

Name : Veeransh Shah

Reg ID : 221070063

Assignment - 6

AIM : Study of aggregate functions, clauses and transaction commands on database using SQL.

TOOL : MariaDB **INDEX :**

Sr No.	CONTENTS
1)	Aggregate Functions
2)	Clauses: group by, having, order by, limit, with, case
3)	Transaction Commands
4)	Operations executed
5)	Order of execution of clauses

PROGRAMMING LANGUAGE : SQL (Structured Query Language)

THEORY :

1. AGGREGATE FUNCTIONS:

Aggregate functions are SQL functions that perform a calculation on a set of values and return a single value. These functions are often used with the SELECT statement to summarize information from a database table. Common aggregate functions include:

COUNT(): Counts the number of rows in a result set or the number of non-null values in a specific column.

SUM(): Calculates the sum of values in a numeric column.

AVG(): Computes the average value of a numeric column.

MIN(): Returns the minimum value in a column.

MAX(): Returns the maximum value in a column.

Example : SELECT COUNT(*), AVG(salary), MAX(age) FROM employees;

2. CLAUSES:

SQL clauses are keywords used in conjunction with SQL statements to filter, sort, group, or perform other operations on data. Common SQL clauses include:

ORDER BY: Sorts the result set based on specified columns.

Example:

```
SELECT * FROM employees
ORDER BY hire_date DESC;
```

GROUP BY: Groups rows that have the same values into summary rows.

Example:

```
SELECT customer_id, SUM(order_total) AS total_orders
FROM orders
GROUP BY customer_id;
```

HAVING: Specifies conditions for grouping when using the GROUP BY clause.

Example:

```
SELECT customer_id, COUNT(order_id) AS order_count,
SUM(order_total) AS total_amount
FROM orders
GROUP BY customer_id
HAVING COUNT(order_id) > 2;
```

LIMIT: Limits the number of rows returned in the result set.

Example:

```
SELECT * FROM products
ORDER BY price ASC
LIMIT 5;
```

WITH : The WITH clause in SQL, also known as Common Table Expressions (CTE), allows you to define temporary result sets that can be used within the scope of a single SQL statement, often for readability and reusability purposes. It's especially useful when you need to break down complex queries into simpler, more manageable parts. **Example:**

```
WITH RecursiveManager AS (
    SELECT employee_id, first_name, last_name, manager_id
    FROM employees
    WHERE employee_id = 5 -- Assuming you want to find the hierarchy
    for employee with id 5
```

```
    UNION ALL
```

```
    SELECT e.employee_id, e.first_name, e.last_name,
e.manager_id
    FROM employees e
    INNER JOIN RecursiveManager rm ON e.manager_id =
rm.employee_id )
```

```
SELECT * FROM RecursiveManager;
```

CASE : In SQL, the CASE statement is used to perform conditional logic within a query. **Example:**

```
SELECT student_id,  
       first_name,  
       last_name,  
       grade,  
       CASE  
           WHEN grade >= 90 THEN 'A'  
           WHEN grade >= 80 THEN 'B'  
           WHEN grade >= 70 THEN 'C'  
           WHEN grade >= 60 THEN 'D'  
           ELSE 'F'  
       END AS grade_category FROM  
students;
```

3. TRANSACTION COMMANDS:

Transaction commands in SQL are used to control transactions, which are sequences of operations performed as a single unit of work. Transactions ensure data integrity by allowing multiple database operations to either succeed or fail together. Common transaction commands include:

BEGIN TRANSACTION or START TRANSACTION: Begins a new transaction.

COMMIT: Saves all changes made during the transaction and ends the transaction.

ROLLBACK: Undoes all changes made during the transaction and ends the transaction.

SAVEPOINT: Sets a point within a transaction to which you can later roll back.

Example:

```
START TRANSACTION;
```

```
UPDATE accounts SET balance = balance - 100 WHERE account_id = 123;
```

```
UPDATE accounts SET balance = balance + 100 WHERE account_id = 456;
```

```
COMMIT;
```

In this example, the UPDATE statements are part of a transaction that transfers 100 units of currency from one account to another. The COMMIT command is used to finalize the transaction and make the changes permanent in the database. If an error occurs or if the transaction needs to be rolled back for any reason, the ROLLBACK command can be used to revert the changes.

OPERATIONS EXECUTED :

Aggregate Function	sum, count(*), count(distinct()), min, max, avg
--------------------	--

clauses Group by,order by,having
Transaction commands Start transaction
savepoint
rollback
commit

other as,with,limit,case **ORDER OF EXECUTION :**

From -> Join -> where -> group by -> having -> select -> distinct -> order by -> limit

CODE :

Enter password: ****

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 3

Server version: 11.4.0-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database db20;

Query OK, 1 row affected (0.015 sec)

MariaDB [(none)]> use database db20;

ERROR 1049 (42000): Unknown database 'database'

MariaDB [(none)]> use db20;

Database changed

MariaDB [db20]> create table products(p_id int, p_name char(30), Supplier_id int, category_id int, price int); Query OK, 0 rows affected (0.058 sec)

MariaDB [db20]> insert into products values

(1,"Chas",1,1,18),(2,"Chai",1,1,19),(3,"Syrup",1,2,10),(4,"Se asoning",2,2,22),(5,"Gumbo Mix",2,2,21),(6,"Berry Spread",3,2,25),(7,"Dried

Pears",3,7,30),(8,"Sauce",3,2,40),(9,"Kobe Niku",4,6,97);

Query OK, 9 rows affected (0.025 sec)

Records: 9 Duplicates: 0 Warnings: 0

MariaDB [db20]> select * from products;

```
+-----+-----+-----+-----+-----+ | p_id |
p_name | Supplier_id | category_id | price |
```

```

+-----+-----+-----+-----+-----+ | 1 |
Chas | 1 | 1 | 18 | | 2 | Chai | 1 | 1 | 19 | | 3 | Syrup | 1 | 2 |
10 | | 4 | Seasoning | 2 | 2 | 22 | | 5 | Gumbo Mix | 2 | 2 | 21 | |
6 | Berry Spread | 3 | 2 | 25 |
| 7 | Dried Pears | 3 | 7 | 30 | | 8 | Sauce | 3 | 2 | 40 | | 9 |
Kobe      Niku      |      4      |      6      |      97      |
+-----+-----+-----+-----+-----+

```

9 rows in set (0.009 sec)

MariaDB [db20]> select sum(price) from products;

```

+-----+ |
sum(price) |
+-----+
| 282 |
+-----+

```

1 row in set (0.003 sec)

MariaDB [db20]> select count(p_id) from products;

```

+-----+ |
count(p_id) |
+-----+
| 9 |
+-----+

```

1 row in set (0.001 sec)

MariaDB [db20]> select min(price) as min_price from products;

```

+-----+ |
min_price |
+-----+
| 10 |
+-----+

```

1 row in set (0.004 sec)

MariaDB [db20]> select max(price) as max_price from products;

```

+-----+ |
max_price |
+-----+
| 97 |
+-----+

```

1 row in set (0.001 sec)

MariaDB [db20]> select avg(price) as avg_price from products;

```

+-----+ |
avg_price |
+-----+
| 31.3333 |
+-----+

```

1 row in set (0.000 sec)

```
MariaDB [db20]> select count(supplier_id) from products;
```

```
+-----+ |
count(supplier_id) |
+-----+
| 9 |
+-----+
1 row in set (0.000 sec)
```

```
MariaDB [db20]> select count(distinct(supplier_id)) from products;
```

```
+-----+ |
count(distinct(supplier_id)) |
+-----+
| 4 |
+-----+ 1
row in set (0.003 sec)
```

```
MariaDB [db20]> select count(*) from products;
```

```
+-----+ |
count(*) |
+-----+
| 9 |
+-----+
1 row in set (0.001 sec)
```

```
MariaDB [db20]> -- Fetch the count of products supplied by each
supplier
```

```
MariaDB [db20]> select supplier_id,count(p_id) as num_of_products
from products group by supplier_id;
```

```
+-----+-----+ |
supplier_id | num_of_products |
+-----+-----+
| 1 | 3 |
| 2 | 2 |
| 3 | 3 |
| 4 | 1 |
+-----+-----+ 4
rows in set (0.003 sec)
```

```
MariaDB [db20]> -- fetch the count of number of records for each
supplier id;
```

```
MariaDB [db20]> select supplier_id,count(*) as num_of_records from
products group by supplier_id; +-----+-----+ |
```

```
supplier_id | num_of_records | +-----+-----+ |
1 | 3 |
| 2 | 2 |
| 3 | 3 |
| 4 | 1 |
+-----+-----+ 4
```

rows in set (0.001 sec)

```
MariaDB [db20]> -- fetch the max price under each category
MariaDB [db20]> select category_id,max(price) as max_price from products
group by category_id;
```

```
+-----+-----+ |
category_id | max_price |
+-----+-----+
| 1 | 19 |
| 2 | 40 |
| 6 | 97 |
| 7 | 30 |
```

```
+-----+-----+ 4
rows in set (0.001 sec)
```

```
MariaDB [db20]> -- fetch the highest price >19 under each category
MariaDB [db20]> select category_id,max(price) as
Highest_price from products group by category_id having
Highest_price>19;
```

```
+-----+-----+ |
category_id | Highest_price |
+-----+-----+
                2 | 40 |
                6 | 97 |
                7 | 30 |
```

```
+-----+-----+ 3
rows in set (0.005 sec)
```

```
MariaDB [db20]> select category_id,max(price) as
Highest_price from products group by category_id having
Highest_price>19 order by Highest_price;
```

```
+-----+-----+ |
category_id | Highest_price |
+-----+-----+
| 7 | 30 |
| 2 | 40 |
| 6 | 97 |
```

```
+-----+-----+ 3
rows in set (0.000 sec)
```

```
MariaDB [db20]> select category_id,max(price) as
Highest_price from products group by category_id having
Highest_price>19 order by category_id desc;
```

```
+-----+-----+ |
category_id | Highest_price |
+-----+-----+
| 7 | 30 |
```

```

| 6 | 97 |
| 2 | 40 |
+-----+-----+-----+ 3
rows in set (0.001 sec)

```

```

MariaDB [db20]> create table student (fname char(50), lname char(50),
age int);

```

```

Query OK, 0 rows affected (0.018 sec)

```

```

MariaDB [db20]> insert into student
values("Anjali","Ade",26), ("Anjali","Ade",26), ("Anjali","Rai"
,27), ("Anjali","Dubey",21);

```

```

Query OK, 4 rows affected (0.003 sec)

```

```

Records: 4 Duplicates: 0 Warnings: 0

```

```

|
|
|
MariaDB [db20]> select * from student;

```

```

+-----+-----+-----+
| fname | lname | age |
+-----+-----+-----+
| Anjali | Ade | 26 |
| Anjali | Ade | 26 |
| Anjali | Rai | 27 |
Anjali | Dubey | 21 |
+-----+-----+-----+
4 rows in set (0.001 sec)

```

```

MariaDB [db20]> -- use of multiple columns in the group by clause

```

```

MariaDB [db20]> select *,count(*) from student group by
fname,lnamet,age;

```

```

ERROR 1054 (42S22): Unknown column 'lnamet' in 'group statement'

```

```

MariaDB [db20]> select *,count(*) from student group by
fname,lname,age;

```

```

+-----+-----+-----+-----+
| fname | lname | age | count(*) |
+-----+-----+-----+-----+
| Anjali | Ade | 26 | 2 | | Anjali | Dubey
| 21 | 1 | | Anjali | Rai | 27 | 1 |
+-----+-----+-----+-----+ 3
rows in set (0.001 sec)

```

```

MariaDB [db20]> -- use of order by clause

```

```

MariaDB [db20]> select p_id,p_name,supplier_id from products order by
1;

```

```

+-----+-----+-----+

```


p_id	p_name	supplier_id
1	Chas	1
2	Chai	1
3	Syrup	1
4	Seasoning	2
5	Gumbo Mix	2
6	Berry Spread	3
7	Dried Pears	3
8	Sauce	3
9	Kobe Niku	4

rows in set (0.002 sec)

MariaDB [db20]> -- data should be sorted as per 1st column
 MariaDB [db20]> select p_id,p_name,supplier_id from products order by 2;

p_id	p_name	supplier_id
6	Berry Spread	3
2	Chai	1
1	Chas	1
7	Dried Pears	3
5	Gumbo Mix	2
9	Kobe Niku	4
8	Sauce	3
4	Seasoning	2
3	Syrup	1

rows in set (0.001 sec)

MariaDB [db20]> -- data should be sorted as per 2nd column
 MariaDB [db20]> select p_id from products limit 3;

p_id
1
2
3

3 rows in set (0.001 sec)

MariaDB [db20]> select p_id from products limit 1,5;

p_id
2
3

```

| 4 |
| 5 |
| 6 |
+-----+
|
|
|
5 rows in set (0.002 sec)

```

```

MariaDB [db20]> -- skip 1st value display next 5 tuples
MariaDB [db20]> select p_id from products limit 0,5;

```

```

+-----+
| p_id |
+-----+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
+-----+
5 rows in set (0.001 sec)

```

```

MariaDB [db20]> select p_id from products limit 10,1;
Empty set (0.001 sec)

```

```

MariaDB [db20]> select p_id from products limit 9,1;
Empty set (0.001 sec)

```

```

MariaDB [db20]> select * from products order by p_id desc limit 5;
+-----+-----+-----+-----+-----+ | p_id |
p_name | Supplier_id | category_id | price |
+-----+-----+-----+-----+-----+ | 9 |
Kobe Niku | 4 | 6 | 97 | 8 | Sauce | 3 | 2 | 40 |
          7 | Dried Pears | 3 | 7 | 30 | 6 | Berry Spread | 3 | 2 | 25
          |
          | 5 | Gumbo Mix | 2 | 2 | 21 |
+-----+-----+-----+-----+-----+ 5 rows
in set (0.001 sec)

```

```

MariaDB [db20]> select * from products order by p_id desc limit 1,5;
+-----+-----+-----+-----+-----+ | p_id |
p_name | Supplier_id | category_id | price |
+-----+-----+-----+-----+-----+
| 8 | Sauce | 3 | 2 | 40 | | 7 | Dried Pears | 3 | 7 | 30 | | 6 |

```

```

Berry Spread | 3 | 2 | 25 | | 5 | Gumbo Mix | 2 | 2 | 21 | | 4 |
Seasoning      |      2      |      2      |      22      |
+-----+-----+-----+-----+-----+ 5 rows
in set (0.001 sec)

```

MariaDB [db20]> -- above shows use of limit

MariaDB [db20]> -- use of concat

```

MariaDB [db20]> select concat('my favourite product
is',upper(p_name),' ',price,' ','!') as yell from products order by
p_id asc;

```

```

+-----+
| yell |
+-----+
| my favourite product isCHAS 18 ! |
| my favourite product isCHAI 19 ! | | my
favourite product isSYRUP 10 ! | | my favourite
product isSEASONING 22 ! | | my favourite product
isGUMBO MIX 21 ! | | my favourite product
isBERRY SPREAD 25 ! | | my favourite product
isDRIED PEARS 30 ! | | my favourite product isSAUCE
40 ! | | my favourite product isKOBE NIKU 97 ! |
+-----+
9 rows in set (0.008 sec)

```

```

MariaDB [db20]> select concat('my favourite product is
',upper(p_name),' ',price,' ','!') as yell from products order by
p_id asc;

```

```

+-----+
| yell |
+-----+
| my favourite product is CHAS 18 ! | | my
favourite product is CHAI 19 ! | | my
favourite product is SYRUP 10 ! | | my favourite
product is SEASONING 22 ! | | my favourite product
is GUMBO MIX 21 ! | | my favourite product is
BERRY SPREAD 25 ! | | my favourite product is DRIED
PEARS 30 ! | | my favourite product is SAUCE 40 !
| | my favourite product is KOBE NIKU 97 ! |
|
|
|
+-----+ 9 rows
in set (0.001 sec)

```

MariaDB [db20]> -- with

```

MariaDB [db20]> with q1 as (select p_id from products where p_id>2)
select * from q1;

```

```

+-----+

```

```

| p_id |
+-----+
| 3 |
| 4 |
| 5 |
| 6 |
| 7 |
| 8 |
| 9 |
+-----+
7 rows in set (0.003 sec)

```

MariaDB [db20]> -- CASE

```

MariaDB [db20]> select p_name,
-> case supplier_id
-> when 1 then 'amazon.in'
-> when 2 then 'bigbasket.com'
-> else 'flipkart.com' end as supplier_name
-> from products;

```

```

+-----+-----+
| p_name | supplier_name |
+-----+-----+
| Chas | amazon.in |
| Chai | amazon.in | | Syrup
| amazon.in |
| Seasoning | bigbasket.com | |
Gumbo Mix | bigbasket.com | | Berry
Spread | flipkart.com | | Dried
Pears | flipkart.com | | Sauce |
flipkart.com | | Kobe Niku |
flipkart.com |
+-----+-----+ 9
rows in set (0.003 sec)

```

MariaDB [db20]> -- use of case and with together MariaDB [db20]>
with q2 as (select p_name, case supplier_id when 1 then
'amazon.in' when 2 then 'bigbasket.com' else 'flipkart.com' end as
supplier_name from products) select supplier_name, count(p_name)
from q2 group by supplier_name;

```

+-----+-----+ |
supplier_name | count(p_name) |
+-----+-----+
| amazon.in | 3 |
| bigbasket.com | 2 | | flipkart.com |
4 |
+-----+-----+ 3
rows in set (0.003 sec)

```

MariaDB [db20]> -- transaction commands MariaDB

[db20]> start transaction

-> ;

Query OK, 0 rows affected (0.003 sec)

MariaDB [db20]> select * from products;

```
+-----+-----+-----+-----+-----+ | p_id |
| p_name | Supplier_id | category_id | price |
+-----+-----+-----+-----+-----+ | 1 |
Chas | 1 | 1 | 18 | | 2 | Chai | 1 | 1 | 19 | | 3 | Syrup | 1 | 2 |
| 10 | | 4 | Seasoning | 2 | 2 | 22 | | 5 | Gumbo Mix | 2 | 2 | 21 |
| | 6 | Berry Spread | 3 | 2 | 25 | | 7 | Dried Pears | 3 | 7 | 30 |
| | 8 | Sauce | 3 | 2 | 40 | | 9 | Kobe Niku | 4 | 6 | 97 |
+-----+-----+-----+-----+-----+ 9 rows
in set (0.001 sec)
```

MariaDB [db20]> insert into products
values(10,"MilkPowder",4,6,20); Query
OK, 1 row affected (0.001 sec)

MariaDB [db20]> savepoint S1;
Query OK, 0 rows affected (0.001 sec)

MariaDB [db20]> insert into products
values(11,"MilkPowder",4,6,20); Query
OK, 1 row affected (0.001 sec)

MariaDB [db20]> savepoint S2;
Query OK, 0 rows affected (0.000 sec)

```
MariaDB [db20]> select * from products;
+-----+-----+-----+-----+-----+ | p_id |
| p_name | Supplier_id | category_id | price |
+-----+-----+-----+-----+-----+ | 1 |
Chas | 1 | 1 | 18 | | 2 | Chai | 1 | 1 | 19 | | 3 | Syrup | 1 | 2 |
10 | | 4 | Seasoning | 2 | 2 | 22 | | 5 | Gumbo Mix | 2 | 2 | 21 | |
6 | Berry Spread | 3 | 2 | 25 | | 7 | Dried Pears | 3 | 7 | 30 | | 8 |
| Sauce | 3 | 2 | 40 | |
| 9 | Kobe Niku | 4 | 6 | 97 | | 10 | MilkPowder | 4 | 6 | 20 | |
11 | | MilkPowder | | 4 | | 6 | | 20 | |
+-----+-----+-----+-----+-----+ 11 rows
in set (0.001 sec)
```

MariaDB [db20]> rollback to S1;
Query OK, 0 rows affected (0.002 sec)

```

MariaDB [db20]> select * from products;
+-----+-----+-----+-----+-----+ | p_id |
p_name | Supplier_id | category_id | price |
+-----+-----+-----+-----+-----+ | 1 |
Chas | 1 | 1 | 18 | | 2 | Chai | 1 | 1 | 19 | | 3 | Syrup | 1 | 2 |
10 | | 4 | Seasoning | 2 | 2 | 22 | | 5 | Gumbo Mix | 2 | 2 | 21 | |
6 | Berry Spread | 3 | 2 | 25 | | 7 | Dried Pears | 3 | 7 | 30 | | 8
| Sauce | 3 | 2 | 40 | | 9 | Kobe Niku | 4 | 6 | 97 | | 10 |
MilkPowder | 4 | 6 | 20 |
+-----+-----+-----+-----+-----+ 10 rows
in set (0.000 sec)

```

```

MariaDB [db20]> commit
-> ;
Query OK, 0 rows affected (0.003 sec)

```

MariaDB [db20]>

SNAPSHOTS :

```

MySQL: Client [MariaDB 11.4.0] - "C:\Program Files\MariaDB 11.4\bin\mysql.exe" --defaults-file="C:\Program Files\MariaDB 11.4\data\my.cnf" --user=p
Enter password: ****
Welcome to the MariaDB monitor.  Commands and with ; or \g.
Your MariaDB connection id is 3
Server version: 11.4.0-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> create database db20;
Query OK, 1 row affected (0.015 sec)

MariaDB [(none)]> use database db20;
ERROR 1009 (42000): Unknown database 'database'
MariaDB [(none)]> use db20;
Database changed
MariaDB [db20]> create table products(p_id int, p_name char(30), Supplier_id int, category_id int, price int);
Query OK, 0 rows affected (0.058 sec)

MariaDB [db20]> insert into products values (1,"Chas",1,1,18),(2,"Chai",1,1,19),(3,"Syrup",1,2,38),(4,"Seasoning",1,2,22),(5,"Gumbo Mix",2,2,21),(6,"Berry Spread",3,2,25),(
7,"Dried Pears",3,7,30),(8,"Sauce",3,2,40),(9,"Kobe Niku",4,6,97);
Query OK, 9 rows affected (0.029 sec)
Records: 9 Duplicates: 0 Warnings: 0

MariaDB [db20]> select * from products;
+-----+-----+-----+-----+-----+
| p_id | p_name      | Supplier_id | category_id | price |
+-----+-----+-----+-----+-----+
| 1    | Chas        | 1           | 1           | 18    |
| 2    | Chai        | 1           | 1           | 19    |
| 3    | Syrup       | 1           | 2           | 38    |
| 4    | Seasoning   | 1           | 2           | 22    |
| 5    | Gumbo Mix   | 2           | 2           | 21    |
| 6    | Berry Spread | 3           | 2           | 25    |
| 7    | Dried Pears | 3           | 7           | 30    |
| 8    | Sauce       | 3           | 2           | 40    |
| 9    | Kobe Niku   | 4           | 6           | 97    |
+-----+-----+-----+-----+-----+
9 rows in set (0.009 sec)

```

```
MySQL Client [MariaDB 11.4.3] - "C:\Program Files\MariaDB 11.4\bin\mysql.exe" --defaults-file="C:\Program Files\MariaDB 11.4\data\my.cnf" --user=p

+-----+
| category_id | max_price |
+-----+
| 1           | 19        |
| 2           | 40        |
| 6           | 97        |
| 7           | 30        |
+-----+
4 rows in set (0.001 sec)

MariaDB [db30] > -- fetch the highest price >19 under each category
MariaDB [db30] > select category_id,max(price) as Highest_price from products group by category_id having Highest_price>19;
+-----+
| category_id | Highest_price |
+-----+
| 2           | 40            |
| 6           | 97            |
| 7           | 30            |
+-----+
3 rows in set (0.005 sec)

MariaDB [db30] > select category_id,max(price) as Highest_price from products group by category_id having Highest_price>19 order by Highest_price;
+-----+
| category_id | Highest_price |
+-----+
| 7           | 30            |
| 2           | 40            |
| 6           | 97            |
+-----+
3 rows in set (0.000 sec)

MariaDB [db30] > select category_id,max(price) as Highest_price from products group by category_id having Highest_price>19 order by category_id desc;
+-----+
| category_id | Highest_price |
+-----+
| 7           | 30            |
| 6           | 97            |
| 2           | 40            |
+-----+
3 rows in set (0.001 sec)
```

```
MySQL Client [MariaDB 11.4.3] - "C:\Program Files\MariaDB 11.4\bin\mysql.exe" --defaults-file="C:\Program Files\MariaDB 11.4\data\my.cnf" --user=p

+-----+
| 7 |
| 6 |
| 2 |
+-----+
3 rows in set (0.003 sec)

MariaDB [db30] > -- case
MariaDB [db30] > select p_name,
-> case supplier_id
-> when 1 then 'amazon.in'
-> when 2 then 'bigbasket.com'
-> else 'flipkart.com' end as supplier_name
-> from products;
+-----+
| p_name      | supplier_name |
+-----+
| Chas        | amazon.in     |
| Chai        | amazon.in     |
| Syrup       | amazon.in     |
| Seasoning   | bigbasket.com |
| Dumbo Mix   | bigbasket.com |
| Berry Spread | flipkart.com  |
| Dried Peas  | flipkart.com  |
| Sauce       | flipkart.com  |
| Koba Miku   | flipkart.com  |
+-----+
9 rows in set (0.003 sec)

MariaDB [db30] > -- use of case and with together
MariaDB [db30] > with q1 as (select p_name, case supplier_id when 1 then 'amazon.in' when 2 then 'bigbasket.com' else 'flipkart.com' end as supplier_name from products) select
supplier_name,count(p_name) from q1 group by supplier_name;
+-----+
| supplier_name | count(p_name) |
+-----+
| amazon.in     | 3              |
| bigbasket.com | 2              |
| flipkart.com  | 4              |
+-----+
3 rows in set (0.003 sec)
```

```
MySQl Client [MariaDB 11.4.36] - "C:\Program Files\MariaDB 11.4\bin\mysql.exe" --defaults-file="C:\Program Files\MariaDB 11.4\data\my.cnf" --user=root -p

MariaDB [db30]> select * from products;
+----+-----+-----+-----+-----+
| p_id | p_name      | Supplier_id | category_id | price |
+----+-----+-----+-----+-----+
| 1    | Oats        | 1           | 1           | 18     |
| 2    | Oat         | 1           | 1           | 19     |
| 3    | Syrup       | 1           | 2           | 18     |
| 4    | Seasoning   | 2           | 2           | 22     |
| 5    | Gumbo Mix   | 2           | 1           | 21     |
| 6    | Berry Spread | 3           | 1           | 25     |
| 7    | Dried Pears | 3           | 7           | 30     |
| 8    | Sauce       | 3           | 2           | 40     |
| 9    | Kobe Nibs   | 4           | 6           | 97     |
| 10   | MilkPowder  | 4           | 6           | 28     |
| 11   | MilkPowder  | 4           | 6           | 28     |
+----+-----+-----+-----+-----+
11 rows in set (0.001 sec)

MariaDB [db30]> rollback to 51;
Query OK, 0 rows affected (0.002 sec)

MariaDB [db30]> select * from products;
+----+-----+-----+-----+-----+
| p_id | p_name      | Supplier_id | category_id | price |
+----+-----+-----+-----+-----+
| 1    | Oats        | 1           | 1           | 18     |
| 2    | Oat         | 1           | 1           | 19     |
| 3    | Syrup       | 1           | 2           | 18     |
| 4    | Seasoning   | 2           | 2           | 22     |
| 5    | Gumbo Mix   | 2           | 1           | 21     |
| 6    | Berry Spread | 3           | 1           | 25     |
| 7    | Dried Pears | 3           | 7           | 30     |
| 8    | Sauce       | 3           | 2           | 40     |
| 9    | Kobe Nibs   | 4           | 6           | 97     |
| 10   | MilkPowder  | 4           | 6           | 28     |
+----+-----+-----+-----+-----+
10 rows in set (0.000 sec)

MariaDB [db30]>
```

CONCLUSION : Hence, we have successfully studied aggregate functions, clauses and transaction commands on database using SQL.