May 20, 2025

CC-215 DATABASE SYSTEM

FAIQA LATIF (110790)
TOOBA FATIMA (110789)

Instructor: Miss Sehrish khan

Database Design and Implementation in MySQL

- Entity Relationship Diagram
- Relational schema\table schema
- Normalization
- Implementation in MySQL

ENTITY RELATIONSHIP DIAGRAM

ERD stands for Entity-Relationship Diagram. It's a visual representation of the structure of a database, showing the relationships between entities (tables) and their attributes.

An ERD typically includes:

1. **Entities (tables):** Representing real-world objects or concepts.

2. **Attributes (columns):** Describing characteristics of entities.

3. **Relationships:** Showing connections between entities, such as one-to-one, one-to-many, or many-to-many.

Steps

Steps used to develop Entity Relationship Diagram for statement problem are given.

➢ Identify Entities

➢ Identify Attributes

➢ Identify Relationships

➢ Developing Diagram

ENTITY:

An entity is a distinct, identifiable object, concept, or thing (e.g., a person, place, or item) with defined attributes. It exists independently and can be uniquely identified within a system or context.

Representation:

An entity is represented by using a rectangle shape. Name of entity is written inside the rectangle and is written in capital letters.

ATTRIBUTE:

An attribute is a characteristic or property that describes an entity (e.g., name, age, colour). It holds specific data values that define the entity's state or features.

Representation:

An attribute is represented by using oval shape, name of attribute is written inside the shape. It is important to note that the primary key attribute is underlined.

Multi-Valued attribute is represented by double lined oval shape, and it is also important to note that the derived attribute is represented by dotted lined oval shape.

RELATIONSHIP:

A relationship describes how two or more entities interact or are connected in a system. It defines the association, such as "owns," "belongs to," or "works for," between them.

Representation:

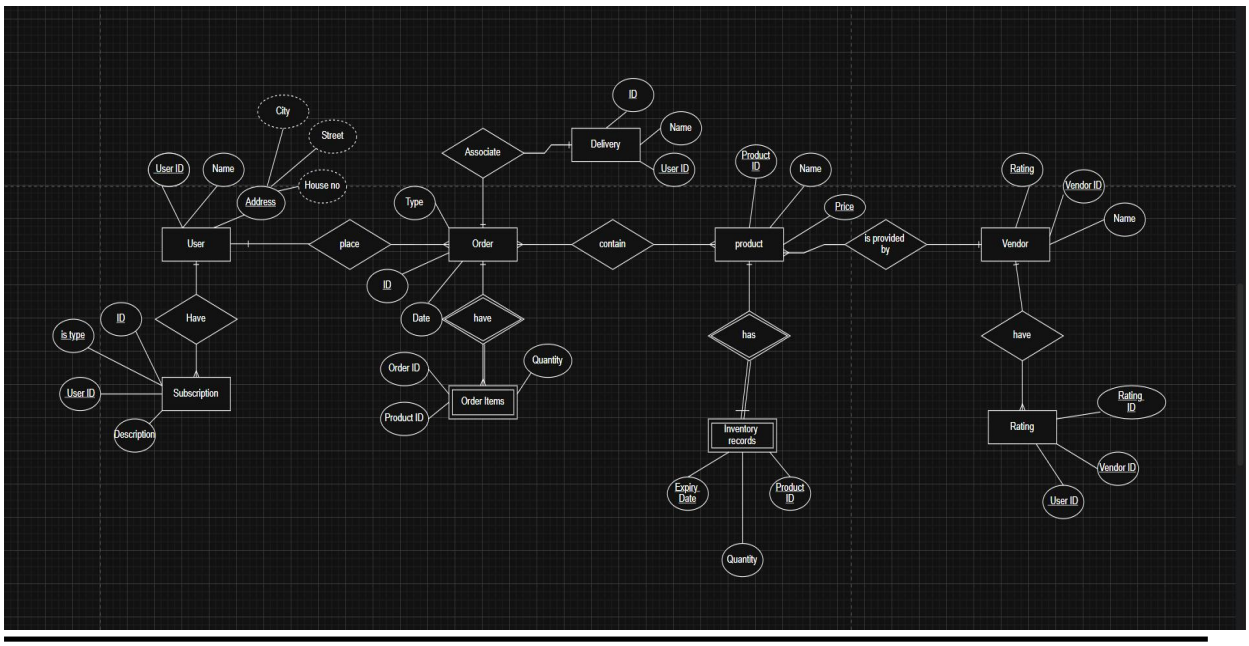
Relationship is represented by using diamond shape. A weak relationship is represented by doublelined rectangle. In the given diagram. Weak relationship is used to relate a weak entity with a strong entity.

Problem Statement

Grocery Delivery App with Vendor Reviews and Weekly Subscriptions

A grocery delivery system enables users to either place instant grocery orders or subscribe to curated weekly bundles (like "Healthy Heart Kit"). Products come from multiple verified vendors. The system tracks vendor reliability and quality through customer ratings. Users can customize subscriptions, postpone deliveries, or swap out items. The inventory system predicts restocking needs and manages perishable item expiration alerts.

ENTITY RELATIONSHIP DIAGRAM USING CHEN MODEL



NORMALIZED TABLES

User (User_ID (PK), Name, Address)

Vendor (Vendor _ID (PK), Name)

Product (Product _ID (PK), Name, Price)

Product Vendor (Product ID (FK), Vendor _ID (FK))

Order (Order _ID (PK), Order _Date, Order _type)

Order _item (Order _item ID (PK) Order _ID (FK)

Subscription (Subscription _ID (PK), User_ID (FK), Bundle _ID (FK), Frequency, Status)

Inventory (inv _ID, Product _ID (FK), Quantity, Expiration _Date)

Delivery (Delivery _ID (PK), User _ID (FK), Delivery _name)

Rating (Rating _ID (PK), Vendor _ID (FK))

NORMALIZATION

Normalization is a process of organizing data in a database to minimize redundancy and improve data integrity.

The **main objective** of database normalization is to eliminate redundant data, minimize data modification errors, and simplify the query process.

- **Every table has primary key.**
- **Other keys depend upon primary key.**
- **Field must contain atomic values.**

FIRST NORMAL FORM (1NF):

Each table cell contains a single value. No repeating groups or arrays in a single column. 1NF eliminates data redundancy and improves data integrity by organizing data into well-structured tables. In 1NF there must be a primary key.

Each attribute contains only atomic (indivisible) values, and there are no repeating groups of columns.

SECOND NORMAL FORM (2NF):

Second Normal Form (2NF) is a level of database normalization that builds on First Normal Form (1NF). A table is in 2NF if:

There is no partial dependency of any column on a composite primary key.

All non-key attributes depend on the entire primary key.

All non-key attributes are fully functionally dependent on the entire primary key in above schema.

THIRD NORMAL FORM (3NF):

Third Normal Form (3NF) in database normalization eliminates transitive dependencies, ensuring that each non-key attribute in a table depends only on the primary key, not on other non-key attributes. It builds upon First Normal Form (1NF) and Second Normal Form (2NF).

3NF eliminates indirect dependencies, ensuring that each non-key attribute depends directly on the primary key.

COMMON COMMANDS:

1. To see existing databases:

```
show databases;
```

2. To Create a database:

```
CREATE DATABASE database _name;
```

3. To use the database:

```
use database _name;
```

4. To see existing TABLES:

```
Show tables;
```

5. To Create a tables:

Create table table _name (attribute_id_1 type (domain) primary key, attribute_2 type(domain),…,n);

6. To describe table:

Describe table _name;

7. To see values from table:

Select * from table _name;

8. To add foreign key in existing table:

Alter table table _name ADD constraint fk _table _name foreign key (attribute) references ref _table (ref _attribute);

IMPLEMENTATION IN MYSQL

- Show databases;

- **Create database** grocery;
- Use grocery;
- Create table **user**(user_id INT(3) PRIMARY KEY,user_name varchar (15),user_address varchar(23) NOT NULL);

```
mysql> create database grocery;
Query OK, 1 row affected (0.02 sec)

mysql> create table user(user_id INT(7) PRIMARY KEY,user_name varchar(17),user_address varchar(21) NOT NULL);
ERROR 1046 (3D000): No database selected
mysql> USE GROCERY;
Database changed
mysql>  create table user(user_id INT(7) PRIMARY KEY,user_name varchar(17),user_address varchar(21) NOT NULL);
Query OK, 0 rows affected, 1 warning (0.04 sec)

mysql> DESCRIBE user;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| user_id      | int         | NO   | PRI | NULL    |       |
| user_name    | varchar(17) | YES  |     | NULL    |       |
| user_address | varchar(21) | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

- **Insert into** user values;
- Select * from user;

```
mysql> insert into user values('1','afifa','skp'),('2','ali','skp'),('3','faiqa','skp'),('4','hina','lhr');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from user;
+---------+-----------+--------------+
| user_id | user_name | user_address |
+---------+-----------+--------------+
|       1 | afifa     | skp          |
|       2 | ali       | skp          |
|       3 | faiqa     | skp          |
|       4 | hina      | lhr          |
+---------+-----------+--------------+
4 rows in set (0.00 sec)
```

- Create table **orderr** (order _id INT(8) PRIMARY KEY, order _date DATE, order _type varchar(25) NOT NULL) ;
- **Insert into** orderr values;
- Select * from user;

```
mysql> INSERT INTO orderr values('5','2023-9-14','online'),('6','2023-8-5','in-store'),('7','2023-6-2','food delivery'),('8','2023-3-6','phone order');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> Select * from orderr;
+----------+------------+---------------+
| order_id | order_date | order_type    |
+----------+------------+---------------+
|        5 | 2023-09-14 | online        |
|        6 | 2023-08-05 | in-store      |
|        7 | 2023-06-02 | food delivery |
|        8 | 2023-03-06 | phone order   |
+----------+------------+---------------+
4 rows in set (0.00 sec)
```

- Create table **product**(prod _id INT(7) PRIMARY KEY, prod _name varchar(25),prod _price INT(25) NOT NULL);
- **Insert into** product values;
- Select * from product;

```
mysql> select * from product;
+---------+---------------+------------+
| prod_id | prod_name     | prod_price |
+---------+---------------+------------+
|      20 | apple watch   |     100000 |
|      21 | nikee shoes   |      25000 |
|      22 | samsung mobile |      70000 |
|      23 | iphone 13     |     150000 |
+---------+---------------+------------+
4 rows in set (0.00 sec)
```

- Create table **orderitem**(orderitem _id INT (8) PRIMARY KEY);
- Add FOREIGN KEY(order _id) in orderitem table REFRENCES orderr(order _id);
- **Describe** orderitem;

```
Records: 0  Duplicates: 0  Warnings: 0

mysql> describe orderitem;
+--------------+------+------+-----+---------+-------+
| Field        | Type | Null | Key | Default | Extra |
+--------------+------+------+-----+---------+-------+
| orderitem_id | int  | NO   | PRI | NULL    |       |
| order_id     | int  | YES  | MUL | NULL    |       |
+--------------+------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

- **Insert into** orderitem values;
- Select * from orderitem;

```
mysql> insert into orderitem values('41','5'),('42','6'),('43','7'),('44','8');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from orderitem;
+--------------+----------+
| orderitem_id | order_id |
+--------------+----------+
|           41 |        5 |
|           42 |        6 |
|           43 |        7 |
|           44 |        8 |
+--------------+----------+
4 rows in set (0.00 sec)
```

- Create table **subscription**(sub _id INT(8) PRIMARY KEY, sub _description varchar(27) NOT NULL);
- **Alter table** subscription **add** user _id INT(7);
- **Alter table** subscription **Add FOREIGN KEY**(user _id) in subscription table REFRENCES user(user _id);
- Select * from subscription;
- **DESCRIBE** subscription;

```
mysql> create table subscription(sub_id INT(8) PRIMARY KEY,sub_description varchar(27) NOT NULL);
Query OK, 0 rows affected, 1 warning (0.03 sec)

mysql> alter table subscription add user_id INT(7);
Query OK, 0 rows affected, 1 warning (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 1

mysql> alter table subscription add constraint fk_gi FOREIGN KEY(user_id) REFERENCES user(user_id);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0

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

mysql> describe subscription;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| sub_id          | int         | NO   | PRI | NULL    |       |
| sub_description | varchar(27) | NO   |     | NULL    |       |
| user_id         | int         | YES  | MUL | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

- **Insert into** subscription values;
- Select * from subscription;

```
mysql> insert into subscription values('51','monthly premium plan','1'),('52','yearly basic plan','2'),('53','weakly trial plan','3'),('54','life time membership','4');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from subscription;
+--------+---------------------+---------+
| sub_id | sub_description     | user_id |
+--------+---------------------+---------+
|     51 | monthly premium plan |       1 |
|     52 | yearly basic plan    |       2 |
|     53 | weakly trial plan    |       3 |
|     54 | life time membership |       4 |
+--------+---------------------+---------+
```

- Create table **delivery**(del_id INT(9) PRIMARY KEY,del_name varchar(16) NOT NULL);
- **Alter table** delivery **add** user_id INT(7);
- **Alter table** delivery **Add FOREIGN KEY**(user _id) in delivery table REFRENCES user(user _id);
- **Describe** delivery;

```
mysql> create table delivery(del_id INT(9) PRIMARY KEY,del_name varchar(10) NOT NULL);
Query OK, 0 rows affected, 1 warning (0.03 sec)

mysql> alter table delivery add user_id INT(7);
Query OK, 0 rows affected, 1 warning (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 1

mysql> ALTER TABLE delivery add constraint fk_gh FOREIGN KEY(user_id) REFERENCES user(user_id);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE delivery;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| del_id   | int         | NO   | PRI | NULL    |       |
| del_name | varchar(10) | NO   |     | NULL    |       |
| user_id  | int         | YES  | MUL | NULL    |       |
+----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

- **Insert into** delivery values;

- Select * from delivery;

```
mysql> select * from delivery;
+--------+----------+---------+
| del_id | del_name | user_id |
+--------+----------+---------+
|     61 | shipping |       1 |
|     62 | express  |       2 |
|     63 | same day |       3 |
|     64 | next day |       4 |
+--------+----------+---------+
4 rows in set (0.00 sec)
```

- **Create table** inventoryrecord;
- **Add FOREIGN KEY** (prod _id) REFRENCES product(prod _id);
- **Describe** inventoryrecord;

```
mysql> create table inventoryrecord(inv_id INT(8) PRIMARY KEY,inv_quantity INT(8),inv_expirydate DATE NOT NULL);
Query OK, 0 rows affected, 2 warnings (0.03 sec)

mysql> ALTER TABLE inventoryrecord add prod_id INT(7);
Query OK, 0 rows affected, 1 warning (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 1

mysql> ALTER TABLE inventoryrecord add constraint fk_gr FOREIGN KEY(prod_id) REFERENCES product(prod_id);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> describe inventoryrecord;
+----------------+------+------+-----+---------+-------+
| Field          | Type | Null | Key | Default | Extra |
+----------------+------+------+-----+---------+-------+
| inv_id         | int  | NO   | PRI | NULL    |       |
| inv_quantity   | int  | YES  |     | NULL    |       |
| inv_expirydate | date | NO   |     | NULL    |       |
| prod_id        | int  | YES  | MUL | NULL    |       |
+----------------+------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

- **Insert into** inventoryrecord values;
- Select * from inventoryrecord;

```
mysql> insert into inventoryrecord values('81','3','2023-5-9','20'),('82','7','2023-5-1','21'),('83','3','2023-5-3','22'),('84','4','2023-5-9','23');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from inventoryrecord;
+--------+--------------+----------------+---------+
| inv_id | inv_quantity | inv_expirydate | prod_id |
+--------+--------------+----------------+---------+
|     81 |            3 | 2023-05-09     |      20 |
|     82 |            7 | 2023-05-01     |      21 |
|     83 |            3 | 2023-05-03     |      22 |
|     84 |            4 | 2023-05-09     |      23 |
+--------+--------------+----------------+---------+
4 rows in set (0.00 sec)
```

- Create table **product**(prod _id INT(7) PRIMARY KEY, prod _name varchar(25),prod _price INT(25) NOT NULL);
- **Insert into** product values;
- Select * from product;

```
mysql> select * from product;
+---------+----------------+------------+
| prod_id | prod_name      | prod_price |
+---------+----------------+------------+
|      20 | apple watch    |     100000 |
|      21 | nikee shoes    |      25000 |
|      22 | samsung mobile |      70000 |
|      23 | iphone 13      |     150000 |
+---------+----------------+------------+
4 rows in set (0.00 sec)
```

- Create table **rating**(rating _id INT(6) PRIMARY KEY);
- **Alter table add FOREIGN KEY** vend _id REFRENCES vendor(vend _id);
- **Describe** rating;

```
mysql> describe rating;
+------------+------+------+-----+---------+-------+
| Field      | Type | Null | Key | Default | Extra |
+------------+------+------+-----+---------+-------+
| rating_id | int  | NO   | PRI | NULL    |       |
| vend_id   | int  | YES  | MUL | NULL    |       |
+------------+------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

- **Insert into** rating values;
- Select * from rating;

```
mysql> insert into rating values('91','31'),('92','32'),('93','33'),('94','34');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from rating;
+------------+----------+
| rating_id | vend_id |
+------------+----------+
|         91 |       31 |
|         92 |       32 |
|         93 |       33 |
|         94 |       34 |
+------------+----------+
4 rows in set (0.00 sec)
```

- Create table productvendor;
- Describe productvendor;

```
mysql> create table product_vendor(prod_id INT,vend_id INT,PRIMARY KEY(prod_id,vend_id),FOREIGN KEY(prod_id) REFERENCES product(prod_id),FOREIGN KEY(vend_id) REFERENCES vendor(vend_id));
Query OK, 0 rows affected (0.07 sec)

mysql> describe product_vendor;
+---------+------+------+-----+---------+-------+
| Field   | Type | Null | Key | Default | Extra |
+---------+------+------+-----+---------+-------+
| prod_id | int  | NO   | PRI | NULL    |       |
| vend_id | int  | NO   | PRI | NULL    |       |
+---------+------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

- Insert into productvendor values;

- Select * from productvendor;

```
mysql> INSERT into product_vendor values('20','31'),('21','32'),('22','33'),('23','34');
Query OK, 4 rows affected (0.02 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from product_vendor;
+---------+---------+
| prod_id | vend_id |
+---------+---------+
|      20 |      31 |
|      21 |      32 |
|      22 |      33 |
|      23 |      34 |
+---------+---------+
4 rows in set (0.00 sec)
```

SOME OTHER QUERIES:

- Show tables;

```
mysql> show tables;
+--------------------+
| Tables_in_grocery  |
+--------------------+
| bundle             |
| bundleproduct      |
| delivery           |
| inventoryrecord    |
| orderitem          |
| orderr             |
| product            |
| rating             |
| subscription       |
| user               |
| vendor             |
+--------------------+
11 rows in set (0.00 sec)

mysql>
```

- Select **SUM, MIN, AVG** product price;

```
mysql> select MIN(prod_price) from product;
+-----------------+
| MIN(prod_price) |
+-----------------+
|           25000 |
+-----------------+
1 row in set (0.00 sec)

mysql> select AVG(prod_price) from product;
+-----------------+
| AVG(prod_price) |
+-----------------+
|      86250.0000 |
+-----------------+
1 row in set (0.00 sec)

mysql> select SUM(prod_price) from product;
+-----------------+
| SUM(prod_price) |
+-----------------+
|          345000 |
+-----------------+
1 row in set (0.00 sec)
```

- Select inv_quantity from inventory record **as scale**;

```
mysql> select inv_quantity from inventoryrecord as scale;
+--------------+
| inv_quantity |
+--------------+
|            3 |
|            7 |
|            3 |
|            4 |
+--------------+
4 rows in set (0.00 sec)
```

- Select **distinctive user** name from users;

```
mysql> select distinct user_name from user;
+-----------+
| user_name |
+-----------+
| afifa     |
| ali       |
| faiqa     |
| hina      |
+-----------+
4 rows in set (0.00 sec)
```

- **Increment** in product price;

```
mysql> select prod_price+500 as incremental_price from product;
+-------------------+
| incremental_price |
+-------------------+
|            100500 |
|             25500 |
|             70500 |
|            150500 |
+-------------------+
4 rows in set (0.00 sec)
```

- **Decrement** in product price;

```
mysql> select prod_price-200 as decremental_price from product;
+-------------------+
| decremental_price |
+-------------------+
|             99800 |
|             24800 |
|             69800 |
|            149800 |
+-------------------+
4 rows in set (0.00 sec)
```

- Select **price** of product **where** product **price>25000**;

```
mysql> select prod_price from product where prod_price>25000;
+------------+
| prod_price |
+------------+
|     100000 |
|      70000 |
|     150000 |
+------------+
3 rows in set (0.00 sec)
```

- Select **name** of product **where** product **price>25000**;

```
mysql> select prod_name from product where prod_price>25000;
+----------------+
| prod_name      |
+----------------+
| apple watch    |
| samsung mobile |
| iphone 13      |
+----------------+
3 rows in set (0.00 sec)
```

- Select **maximum** product price;

```
mysql> select MAX(prod_price) from product;
+-----------------+
| MAX(prod_price) |
+-----------------+
|          150000 |
+-----------------+
1 row in set (0.00 sec)
```

- Use of **logical operator;**

```
mysql> select prod_name from product where prod_name='iphone 13' && prod_price<200000 && prod_price>600;
+-----------+
| prod_name |
+-----------+
| iphone 13 |
+-----------+
1 row in set, 2 warnings (0.00 sec)
```

- Select product name by using **IN** command;

```
mysql> select prod_name from product where prod_price IN(25000,70000,150000);
+----------------+
| prod_name      |
+----------------+
| nikee shoes    |
| samsung mobile |
| iphone 13      |
+----------------+
3 rows in set (0.00 sec)
```

- Select name where **first letter** is **a;**

```
mysql> select user_name from user where user_name like 'a%';
+-----------+
| user_name |
+-----------+
| afifa     |
| faiqa     |
| hina      |
+-----------+
3 rows in set (0.00 sec)
```

- Select name where **second letter** is **a;**

```
mysql> select user_name from user where user_name like '_a%';
+-----------+
| user_name |
+-----------+
| faiqa     |
+-----------+
1 row in set (0.00 sec)
```

- Select name where **first and last letter** is **a;**

```
mysql> select user_name from user where user_name like 'a%a';
+-----------+
| user_name |
+-----------+
| afifa     |
+-----------+
1 row in set (0.00 sec)
```

- Select user name where user address **IS NOT NULL**;

```
mysql> select user_name from user where user_address is not null;
+-----------+
| user_name |
+-----------+
| afifa     |
| ali       |
| faiqa     |
| hina      |
+-----------+
4 rows in set (0.00 sec)
```

- Select **count(*);**

```
mysql> select count(*) from vendor;
+----------+
| count(*) |
+----------+
|        4 |
+----------+
1 row in set (0.01 sec)
```

- Select **count()**;

```
mysql> select count(user_name) from user;
+-----------------+
| count(user_name) |
+-----------------+
|               4 |
+-----------------+
1 row in set (0.00 sec)

mysql> select count(vend_id) from vendor;
+----------------+
| count(vend_id) |
+----------------+
|              4 |
+----------------+
1 row in set (0.00 sec)
```

- Select **AVG** price by using **GROUP BY** command;

```
mysql> select AVG(prod_price) from product group by prod_name;
+-----------------+
| AVG(prod_price) |
+-----------------+
|     100000.0000 |
|      25000.0000 |
|      70000.0000 |
|     150000.0000 |
+-----------------+
4 rows in set (0.00 sec)
```

- Select **MAX, MIN** price by using **GROUP BY** command;

```
mysql> select MAX(prod_price) from product group by prod_name;
+-----------------+
| MAX(prod_price) |
+-----------------+
|          100000 |
|           25000 |
|           70000 |
|          150000 |
+-----------------+
4 rows in set (0.00 sec)

mysql> select MIN(prod_price) from product group by prod_id;
+-----------------+
| MIN(prod_price) |
+-----------------+
|          100000 |
|           25000 |
|           70000 |
|          150000 |
+-----------------+
4 rows in set (0.00 sec)
```

- Select * from product;

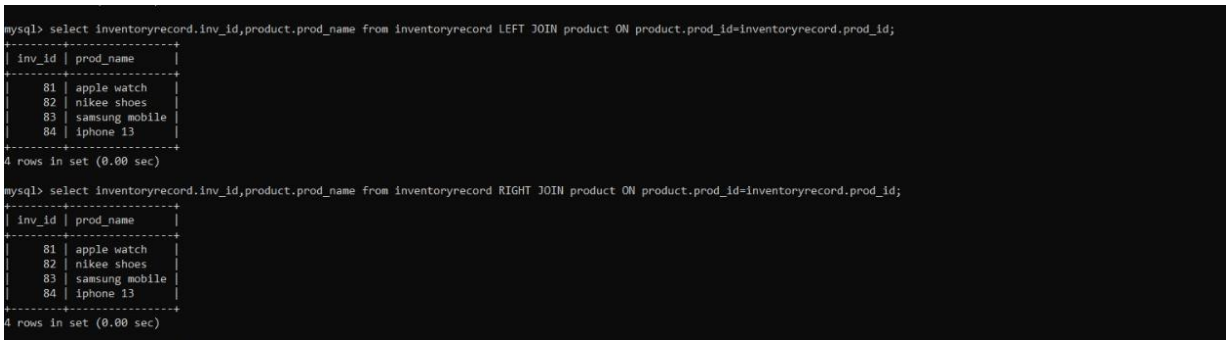
```
mysql> select * from product;
+----------+-----------------+-------------+
| prod_id | prod_name       | prod_price |
+----------+-----------------+-------------+
|       20 | apple watch     |      100000 |
|       21 | nikee shoes     |       25000 |
|       22 | samsung mobile |       70000 |
|       23 | iphone 13       |      150000 |
+----------+-----------------+-------------+
4 rows in set (0.00 sec)
```

Inner join;

```
mysql> select inventoryrecord.inv_id,product.prod_name from inventoryrecord INNER JOIN product ON product.prod_id=inventoryrecord.prod_id;
+--------+---------------+
| inv_id | prod_name     |
+--------+---------------+
|     81 | apple watch   |
|     82 | nikee shoes   |
|     83 | samsung mobile |
|     84 | iphone 13     |
+--------+---------------+
4 rows in set (0.02 sec)

mysql> select * from inventoryrecord INNER JOIN product ON product.prod_id=inventoryrecord.prod_id;
+--------+--------------+---------------+---------+---------+---------------+------------+
| inv_id | inv_quantity | inv_expirydate | prod_id | prod_id | prod_name     | prod_price |
+--------+--------------+---------------+---------+---------+---------------+------------+
|     81 |            3 | 2023-05-09    |      20 |      20 | apple watch   |     100000 |
|     82 |            7 | 2023-05-01    |      21 |      21 | nikee shoes   |      25000 |
|     83 |            3 | 2023-05-03    |      22 |      22 | samsung mobile |      70000 |
|     84 |            4 | 2023-05-09    |      23 |      23 | iphone 13     |     150000 |
+--------+--------------+---------------+---------+---------+---------------+------------+
4 rows in set (0.00 sec)

mysql> select inventoryrecord.inv_id,product.prod_name from inventoryrecord INNER JOIN product ON product.prod_id>inventoryrecord.prod_id;
+--------+---------------+
| inv_id | prod_name     |
+--------+---------------+
|     81 | nikee shoes   |
|     81 | samsung mobile |
|     81 | iphone 13     |
|     82 | samsung mobile |
|     82 | iphone 13     |
|     83 | iphone 13     |
+--------+---------------+
6 rows in set (0.00 sec)
```

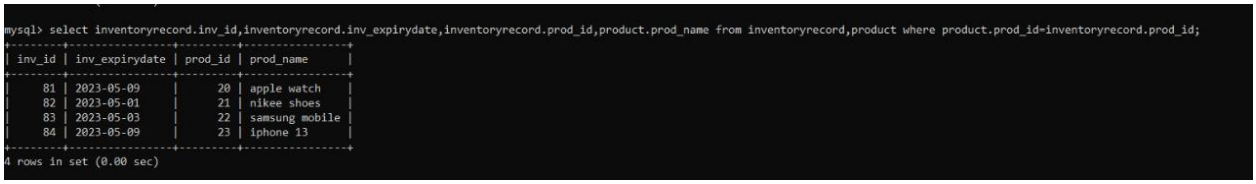
```
mysql> select inventoryrecord.inv_id,product.prod_name from inventoryrecord INNER JOIN product ON product.prod_id<inventoryrecord.prod_id;
+--------+---------------+
| inv_id | prod_name     |
+--------+---------------+
|     82 | apple watch   |
|     83 | apple watch   |
|     83 | nikee shoes   |
|     84 | apple watch   |
|     84 | nikee shoes   |
|     84 | samsung mobile |
+--------+---------------+
6 rows in set (0.00 sec)

mysql> select DISTINCT * from inventoryrecord INNER JOIN product ON product.prod_id=inventoryrecord.prod_id;
+--------+--------------+---------------+---------+---------+---------------+------------+
| inv_id | inv_quantity | inv_expirydate | prod_id | prod_id | prod_name     | prod_price |
+--------+--------------+---------------+---------+---------+---------------+------------+
|     81 |            3 | 2023-05-09    |      20 |      20 | apple watch   |     100000 |
|     82 |            7 | 2023-05-01    |      21 |      21 | nikee shoes   |      25000 |
|     83 |            3 | 2023-05-03    |      22 |      22 | samsung mobile |      70000 |
|     84 |            4 | 2023-05-09    |      23 |      23 | iphone 13     |     150000 |
+--------+--------------+---------------+---------+---------+---------------+------------+
4 rows in set (0.00 sec)
```

LEFT join and RIGHT join;



```
mysql> select inventoryrecord.inv_id,product.prod_name from inventoryrecord LEFT JOIN product ON product.prod_id=inventoryrecord.prod_id;
+--------+---------------+
| inv_id | prod_name     |
+--------+---------------+
|     81 | apple watch   |
|     82 | nikee shoes   |
|     83 | samsung mobile |
|     84 | iphone 13     |
+--------+---------------+
4 rows in set (0.00 sec)

mysql> select inventoryrecord.inv_id,product.prod_name from inventoryrecord RIGHT JOIN product ON product.prod_id=inventoryrecord.prod_id;
+--------+---------------+
| inv_id | prod_name     |
+--------+---------------+
|     81 | apple watch   |
|     82 | nikee shoes   |
|     83 | samsung mobile |
|     84 | iphone 13     |
+--------+---------------+
4 rows in set (0.00 sec)
```

Equi join;

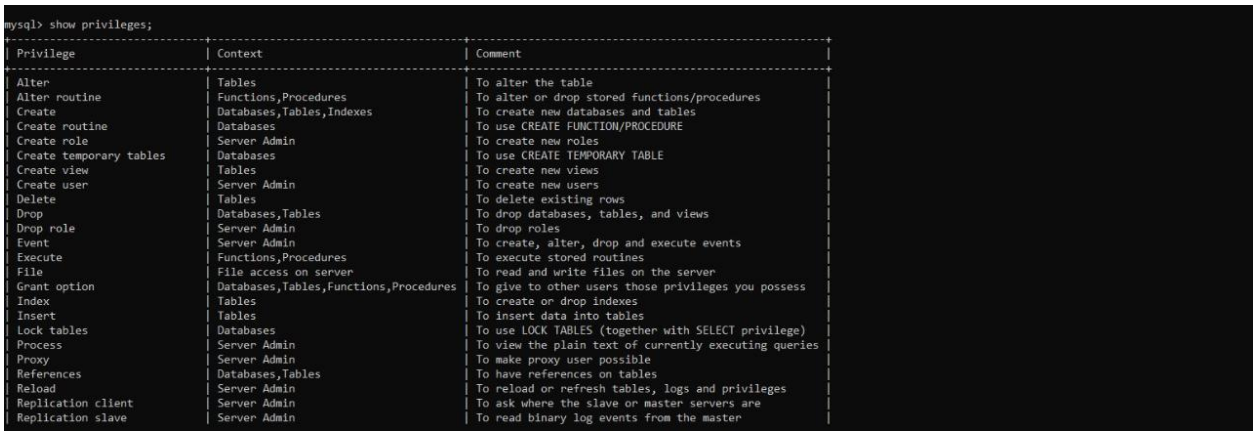


```
mysql> select inventoryrecord.inv_id,inventoryrecord.inv_expirydate,inventoryrecord.prod_id,product.prod_name from inventoryrecord,product where product.prod_id=inventoryrecord.prod_id;
+--------+---------------+---------+---------------+
| inv_id | inv_expirydate | prod_id | prod_name     |
+--------+---------------+---------+---------------+
|     81 | 2023-05-09    |      20 | apple watch   |
|     82 | 2023-05-01    |      21 | nikee shoes   |
|     83 | 2023-05-03    |      22 | samsung mobile |
|     84 | 2023-05-09    |      23 | iphone 13     |
+--------+---------------+---------+---------------+
4 rows in set (0.00 sec)
```

View:

```
mysql> create or replace view inventoryrecord_view AS select inv_expirydate,inv_id from inventoryrecord;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from inventoryrecord_view;
+---------------+--------+
| inv_expirydate | inv_id |
+---------------+--------+
| 2023-05-09    |     81 |
| 2023-05-01    |     82 |
| 2023-05-03    |     83 |
| 2023-05-09    |     84 |
+---------------+--------+
4 rows in set (0.01 sec)
```

Show privileges;



```
mysql> show privileges;
+---------------------------+---------------------------------------+-------------------------------------------------------+
| Privilege                 | Context                               | Comment                                               |
+---------------------------+---------------------------------------+-------------------------------------------------------+
| Alter                     | Tables                                | To alter the table                                    |
| Alter routine             | Functions,Procedures                  | To alter or drop stored functions/procedures          |
| Create                    | Databases,Tables,Indexes              | To create new databases and tables                    |
| Create routine            | Databases                             | To use CREATE FUNCTION/PROCEDURE                      |
| Create role               | Server Admin                          | To create new roles                                   |
| Create temporary tables   | Databases                             | To use CREATE TEMPORARY TABLE                         |
| Create view               | Tables                                | To create new views                                   |
| Create user               | Server Admin                          | To create new users                                   |
| Delete                    | Tables                                | To delete existing rows                               |
| Drop                      | Databases,Tables                      | To drop databases, tables, and views                  |
| Drop role                 | Server Admin                          | To drop roles                                         |
| Event                     | Server Admin                          | To create, alter, drop and execute events             |
| Execute                   | Functions,Procedures                  | To execute stored routines                            |
| File                      | File access on server                 | To read and write files on the server                 |
| Grant option              | Databases,Tables,Functions,Procedures | To give to other users those privileges you possess   |
| Index                     | Tables                                | To create or drop indexes                             |
| Insert                    | Tables                                | To insert data into tables                            |
| Lock tables               | Databases                             | To use LOCK TABLES (together with SELECT privilege)   |
| Process                   | Server Admin                          | To view the plain text of currently executing queries |
| Proxy                     | Server Admin                          | To make proxy user possible                           |
| References                | Databases,Tables                      | To have references on tables                          |
| Reload                    | Server Admin                          | To reload or refresh tables, logs and privileges      |
| Replication client        | Server Admin                          | To ask where the slave or master servers are          |
| Replication slave         | Server Admin                          | To read binary log events from the master             |
```

```
| References                  | Databases,Tables                  | To have references on tables                                |
| Reload                      | Server Admin                      | To reload or refresh tables, logs and privileges            |
| Replication client          | Server Admin                      | To ask where the slave or master servers are                |
| Replication slave           | Server Admin                      | To read binary log events from the master                   |
| Select                      | Tables                            | To retrieve rows from table                                 |
| Show databases              | Server Admin                      | To see all databases with SHOW DATABASES                    |
| Show view                   | Tables                            | To see views with SHOW CREATE VIEW                          |
| Shutdown                    | Server Admin                      | To shut down the server                                     |
| Super                       | Server Admin                      | To use KILL thread, SET GLOBAL, CHANGE MASTER, etc.         |
| Trigger                     | Tables                            | To use triggers                                             |
| Create tablespace           | Server Admin                      | To create/alter/drop tablespaces                            |
| Update                      | Tables                            | To update existing rows                                     |
| Usage                       | Server Admin                      | No privileges - allow connect only                          |
| ENCRYPTION_KEY_ADMIN        | Server Admin                      |                                                             |
| INNODB_REDO_LOG_ARCHIVE     | Server Admin                      |                                                             |
| RESOURCE_GROUP_USER         | Server Admin                      |                                                             |
| FIREWALL_EXEMPT             | Server Admin                      |                                                             |
| SET_USER_ID                 | Server Admin                      |                                                             |
| SERVICE_CONNECTION_ADMIN    | Server Admin                      |                                                             |
| GROUP_REPLICATION_ADMIN     | Server Admin                      |                                                             |
| AUDIT_ABORT_EXEMPT          | Server Admin                      |                                                             |
| GROUP_REPLICATION_STREAM    | Server Admin                      |                                                             |
| CLONE_ADMIN                 | Server Admin                      |                                                             |
| SYSTEM_USER                 | Server Admin                      |                                                             |
| AUTHENTICATION_POLICY_ADMIN | Server Admin                      |                                                             |
| SHOW_ROUTINE                | Server Admin                      |                                                             |
| BACKUP_ADMIN                | Server Admin                      |                                                             |
| CONNECTION_ADMIN            | Server Admin                      |                                                             |
| PERSIST_RO_VARIABLES_ADMIN  | Server Admin                      |                                                             |
| RESOURCE_GROUP_ADMIN        | Server Admin                      |                                                             |
| SESSION_VARIABLES_ADMIN     | Server Admin                      |                                                             |
| SYSTEM_VARIABLES_ADMIN      | Server Admin                      |                                                             |
| APPLICATION_PASSWORD_ADMIN  | Server Admin                      |                                                             |
| FLUSH_OPTIMIZER_COSTS       | Server Admin                      |                                                             |
| AUDIT_ADMIN                 | Server Admin                      |                                                             |
| BINLOG_ADMIN                | Server Admin                      |                                                             |
| BINLOG_ENCRYPTION_ADMIN     | Server Admin                      |                                                             |
| FLUSH_STATUS                | Server Admin                      |                                                             |
| FLUSH_TABLES                | Server Admin                      |                                                             |
| FLUSH_USER_RESOURCES        | Server Admin                      |                                                             |
| REPLICATION_APPLIER         | Server Admin                      |                                                             |
| INNODB_REDO_LOG_ENABLE      | Server Admin                      |                                                             |
| XA_RECOVER_ADMIN            | Server Admin                      |                                                             |
| PASSWORDLESS_USER_ADMIN     | Server Admin                      |                                                             |
| TABLE_ENCRYPTION_ADMIN      | Server Admin                      |                                                             |
| ROLE_ADMIN                  | Server Admin                      |                                                             |
| REPLICATION_SLAVE_ADMIN     | Server Admin                      |                                                             |
| SENSITIVE_VARIABLES_OBSERVER| Server Admin                      |                                                             |
| TELEMETRY_LOG_ADMIN         | Server Admin                      |                                                             |
+-----------------------------+-----------------------------------+-------------------------------------------------------------+
69 rows in set (0.01 sec)
```

Drop user;

```
mysql> describe user;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| user_id      | int         | NO   | PRI | NULL    |       |
| user_name    | varchar(25) | NO   |     | NULL    |       |
| user_address | varchar(25) | NO   |     | NULL    |       |
| ph_no        | int         | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> drop table user;
Query OK, 0 rows affected (0.03 sec)

mysql> select * from user;
ERROR 1146 (42S02): Table 'grocery.user' doesn't exist
mysql>
```

File Explorer

THE END!