmad   |
| Rehman  |
+-----+
4 rows in set (0.00 sec)
```

Activate Windows
Go to Settings to activate Windows.

USE OF DINTINCT KEYWORD

Distinct keyword is used to retrieve unique values from a specified column or set of columns. It eliminates duplicate records.

```
mysql> select DISTINCT commission from Employee;
+-----+
| commission |
+-----+
| 5000       |
| 8000       |
| 3000       |
| 9000       |
+-----+
4 rows in set (0.01 sec)

mysql>
```

thanks