**SQL**

1. Structure Query Language.
2. This language is used to interact with the Database.
3. SQL is a case insensitive language.
4. This is the Query language where you can execute a single query at a time.
5. You can also execute multiple queries at a time using procedure, functions, trigger which is a part of PL/SQL.
6. There are Different categories of SQL Query
   1. DDL
      1. Data Definition Language
      2. In this categories queries executes to work with the schema and structures such as table, DB, constrains etc.
      3. Queries: **Create, alter, drop, truncate**.
   2. DML
      1. Data Manipulation Language
      2. Here, you will work the Data from the table. The Data operations such as inserting data, modify data, delete data will be done in this type.
      3. Queries: **insert, update, Delete**
   3. DCL
      1. Data Control Language
      2. Is used to control on the data. You can manage the access on the operations perform by different user.
      3. Queries: **Grant, Revoke**
   4. TCL
      1. Transaction Control Language
      2. Is used to manage a database transaction. One transaction can be a combination of multiple queries.
      3. Queries: **commit, rollback, savepoint**
   5. DQL
      1. Data Query Language
      2. This is used to retrieve the data from the database.
      3. Queries: **Select**

**MySql Data Type:**

* + - 1. String Data Type
      2. Numeric Data Type
      3. Date, time Data Type

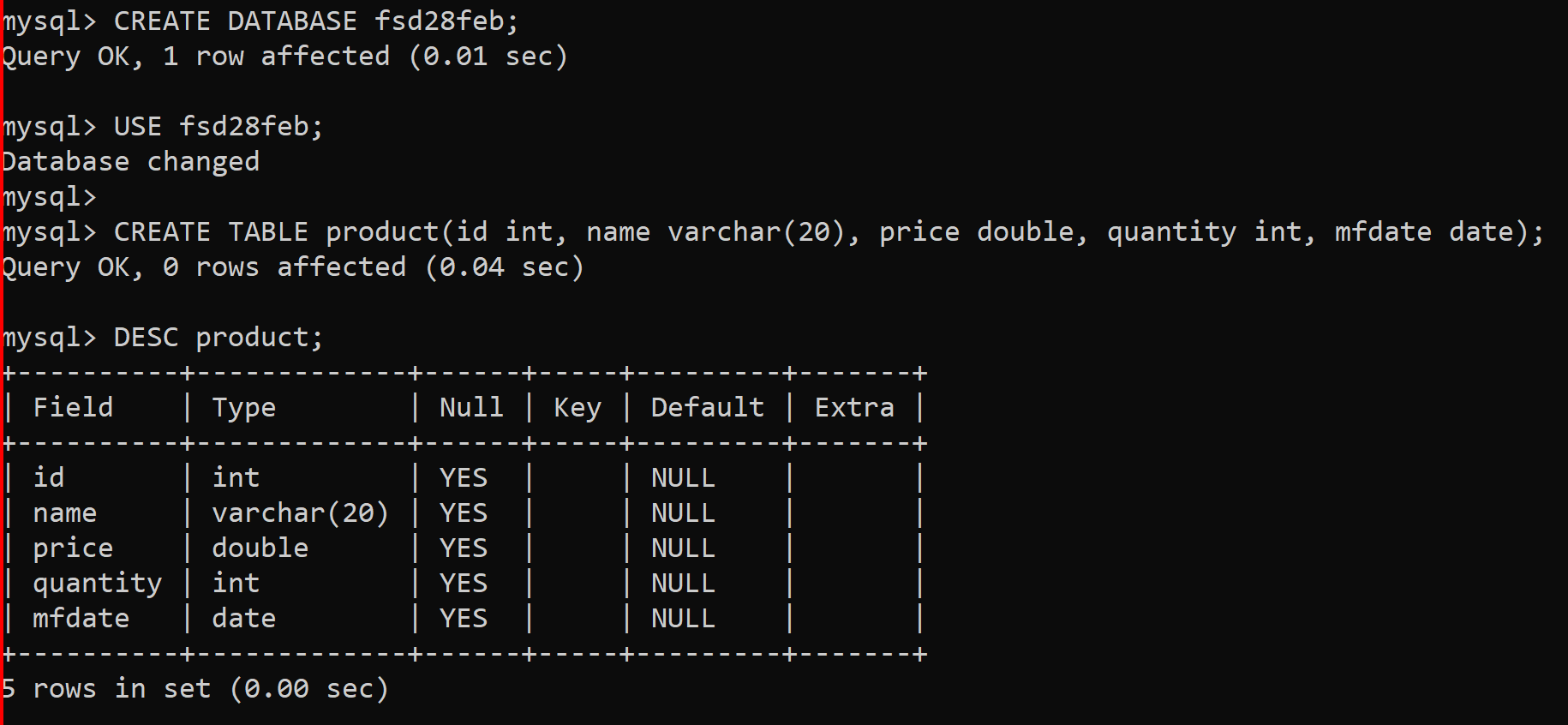
[**https://www.w3schools.com/sql/sql\_datatypes.asp**](https://www.w3schools.com/sql/sql_datatypes.asp)

**Data Definition Language (DDL)**

1. Create a data base Object (Database, table, function, procedure, trigger etc.)
   1. To create a data base object you can use create query
   2. Syntax:

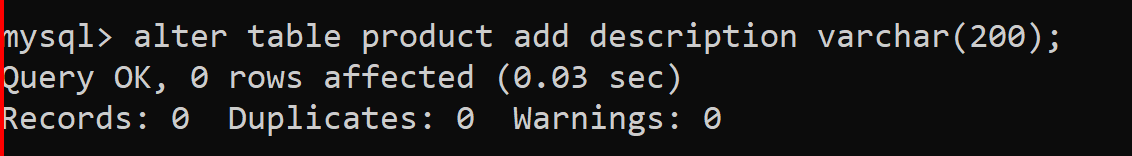
CREATE <Object\_Type> <Object\_Name> ;

CREATE DATABASE <database\_name> : Create a database.

CREATE TABLE <table\_name> (columns datatype,.,.,.); 

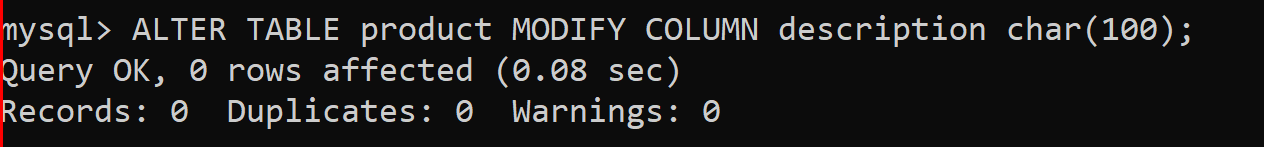
1. Alter the column
   1. Is use to add, remove, modify, rename the column.
      1. Add Column Syntax

ALTER TABLE <table\_name> ADD <column\_name> <datatype>;



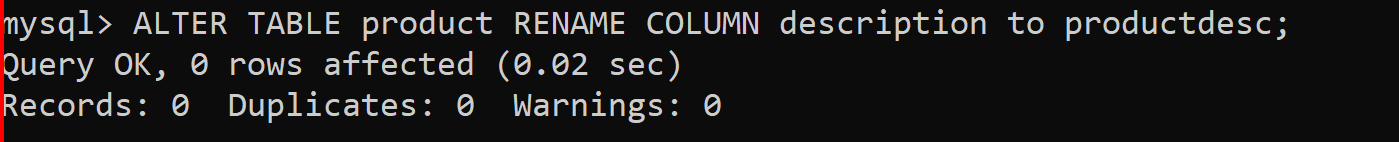
* + 1. Modify Column Syntax

ALTER TABLE <table\_name> MODIFY COLUMN <column\_name> <datatype>;



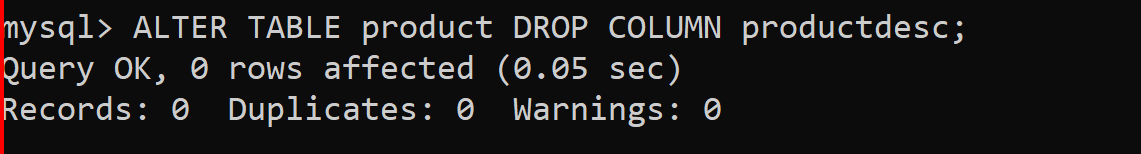
* + 1. Rename Column Syntax

ALTER TABLE <table\_name> RENAME COLUMN <old\_column\_name > TO <new\_column\_name>;



* + 1. Drop/Delete column syntac

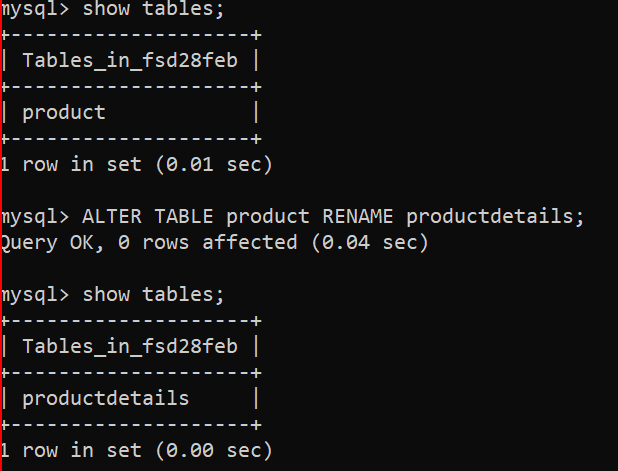
ALTER TABLE <table\_name> DROP COLUMN < column\_name>;



* + 1. Add Constraints in to existing column.

**ALTER TABLE employee ADD CONSTRAINT salary\_chk CHECK (salary BETWEEN 10000 AND 100000);**

* + 1. Rename Table

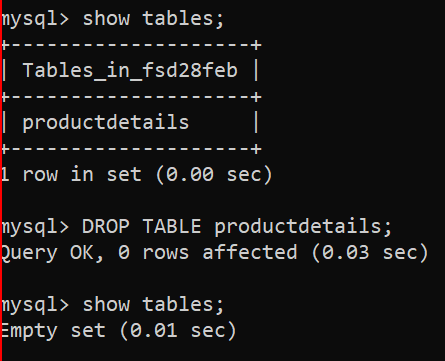


1. Drop Database Objects

You can delete the database objects like database, table, view, procedure, trigger, function

Syntax:

DROP <OBJECT\_NAME> <name>;



**View**

View is an Object of the database which can be use to get data faster instead of executing query repetitively.

**Syntax:**

CREATE VIEW view\_name AS  
SELECT column1, column2, ...  
FROM table\_name  
WHERE condition;

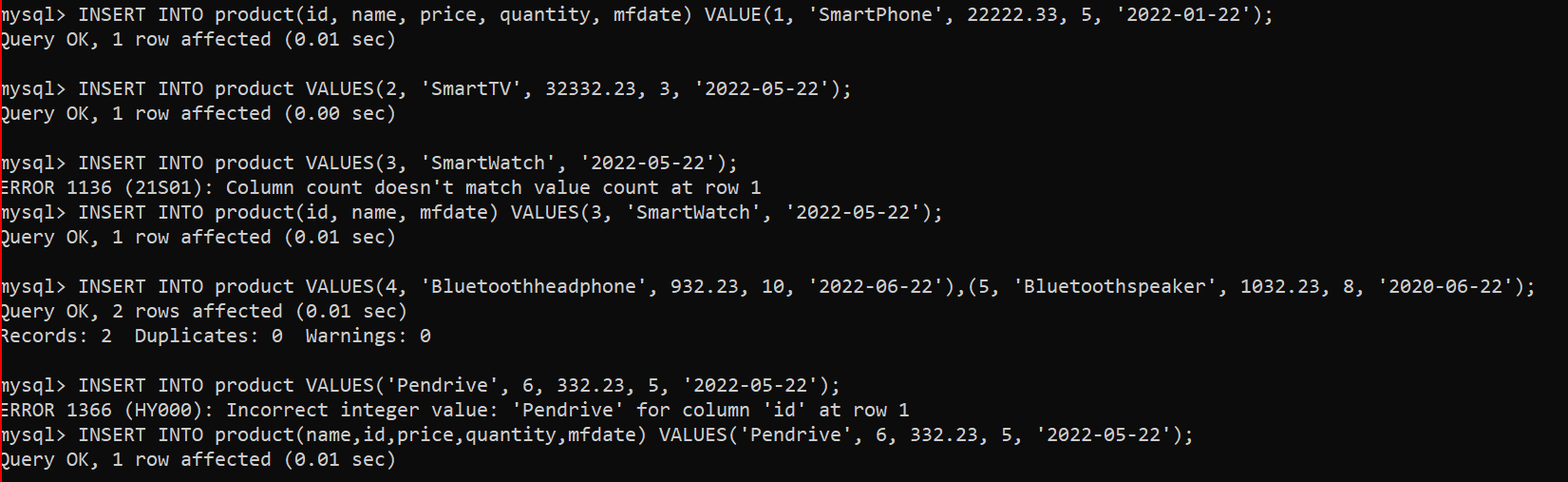
**Data Manipulation Language (DML)**

1. These types of queries are used to perform the Data related operations. Such as creating new data, updating existing data, deleting the existing data etc.
   1. Insert Data

Is use to insert new records into table.

Syntax:

INSERT INTO <TABLE\_NAME>(columns) VALUES(values);



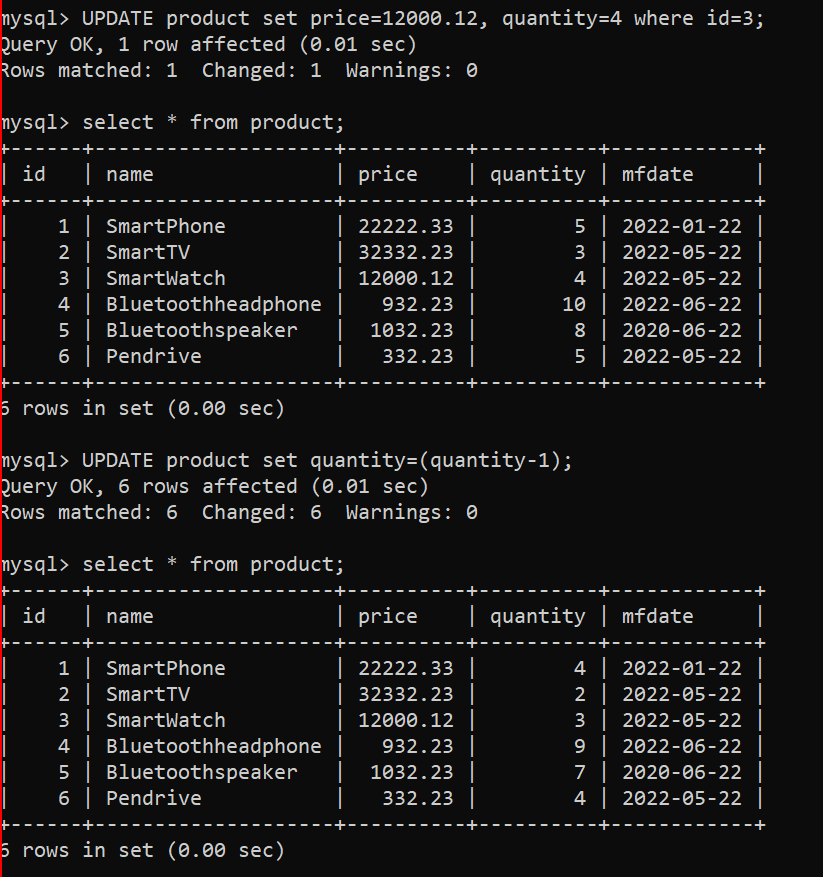
* 1. Update the existing record

You can update the record of all the rows or selected rows

Syntax:

UPDATE <table\_name> set column=value, column=value where <condition>;

UPDATE <table\_name> set column=value, column=value;

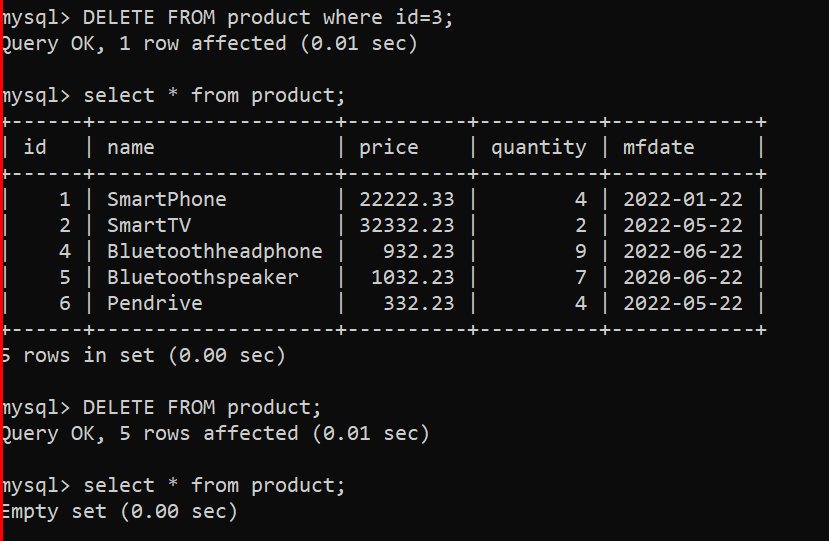


* 1. Delete The records.

You can delete the specific rows from the table using delete query.

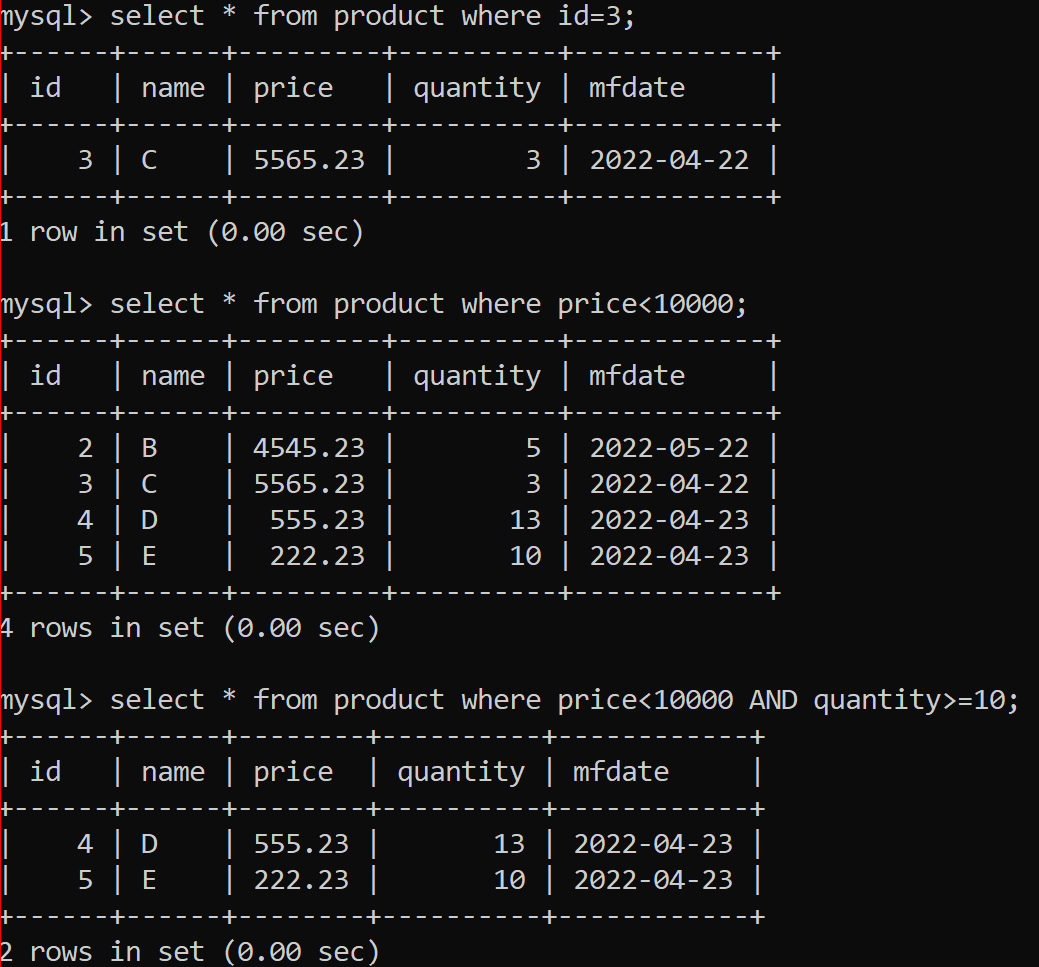
Syntax:

DELETE FROM <tablename> where <condition>;

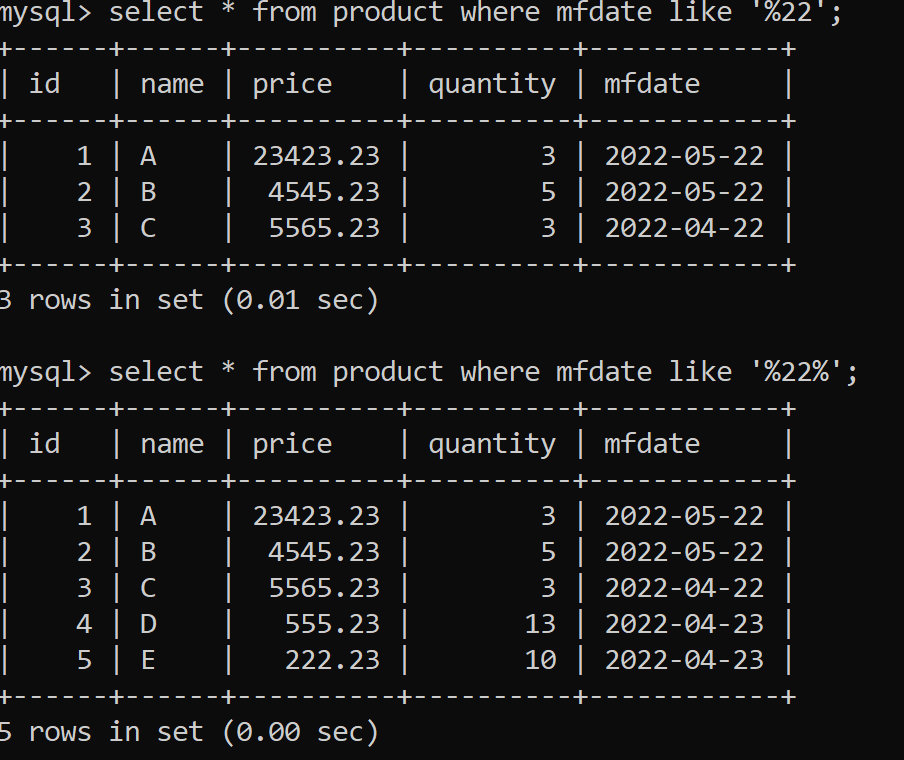


**Clauses**

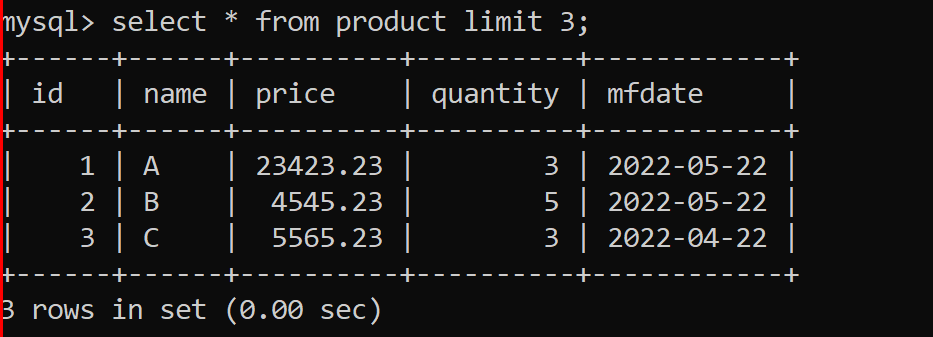
1. This is used to apply the row level filtration.
2. Clauses are followed with operators or conditions
3. Some Commonly used clauses
   1. Where clause.
      1. Using this clause you can write a condition on the rows.
      2. This is one of the way to filter the data while retrieving the data or update or delete.



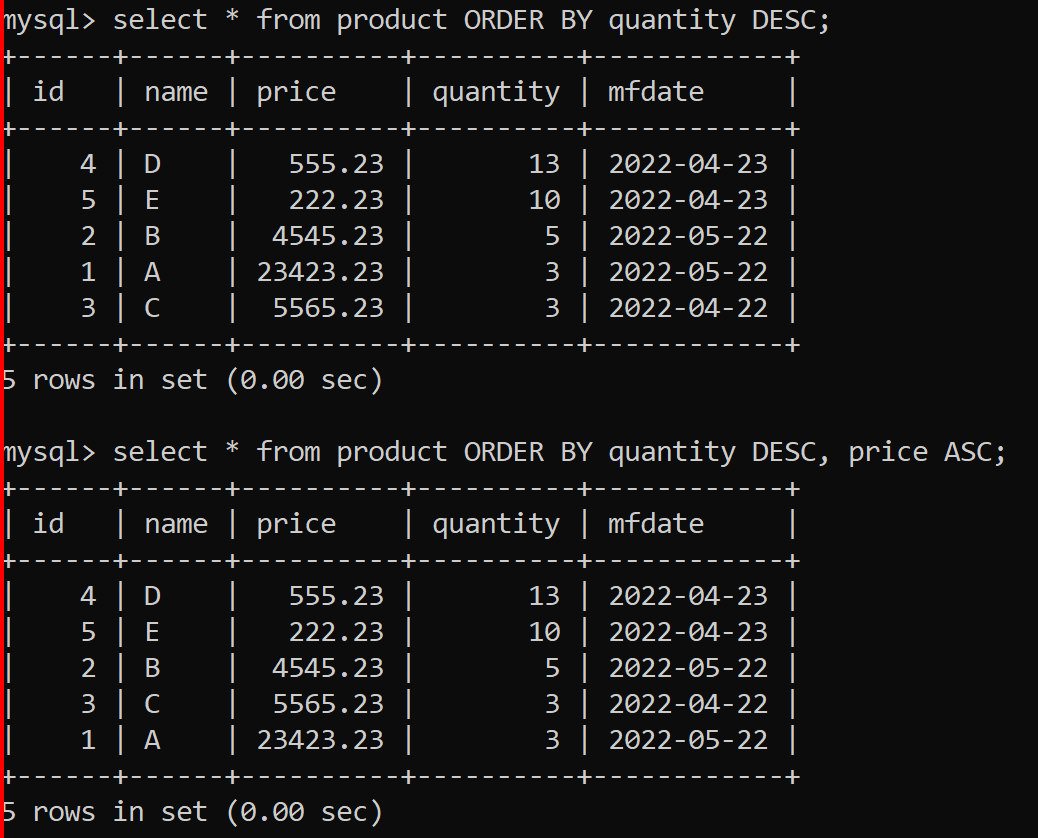
* 1. Like Clause
     1. Using this clause you can select the records using the partial values



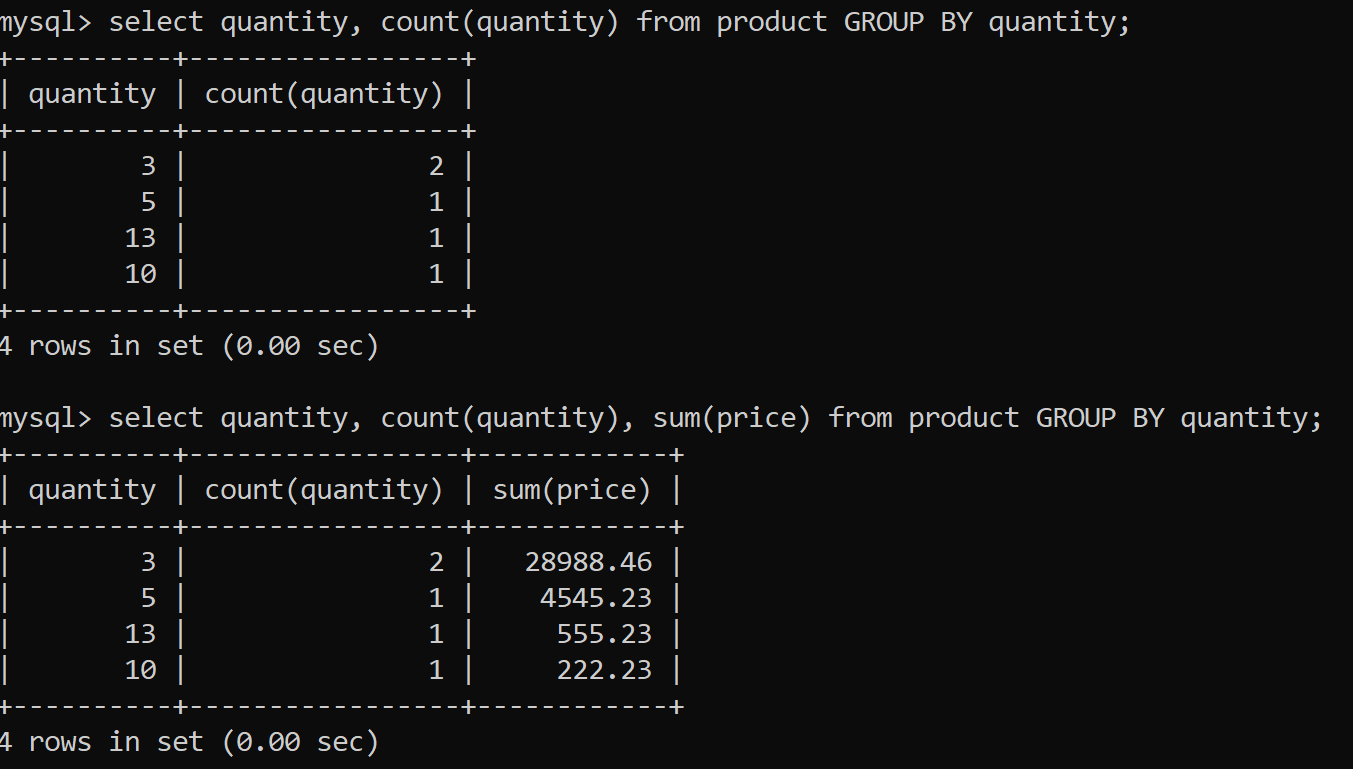
* 1. Limit Clause
     1. Used to limit the number of records getting selected.



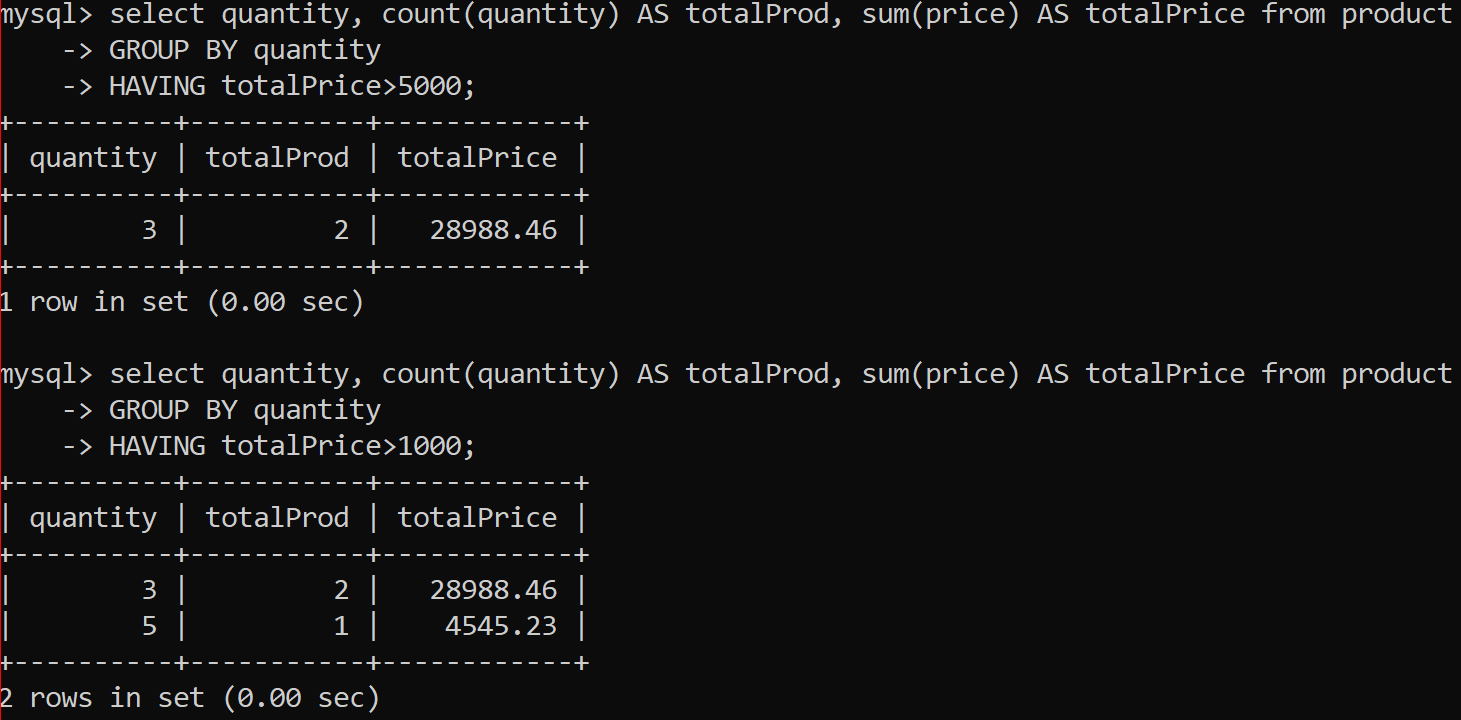
* 1. Order by clause
     1. To select a records based on specific order.
     2. Records can be selected by Ascending (ASC) or Descending (DESC) order.



* 1. **Group By**
     1. Is use to group the records using some functions
     2. This is mostly used with a Aggregate functions.



* 1. **Having Clause**
     1. Is used to apply conditions on the Group record output
     2. This has to use with GROUP BY clause only

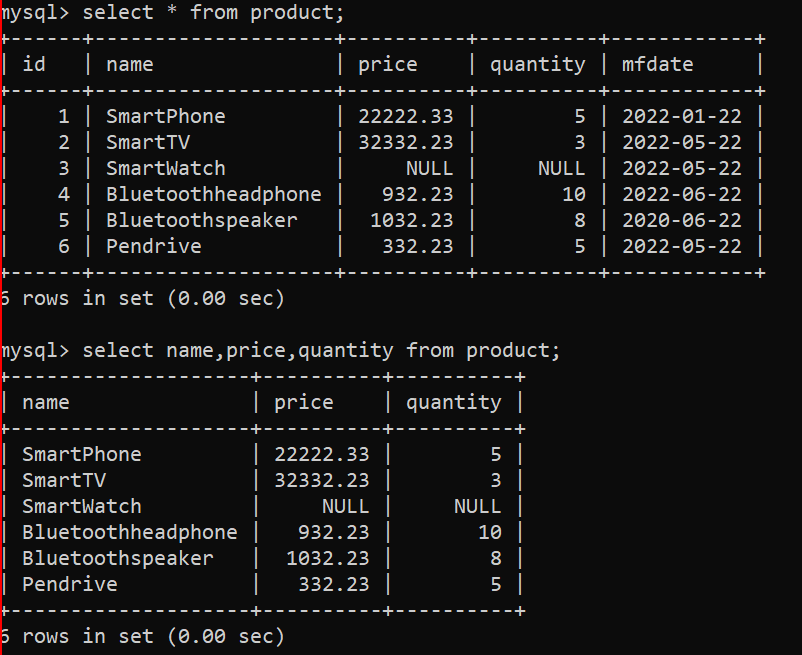


**Data Query Language (DQL)**

1. In this category you can retrieve the data from the table.
2. In this SELECT query is used to retrieve the data.
   1. Syntax

SELECT \* FROM <table>;

SELECT columns FROM <table>;



**Sql Constraints**

1. It is a way to apply a restriction on the columns of the table.
2. This restriction will be applied at the time of inserting data.
3. Using constraint, you can make sure that the valid data is getting inserted into DB.
4. There are some constraints are as follows
   1. **Not Null**: You can make sure that value are not null, and user is inserting value for the column.
   2. **Unique**: you can make sure that the values are unique and no duplicate values are allowed for the column.
   3. **Check**: you can apply a check on the values of the column. Check like the range of value or can allowed specific list of values.
   4. **Default**: To provide a default value for the column. By default the NULL values will be used as a default value.
   5. **Primary Key**: It is a combination of Not Null and Unique. Primary key column can be used as a foreign key in another table to make a relation between 2 tables. By Default, the indexes (Used for the faster searching/retrieval) will be applied on the Primary key column.
   6. **Foreign Key**: Foreign key values are the values refer from the primary key. In foreign key column values can be duplicate or values can be null. Using this key you can relate with the table having primary Key.

CREATE TABLE employee(

id int PRIMARY KEY,

name varchar(20) NOT NULL,

email varchar(50) NOT NULL UNIQUE,

age int,

gender varchar(10),

country varchar(10) DEFAULT 'IN',

CONSTRAINT gender\_chk CHECK (gender IN ('Male', 'Female', 'Others')),

CONSTRAINT age\_chk CHECK (age between 21 AND 60)

);

CREATE TABLE edudetails(

eduid int PRIMARY KEY,

title varchar(20) NOT NULL,

passingyear varchar(4),

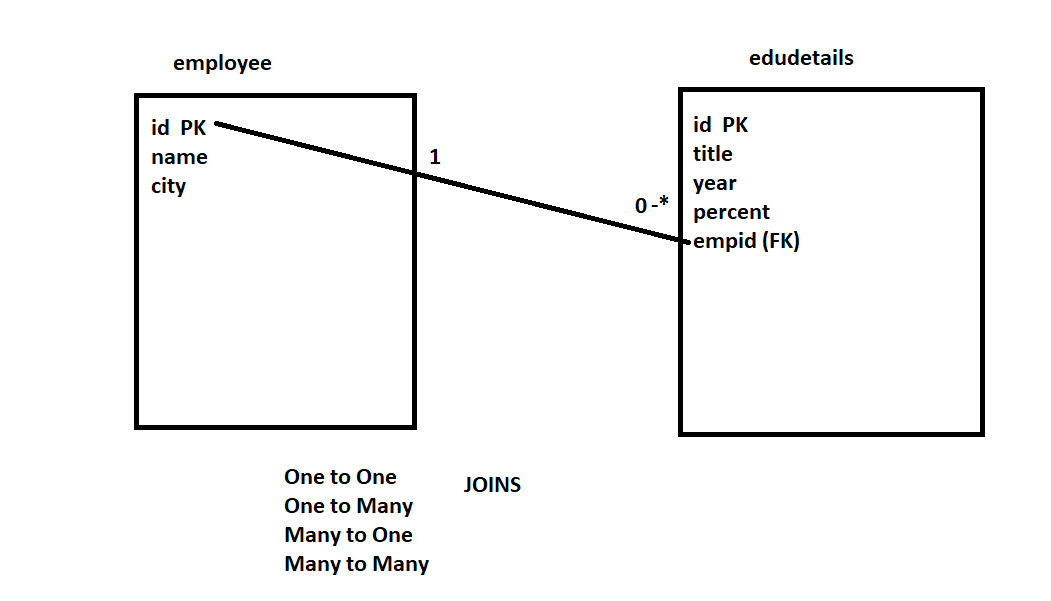
percent double,

empid int,

FOREIGN KEY(empid) REFERENCES employee(id),

CONSTRAINT percent\_chk CHECK (percent between 0 AND 100)

);



**Joins**

1. Is used to retrieve the data from more than one table.
2. Join queries are always used with select query.
3. There are different types of join
   1. Cross Join (Cartesian Join)
   2. Inner Join
   3. Outer Join
      1. Left Join
      2. Right Join
      3. Full Join (Not supported in MYSQL can be used with Oracle, Postgrese)



**Cross Join**

1. Every row of the one table will be mapped with the every row from the second table.
2. In this Join no where clause is used



**Inner Join**

1. Inner join selects only the matching records from both the table.
2. Inner join has to come with the condition which can be applied to mapped a records from one table to another table.
3. Syntax

SELECT coumn(s)

FROM table1 **INNER JOIN** table2

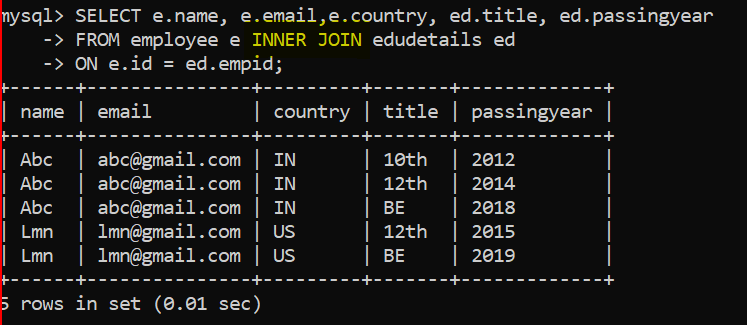
**ON** condition

**INNER JOIN** table3

**ON** condition

**INNER JOIN** table4

**ON** condition

****

**Left Outer Join**

1. In this join all the records (matching and non-matching records) from the left table will be selected and only the matching record from the right table will be selected.
2. Syntax

SELECT coumn(s)

FROM table1 **LEFT JOIN** table2

**ON** condition

**LEFT JOIN** table3

**ON** condition

**LEFT JOIN** table4

**ON** condition



**Right Outer Join**

1. In this join all the records (matching and non-matching records) from the right table will be selected and only the matching record from the left table will be selected.
2. Syntax

SELECT coumn(s)

FROM table1 **RIGHT JOIN** table2

**ON** condition

**RIGHT JOIN** table3

**ON** condition

**RIGHT JOIN** table4

**ON** condition



**Full Outer Join**

1. In this join all the records (matching and non-matching records) all the tables will be selected.
2. Syntax

SELECT coumn(s)

FROM table1 **FULL JOIN** table2

**ON** condition

**FULL JOIN** table3

**ON** condition

**FULL JOIN** table4

**ON** condition