DBMS Practical

Abhay Shanker Pathak

2021-01-20

Contents

1 To create a DDL to perform creation of table, alter, modify and drop column 3
1.1 The Create Table Command 3
1.2 Modify the structure of table 3
1.2.1 Add new column 3
1.3 Dropping a column from a table 3
1.4 Modify existing column 4
1.5 Renaming the table 4
1.6 Destroying table 4

2 To study various DML commands 5
2.1 Insert a single record into dept table 5
2.2 Insert more than a recored into emp table using a single insert command . . 5
2.3 Update the emp table to set the salary of all the employees to Rs 15000/- who are working as ASP 5
2.4 Create a pseudo table employee with the same structure as the table emp and insert rows into the table using select clauses. 6
2.5 Select employee name, job from the emp table 6
2.6 Delete only those who are working as lecturer 6
2.7 List the records in the emp table orderby salary in ascending order 7
2.8 List the records in the emp table order by salary in descending order . . . 7
2.9 Display only those employees whose deptno is 1 7
2.10 Display deptno from the table employee avoiding the duplicated values . . . 8

3 To implement DCL and TCL Commands 9
3.1 DCL Commands 9
3.1.1 Develop a query to grant all privileges of employees table into departments table 9
3.1.2 Develop a query to grant some privileges of employees table into department table 9
3.1.3 Develop a query to revoke all privileges of employees table from departments table 9
3.2 TCL commands 9
3.2.1 Develop a query to revoke some privileges of employees table from departments table 9
3.2.2 Write a query to implement the save point 9
3.2.3 Write a query to implement the rollback 10
3.2.4 Write a query to implement the commit 10

4 Implementation of Cursor 11
4.1 Declare Cursor 11
4.2 Open Cursor 11
4.3 Fetch cursor 11
4.4 Close cursor 11

4.5 Demo 11

5 Implementation of Triggers **13**
5.0.1 table for demo 13
5.1 Create Trigger 13
5.2 Show/List Triggers MySQL 14
5.3 Drop Triggers 14
5.4 before insert trigger 14
5.5 after insert trigger 15

6 Implementation of Aggregate Functions **16**
6.0.1 prepare database 16
6.1 count function 16
6.2 distinct keyword 16
6.3 min function 17
6.4 max function 17
6.5 sum function 17
6.6 avg function 17

7 Procedures **18**
7.1 create procedures 18
7.2 delete procedure 18

8 Functions **19**
8.1 function to find factorial of a number 19
8.2 drop function 19

9 Subquery MySQL **20**
9.1 simple select operation 20
9.2 Subquery with comparision operator 20
9.3 Subquery with IN and NOT IN operator 21

10 Demonstrate an example of ER-Diagram and its relational database schema **22**
10.1 Example for ER-diagram 22
10.2 Relation Database schema 24
10.2.1 Tools used in creating this practical(pdf) 27

1 To create a DDL to perform creation of table, alter, modify and drop column

1.1 The Create Table Command

Syntax:

```
Create table <table name> (<col1> <datatype>(<size>), <col2> <datatype> (<size>));
-- demo
MariaDB [csb]> create table emptemp(empno int primary key, ename char(10));
Query OK, 0 rows affected (0.192 sec)

MariaDB [csb]> describe emptemp;
+-------+----------+------+-----+---------+-------+
| Field | Type     | Null | Key | Default | Extra |
+-------+----------+------+-----+---------+-------+
| empno | int(11)  | NO   | PRI | NULL    |       |
| ename | char(10) | YES  |     | NULL    |       |
+-------+----------+------+-----+---------+-------+
2 rows in set (0.002 sec)
```

1.2 Modify the structure of table

1.2.1 Add new column

Syntax:

```
Alter table <tablename> add(<new col><datatype>(size), <newcol>datatype(size));
--demo
MariaDB [csb]> alter table emptemp add(sal numeric(7, 2));
Query OK, 0 rows affected (0.064 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [csb]> describe emptemp;
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| empno | int(11)      | NO   | PRI | NULL    |       |
| ename | char(10)     | YES  |     | NULL    |       |
| sal   | decimal(7,2) | YES  |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
3 rows in set (0.002 sec)
```

1.3 Dropping a column from a table

Syntax:

```
Alter table <tablename> drop column <col>;
--demo
MariaDB [csb]> alter table emptemp drop column sal;
Query OK, 0 rows affected (0.044 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [csb]> describe emptemp;
+-------+----------+------+-----+---------+-------+
| Field | Type     | Null | Key | Default | Extra |
+-------+----------+------+-----+---------+-------+
```

```
| empno | int(11)  | NO   | PRI | NULL    |       |
| ename | char(10) | YES  |     | NULL    |       |
+-------+----------+------+-----+---------+-------+
2 rows in set (0.002 sec)
```

1.4 Modify existing column

Syntax:

```
Alter table <tablename> modify <col> <datatype>(<size>);
--demo
MariaDB [csb]> alter table emptemp modify ename varchar(15);
Query OK, 0 rows affected (0.370 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [csb]> describe emptemp;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| empno | int(11)     | NO   | PRI | NULL    |       |
| ename | varchar(15) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.001 sec)
```

1.5 Renaming the table

Syntax:

```
rename table <old_table_name> to <new_table_name>;
--demo
MariaDB [csb]> rename table emptemp to emptmp;
Query OK, 0 rows affected (0.068 sec)

MariaDB [csb]> describe emptmp;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| empno | int(11)     | NO   | PRI | NULL    |       |
| ename | varchar(15) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.001 sec)
```

1.6 Destroying table

Syntax:

```
Drop table <tablename>;
--demo
MariaDB [csb]> drop table emptmp;
Query OK, 0 rows affected (0.064 sec)

MariaDB [csb]> describe emptmp;
ERROR 1146 (42S02): Table 'csb.emptmp' doesn't exist
```

2 To study various DML commands

2.1 Insert a single record into dept table

```
MariaDB [csb]> insert into dept values(1, 'IT', 'Tholudur');
Query OK, 1 row affected (0.015 sec)

MariaDB [csb]> select * from dept;
+----+-----------+----------+
| ID | Dept_name | Emp_name |
+----+-----------+----------+
|  1 | IT        | Tholudur |
+----+-----------+----------+
1 row in set (0.001 sec)
```

2.2 Insert more than a recored into emp table using a single insert command

```
MariaDB [csb]> insert into emp values
    -> ( 1, 'Mathi', 'AP', 1, 10000 ),
    -> ( 2, 'Arjun', 'ASP', 2, 12000 ),
    -> ( 3, 'Gugan', 'ASP', 1, 12000 );
Query OK, 3 rows affected (0.022 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [csb]> select * from emp;
+-------+-------+------+--------+-------+
| Empno | Ename | Job  | Deptno | Sal   |
+-------+-------+------+--------+-------+
|     1 | Mathi | AP   |      1 | 10000 |
|     2 | Arjun | ASP  |      2 | 12000 |
|     3 | Gugan | ASP  |      1 | 12000 |
+-------+-------+------+--------+-------+
3 rows in set (0.001 sec)
```

2.3 Update the emp table to set the salary of all the employees to Rs 15000/- who are working as ASP

```
MariaDB [csb]> select * from emp;
+-------+-------+------+--------+-------+
| Empno | Ename | Job  | Deptno | Sal   |
+-------+-------+------+--------+-------+
|     1 | Mathi | AP   |      1 | 10000 |
|     2 | Arjun | ASP  |      2 | 12000 |
|     3 | Gugan | ASP  |      1 | 12000 |
+-------+-------+------+--------+-------+
3 rows in set (0.001 sec)

MariaDB [csb]> update emp set Sal=15000 where job='ASP';
Query OK, 2 rows affected (0.049 sec)
Rows matched: 2  Changed: 2  Warnings: 0

MariaDB [csb]> select * from emp;
+-------+-------+------+--------+-------+
```

```
| Empno | Ename | Job  | Deptno | Sal   |
+-------+-------+------+--------+-------+
|     1 | Mathi | AP   |      1 | 10000 |
|     2 | Arjun | ASP  |      2 | 15000 |
|     3 | Gugan | ASP  |      1 | 15000 |
+-------+-------+------+--------+-------+
3 rows in set (0.001 sec)
```

2.4 Create a pseudo table employee with the same structure as the table emp and insert rows into the table using select clauses.

```
MariaDB [csb]> create table employee as select * from emp;
Query OK, 3 rows affected (0.147 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [csb]> desc employee;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| Empno  | int(11)     | NO   |     | NULL    |       |
| Ename  | varchar(32) | YES  |     | NULL    |       |
| Job    | varchar(40) | YES  |     | NULL    |       |
| Deptno | int(11)     | YES  |     | NULL    |       |
| Sal    | int(11)     | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
5 rows in set (0.001 sec)
```

2.5 Select employee name, job from the emp table

```
MariaDB [csb]> select ename, job from emp;
+-------+------+
| ename | job  |
+-------+------+
| Mathi | AP   |
| Arjun | ASP  |
| Gugan | ASP  |
+-------+------+
3 rows in set (0.001 sec)
```

2.6 Delete only those who are working as lecturer

```
MariaDB [csb]> select * from emp;
+-------+--------+------+--------+-------+
| Empno | Ename  | Job  | Deptno | Sal   |
+-------+--------+------+--------+-------+
|     1 | Mathi  | AP   |      1 | 10000 |
|     2 | Arjun  | ASP  |      2 | 15000 |
|     3 | Gugan  | ASP  |      1 | 15000 |
|     6 | Suresh | lect |      1 |  8000 |
+-------+--------+------+--------+-------+
4 rows in set (0.001 sec)

MariaDB [csb]> delete from emp where job="lect";
```

```
Query OK, 1 row affected (0.017 sec)

MariaDB [csb]> select * from emp;
+-------+-------+------+--------+-------+
| Empno | Ename | Job  | Deptno | Sal   |
+-------+-------+------+--------+-------+
|     1 | Mathi | AP   |      1 | 10000 |
|     2 | Arjun | ASP  |      2 | 15000 |
|     3 | Gugan | ASP  |      1 | 15000 |
+-------+-------+------+--------+-------+
3 rows in set (0.001 sec)
```

2.7 List the records in the emp table orderby salary in ascending order

```
MariaDB [csb]> select * from emp order by sal;
+-------+---------+------+--------+-------+
| Empno | Ename   | Job  | Deptno | Sal   |
+-------+---------+------+--------+-------+
|     1 | Mathi   | AP   |      1 | 10000 |
|     5 | Akalya  | AP   |      1 | 10000 |
|     2 | Arjun   | ASP  |      2 | 15000 |
|     3 | Gugan   | ASP  |      1 | 15000 |
|     4 | Karthik | Proj |      2 | 30000 |
+-------+---------+------+--------+-------+
5 rows in set (0.001 sec)
```

2.8 List the records in the emp table order by salary in descending order

```
MariaDB [csb]> select * from emp order by sal desc;
+-------+---------+------+--------+-------+
| Empno | Ename   | Job  | Deptno | Sal   |
+-------+---------+------+--------+-------+
|     4 | Karthik | Proj |      2 | 30000 |
|     2 | Arjun   | ASP  |      2 | 15000 |
|     3 | Gugan   | ASP  |      1 | 15000 |
|     1 | Mathi   | AP   |      1 | 10000 |
|     5 | Akalya  | AP   |      1 | 10000 |
+-------+---------+------+--------+-------+
5 rows in set (0.001 sec)
```

2.9 Display only those employees whose deptno is 1

```
MariaDB [csb]> select * from emp where Deptno=1;
+-------+--------+------+--------+-------+
| Empno | Ename  | Job  | Deptno | Sal   |
+-------+--------+------+--------+-------+
|     1 | Mathi  | AP   |      1 | 10000 |
|     3 | Gugan  | ASP  |      1 | 15000 |
|     5 | Akalya | AP   |      1 | 10000 |
+-------+--------+------+--------+-------+
3 rows in set (0.001 sec)
```

2.10 Display deptno from the table employee avoiding the duplicated values

```
MariaDB [csb]> select distinct Deptno from emp;
+--------+
| Deptno |
+--------+
|      1 |
|      2 |
+--------+
2 rows in set (0.001 sec)
```

3 To implement DCL and TCL Commands

3.1 DCL Commands

3.1.1 Develop a query to grant all privileges of employees table into departments table

```
MariaDB [csb]> create table departments
    -> (dept_no int primary key, dept_name varchar(24), dept_location varchar(32));
Query OK, 0 rows affected (0.166 sec)

MariaDB [csb]> create table employees
    -> (emp_id int primary key, emp_name varchar(24), emp_salary numeric(10, 2));
Query OK, 0 rows affected (0.134 sec)

MariaDB [csb]> Grant all on employees to departments;
Grant Succeeded
```

3.1.2 Develop a query to grant some privileges of employees table into department table

```
MariaDB [csb]> grant select, update, insert on departments to departments with grant option;
Grant succeeded
```

3.1.3 Develop a query to revoke all privileges of employees table from departments table

```
MariaDB [csb]> revoke all on employees from departments;
Revoke succeeded
```

3.2 TCL commands

3.2.1 Develop a query to revoke some privileges of employees table from departments table

```
MariaDB [csb]> revoke select, update, insert on departments from departments;
Revoke succeeded
```

3.2.2 Write a query to implement the save point

```
MariaDB [csb]> savepoint s1;
Query OK, 0 rows affected (0.000 sec)

MariaDB [csb]> select * from emp;
+-------+---------+------+--------+-------+
| Empno | Ename   | Job  | Deptno | Sal   |
+-------+---------+------+--------+-------+
|     1 | Mathi   | AP   |      1 | 10000 |
|     2 | Arjun   | ASP  |      2 | 15000 |
|     3 | Gugan   | ASP  |      1 | 15000 |
|     4 | Karthik | Proj |      2 | 30000 |
|     5 | Akalya  | AP   |      1 | 10000 |
+-------+---------+------+--------+-------+
5 rows in set (0.000 sec)
```

```
MariaDB [csb]> insert into emp values(6, 'Darpan', 'Proj', 1, 28000);
Query OK, 1 row affected (0.030 sec)

MariaDB [csb]> select * from emp;
+-------+---------+------+--------+-------+
| Empno | Ename   | Job  | Deptno | Sal   |
+-------+---------+------+--------+-------+
|     1 | Mathi   | AP   |      1 | 10000 |
|     2 | Arjun   | ASP  |      2 | 15000 |
|     3 | Gugan   | ASP  |      1 | 15000 |
|     4 | Karthik | Proj |      2 | 30000 |
|     5 | Akalya  | AP   |      1 | 10000 |
|     6 | Darpan  | Proj |      1 | 28000 |
+-------+---------+------+--------+-------+
6 rows in set (0.000 sec)
```

3.2.3 Write a query to implement the rollback

```
MariaDB [csb]> savepoint s1;
Query OK, 0 rows affected (0.000 sec)

MariaDB [csb]> select * from emp;
+-------+---------+------+--------+-------+
| Empno | Ename   | Job  | Deptno | Sal   |
+-------+---------+------+--------+-------+
|     1 | Mathi   | AP   |      1 | 10000 |
|     2 | Arjun   | ASP  |      2 | 15000 |
|     3 | Gugan   | ASP  |      1 | 15000 |
|     4 | Karthik | Proj |      2 | 30000 |
|     5 | Akalya  | AP   |      1 | 10000 |
+-------+---------+------+--------+-------+
5 rows in set (0.000 sec)
```

3.2.4 Write a query to implement the commit

```
MariaDB [csb]> commit;
Query OK, 0 rows affected (0.000 sec)
```

4 Implementation of Cursor

this is implementation of cursor in mysql

4.1 Declare Cursor

Syntax:

```
declare <cursor name> cursor for select <statement>;
```

4.2 Open Cursor

Syntax:

```
open <cursor name>;
```

4.3 Fetch cursor

Syntax:

```
fetch [next[from]] <curor name> into <variable list>;
```

4.4 Close cursor

Syntax:

```
close <cursor name>;
```

4.5 Demo

Following is the *demo* for implementation of cursor in mysql

```
-- table to operate on
MariaDB [csb]> select * from cursordemo;
+----+----------+--------+
| id | name     | course |
+----+----------+--------+
|  1 | Shristee | MCA    |
|  2 | Ajay     | BCA    |
|  3 | Shweta   | MCA    |
|  4 | Dolly    | BCA    |
|  5 | Heena    | MCA    |
|  6 | Kiran    | BCA    |
|  7 | Sonal    | MCA    |
|  8 | Dimple   | BCA    |
|  9 | Shyam    | MCA    |
| 10 | Mohit    | BCA    |
+----+----------+--------+
10 rows in set (0.001 sec)

-- creation of cursor
MariaDB [csb]> create procedure list_name (inout name_list varchar(4000))
    -> begin
    -> declare is_done integer default 0;
    -> declare s_name varchar(100) default "";
    -> declare stud_cursor cursor for
    -> select name from cursordemo;
```

```
    -> declare continue handler for not found set is_done=1;
    -> open stud_cursor;
    -> get_list: loop
    -> fetch stud_cursor into s_name;
    -> if is_done=1 then
    -> leave get_list;
    -> end if;
    -> set name_list=concat(s_name, "; ", name_list);
    -> end loop get_list;
    -> close stud_cursor;
    -> end$$
Query OK, 0 rows affected (0.414 sec)

-- call the cursor
MariaDB [csb]> set @name_list ="";
Query OK, 0 rows affected (0.000 sec)

MariaDB [csb]> call list_name(@name_list);
Query OK, 0 rows affected (0.225 sec)

MariaDB [csb]> select @name_list;
+-------------------------------------------------------------------------------+
| @name_list                                                                    |
+-------------------------------------------------------------------------------+
| Mohit; Shyam; Dimple; Sonal; Kiran; Heena; Dolly; Shweta; Ajay; Shristee;  |
+-------------------------------------------------------------------------------+
1 row in set (0.000 sec)
```

5 Implementation of Triggers

this is implementation of triggers in mysql. MySQL doesn't support statement-level triggers, only support row-level triggers

5.0.1 table for demo

```
MariaDB [csb]> select * from employee;
+---------+------------+--------------+---------------+
| name    | occupation | working_date | working_hours |
+---------+------------+--------------+---------------+
| Robin   | Scientist  | 2020-10-04   | 12            |
| Warner  | Engineer   | 2020-10-04   | 10            |
| Peter   | Actor      | 2020-10-04   | 13            |
| Marco   | Doctor     | 2020-10-04   | 14            |
| Brayden | Teacher    | 2020-10-04   | 12            |
| Antonio | Business   | 2020-10-04   | 11            |
+---------+------------+--------------+---------------+
6 rows in set (0.001 sec)
```

5.1 Create Trigger

```
MariaDB [csb]> create trigger before_insert_empworkinghours
    -> before insert on employee for each row
    -> begin
    -> if new.working_hours < 0 then set new.working_hours = 0;
    -> end if;
    -> end //;
Query OK, 0 rows affected (0.118 sec)

-- invoke the trigger
MariaDB [csb]> insert into employee values('Markus', 'Former', '2020-10-08', 14);
Query OK, 1 row affected (0.090 sec)

MariaDB [csb]> insert into employee values('Alexander', 'Actor', '2020-12-10', -13);
Query OK, 1 row affected (0.011 sec)

MariaDB [csb]> select * from employee;
+-----------+------------+--------------+---------------+
| name      | occupation | working_date | working_hours |
+-----------+------------+--------------+---------------+
| Robin     | Scientist  | 2020-10-04   | 12            |
| Warner    | Engineer   | 2020-10-04   | 10            |
| Peter     | Actor      | 2020-10-04   | 13            |
| Marco     | Doctor     | 2020-10-04   | 14            |
| Brayden   | Teacher    | 2020-10-04   | 12            |
| Antonio   | Business   | 2020-10-04   | 11            |
| Markus    | Former     | 2020-10-08   | 14            |
| Alexander | Actor      | 2020-12-10   | 0             |
+-----------+------------+--------------+---------------+
8 rows in set (0.001 sec)

-- in above output pay attention to last person's working_hours, -ve value is set to 0
```

5.2 Show/List Triggers MySQL

```
MariaDB [(none)]> show triggers;
ERROR 1046 (3D000): No database selected
MariaDB [(none)]> use csb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [csb]> show triggers;
+------------------------------+--------+----------+-----------------------------------------------------------------
| Trigger                      | Event  | Table    | Statement
+------------------------------+--------+----------+-----------------------------------------------------------------
| before_insert_empworkinghours | INSERT | employee | begin
if new.working_hours < 0 then set new.working_hours = 0;
end if;
end | BEFORE | 2021-01-21 22:08:33.85 | STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER
+------------------------------+--------+----------+-----------------------------------------------------------------
1 row in set (0.073 sec)
-- you can also list out triggers with patterns
```

5.3 Drop Triggers

```
MariaDB [csb]> drop trigger before_insert_empworkinghours;
Query OK, 0 rows affected (0.001 sec)

MariaDB [csb]> drop trigger before_insert_empworkinghours;
ERROR 1360 (HY000): Trigger does not exist

-- if trigger doesn't exist in db this command will throw an error
-- you can use 'if exists' to check for existance

MariaDB [csb]> drop trigger if exists before_insert_empworkinghours;
Query OK, 0 rows affected, 1 warning (0.000 sec)
```

5.4 before insert trigger

```
MariaDB [csb]> delimiter //;
MariaDB [csb]> create trigger before_insert_occupation
    -> before insert on employee for each row
    -> begin
    -> if new.occupation = 'Scientist' then set new.occupation = 'Doctor';
    -> end if;
    -> end //;
Query OK, 0 rows affected (0.078 sec)

MariaDB [csb]> insert into employee values('Daniel', 'Scientist', '2017
-05-23', 8);
Query OK, 1 row affected (0.113 sec)

MariaDB [csb]> select * from employee;
+-----------+------------+--------------+---------------+
| name      | occupation | working_date | working_hours |
+-----------+------------+--------------+---------------+
```

```
| Robin     | Scientist  | 2020-10-04   | 12            |
| Warner    | Engineer   | 2020-10-04   | 10            |
| Peter     | Actor      | 2020-10-04   | 13            |
| Marco     | Doctor     | 2020-10-04   | 14            |
| Brayden   | Teacher    | 2020-10-04   | 12            |
| Antonio   | Business   | 2020-10-04   | 11            |
| Markus    | Former     | 2020-10-08   | 14            |
| Alexander | Actor      | 2020-12-10   | 0             |
| Daniel    | Doctor     | 2017-05-23   | 8             |
+-----------+------------+--------------+---------------+
9 rows in set (0.001 sec)
-- notice the last tuple, 'Daniel is Doctor not Scientist'
```

5.5 after insert trigger

```
MariaDB [csb]> create table employee_detail(name varchar(45), occupatio
n varchar(35), working_date date, working_hours varchar(10), last_inser
ted time);
Query OK, 0 rows affected (0.225 sec)

MariaDB [csb]> select * from employee_detail;
Empty set (0.001 sec)

MariaDB [csb]> delimiter //;
MariaDB [csb]> create trigger after_insert_details
    -> after insert on employee for each row
    -> begin
    -> insert into employee_detail values(
    -> new.name,
    -> new.occupation,
    -> new.working_date,
    -> new.working_hours,
    -> curtime());
    -> end//;
Query OK, 0 rows affected (0.135 sec)

MariaDB [csb]> insert into employee values('Jacob', 'Zoologist', '2019-07-28', 11);
Query OK, 1 row affected (0.016 sec)

MariaDB [csb]> select * from employee_detail;
+-------+------------+--------------+---------------+---------------+
| name  | occupation | working_date | working_hours | last_inserted |
+-------+------------+--------------+---------------+---------------+
| Jacob | Zoologist  | 2019-07-28   | 11            | 22:32:04      |
+-------+------------+--------------+---------------+---------------+
1 row in set (0.001 sec)
-- you may notice that insert into employee also filled employee_detail after filling up employee table
```

In same way there's also `before update`, `after update`, `before delete` & `after delete` trigger operations supported by MySQL.

6 Implementation of Aggregate Functions

6.0.1 prepare database

```
MariaDB [csb]> create table movierent(
    -> ref_no int primary key,
    -> trans_date date,
    -> return_date date,
    -> membership_number int,
    -> movie_id int,
    -> movie_returned int);
Query OK, 0 rows affected (0.122 sec)

-- data inserted in table movierent

MariaDB [csb]> select * from movierent;
+--------+------------+-------------+------------------+----------+----------------+
| ref_no | trans_date | return_date | membership_number | movie_id | movie_returned |
+--------+------------+-------------+------------------+----------+----------------+
|     11 | 2020-06-20 | NULL        |                1 |        1 |              0 |
|     12 | 2020-06-21 | 2020-06-22  |                1 |        2 |              0 |
|     13 | 2020-06-22 | 2020-06-23  |                3 |        2 |              0 |
|     14 | 2020-06-23 | 2020-06-24  |                2 |        2 |              0 |
|     15 | 2020-06-24 | NULL        |                3 |        3 |              0 |
+--------+------------+-------------+------------------+----------+----------------+
5 rows in set (0.000 sec)
```

6.1 count function

```
MariaDB [csb]> select count(movie_id) from movierent where movie_id=2;
+-----------------+
| count(movie_id) |
+-----------------+
|               3 |
+-----------------+
1 row in set (0.001 sec)
```

6.2 distinct keyword

To check the concept of `distinct`, lets execute a simple query first:

```
MariaDB [csb]> select movie_id from movierent;
+----------+
| movie_id |
+----------+
|        1 |
|        2 |
|        2 |
|        2 |
|        3 |
+----------+
5 rows in set (0.001 sec)
```

Now, let's execute same query with `distinct` keyword

```
MariaDB [csb]> select distinct movie_id from movierent;
+----------+
| movie_id |
+----------+
|        1 |
|        2 |
|        3 |
+----------+
3 rows in set (0.066 sec)
```

6.3 min function

```
MariaDB [csb]> select min(trans_date) from movierent;
+-----------------+
| min(trans_date) |
+-----------------+
| 2020-06-20      |
+-----------------+
1 row in set (0.029 sec)
```

6.4 max function

```
MariaDB [csb]> select max(return_date) from movierent;
+------------------+
| max(return_date) |
+------------------+
| 2020-06-24       |
+------------------+
1 row in set (0.001 sec)
```

6.5 sum function

```
MariaDB [csb]> select sum(membership_number) from movierent;
+------------------------+
| sum(membership_number) |
+------------------------+
|                     10 |
+------------------------+
1 row in set (0.001 sec)
```

6.6 avg function

```
MariaDB [csb]> select avg(membership_number) from movierent;
+------------------------+
| avg(membership_number) |
+------------------------+
|                 2.0000 |
+------------------------+
1 row in set (0.022 sec)
```

7 Procedures

7.1 create procedures

```
MariaDB [csb]> create table emp1(id numeric(3), first_name varchar(20))
;
Query OK, 0 rows affected (0.107 sec)

MariaDB [csb]> insert into emp1 values(101, 'Nithya');
Query OK, 1 row affected (0.018 sec)

MariaDB [csb]> insert into emp1 values(102, 'Maya');
Query OK, 1 row affected (0.010 sec)

MariaDB [csb]> select * from emp1;
+------+------------+
| id   | first_name |
+------+------------+
|  101 | Nithya     |
|  102 | Maya       |
+------+------------+
2 rows in set (0.001 sec)

-- create procedure
MariaDB [csb]> delimiter //;
MariaDB [csb]> create procedure get_persons()
    -> begin
    -> select * from emp1;
    -> end //;
Query OK, 0 rows affected (0.072 sec)

-- call procedure
MariaDB [csb]> call get_persons();
+------+------------+
| id   | first_name |
+------+------------+
|  101 | Nithya     |
|  102 | Maya       |
+------+------------+
2 rows in set (0.001 sec)

Query OK, 0 rows affected (0.001 sec)
```

7.2 delete procedure

```
MariaDB [csb]> drop procedure if exists get_persons;
Query OK, 0 rows affected (0.093 sec)
-- if exists is not necessary
```

8 Functions

8.1 function to find factorial of a number

```
-- create function 'factorial'
MariaDB [csb]> delimiter //;
MariaDB [csb]> create function factorial(num int)
    -> returns int(12);
    -> begin
    -> declare factorial int;
    -> set factorial = num;
    -> if num <= 0 then
    -> return 1;
    -> end if;
    ->
    -> bucle: loop
    -> set num=num-1;
    -> if num < 1 then
    -> leave bucle;
    -> end if;
    -> set factorial = factorial * num;
    -> end loop bucle;
    -> return factorial;
    -> end//;
Query OK, 0 rows affected (0.071 sec)

delimiter ;  -- set to default delimiter

-- execute the function
MariaDB [csb]> select factorial(5);
+--------------+
| factorial(5) |
+--------------+
|          120 |
+--------------+
1 row in set (0.058 sec)

MariaDB [csb]> select factorial(10);
+---------------+
| factorial(10) |
+---------------+
|       3628800 |
+---------------+
1 row in set (0.001 sec)
```

8.2 drop function

```
MariaDB [csb]> drop function if exists factorial;
Query OK, 0 rows affected (0.074 sec)
```

9 Subquery MySQL

9.1 simple select operation

```
-- emp table already created and filled before
MariaDB [csb]> select * from emp ;
+-------+---------+------+--------+-------+
| Empno | Ename   | Job  | Deptno | Sal   |
+-------+---------+------+--------+-------+
|     1 | Mathi   | AP   |      1 | 10000 |
|     2 | Arjun   | ASP  |      2 | 15000 |
|     3 | Gugan   | ASP  |      1 | 15000 |
|     4 | Karthik | Proj |      2 | 30000 |
|     5 | Akalya  | AP   |      1 | 10000 |
+-------+---------+------+--------+-------+
5 rows in set (0.001 sec)

-- simple select subquery execution
MariaDB [csb]> select Ename,Job,Deptno from emp
    -> where Empno in (select Empno from emp);
+---------+------+--------+
| Ename   | Job  | Deptno |
+---------+------+--------+
| Mathi   | AP   |      1 |
| Arjun   | ASP  |      2 |
| Gugan   | ASP  |      1 |
| Karthik | Proj |      2 |
| Akalya  | AP   |      1 |
+---------+------+--------+
5 rows in set (0.145 sec)
```

9.2 Subquery with comparision operator

```
MariaDB [csb]> select Ename,Job,Deptno from emp
    -> where Empno in ( select Empno from emp where sal >= 15000);
+---------+------+--------+
| Ename   | Job  | Deptno |
+---------+------+--------+
| Arjun   | ASP  |      2 |
| Gugan   | ASP  |      1 |
| Karthik | Proj |      2 |
+---------+------+--------+
3 rows in set (0.001 sec)

-- or

MariaDB [csb]> select Ename,Job,Deptno from emp where sal = (select max(sal) from emp);
+---------+------+--------+
| Ename   | Job  | Deptno |
+---------+------+--------+
| Karthik | Proj |      2 |
+---------+------+--------+
1 row in set (0.033 sec)
```

9.3 Subquery with IN and NOT IN operator

```
-- tables
MariaDB [csb]> select * from student1;
+----+------+-------------+---------+
| id | name | email       | city    |
+----+------+-------------+---------+
|  1 | abc  | abc@xyz.com | cityabc |
|  2 | bcd  | bcd@xyz.com | citybcd |
|  3 | cde  | cde@xyz.com | citycde |
|  4 | def  | def@xyz.com | citydef |
|  5 | efg  | efg@xyz.com | cityefg |
+----+------+-------------+---------+
5 rows in set (0.001 sec)

MariaDB [csb]> select * from student2;
+----+------+-------------+---------+
| id | name | email       | city    |
+----+------+-------------+---------+
|  1 | fgh  | fgh@xyz.com | cityabc |
|  2 | cde  | cde@xyz.com | citydef |
|  3 | abc  | abc@xyz.com | citybcd |
|  4 | ghi  | ghi@xyz.com | cityefg |
|  5 | hij  | hij@xyz.com | citydef |
+----+------+-------------+---------+
5 rows in set (0.001 sec)

-- not in operator
MariaDB [csb]> select name, city from student1
    -> where city not in
    -> ( select city from student2 where city='citydef');
+------+---------+
| name | city    |
+------+---------+
| abc  | cityabc |
| bcd  | citybcd |
| cde  | citycde |
| efg  | cityefg |
+------+---------+
4 rows in set (0.026 sec)
```

Some other operators for `subquery` are: `from`, `exists`, `not exists`, `all`, `any`, `some` etc.

10 Demonstrate an example of ER-Diagram and its relational database schema

10.1 Example for ER-diagram

A record company XYZ has decided to store information about musicians who perform on all its albums(as well as other company data) in a database. Following are the conditions/constraints:

- Each musician that records at XYZ has an *ssn*, a *name*, an *address*, and a *phone number*.
- Each instrument used ins the songs recoreded at XYZ has a unique *identification* number, a *name*(e.g., guitar, flute etc.) and a *musical key*.
- Each albumn recorded on the XYZ label has a unique *identification number*, a *title*, a *copyright date*, a *format* and an *album identifier*.
- Each song recoreded at XYZ has a *title* and an *author*.
- Each musician may play several instruments and a given instrument may be played by several musicians.
- Each album has a number of songs on it, but no song may appear on more than one album.
- Each song is performed by one or more musicians, and a musician my perform a number of songs.
- Each album has exactly one musician who acts as its producer. A musician can produce several albums.

Considering above conditions/constraints we'll obtain a *ER-Diagram* which will look something like this:

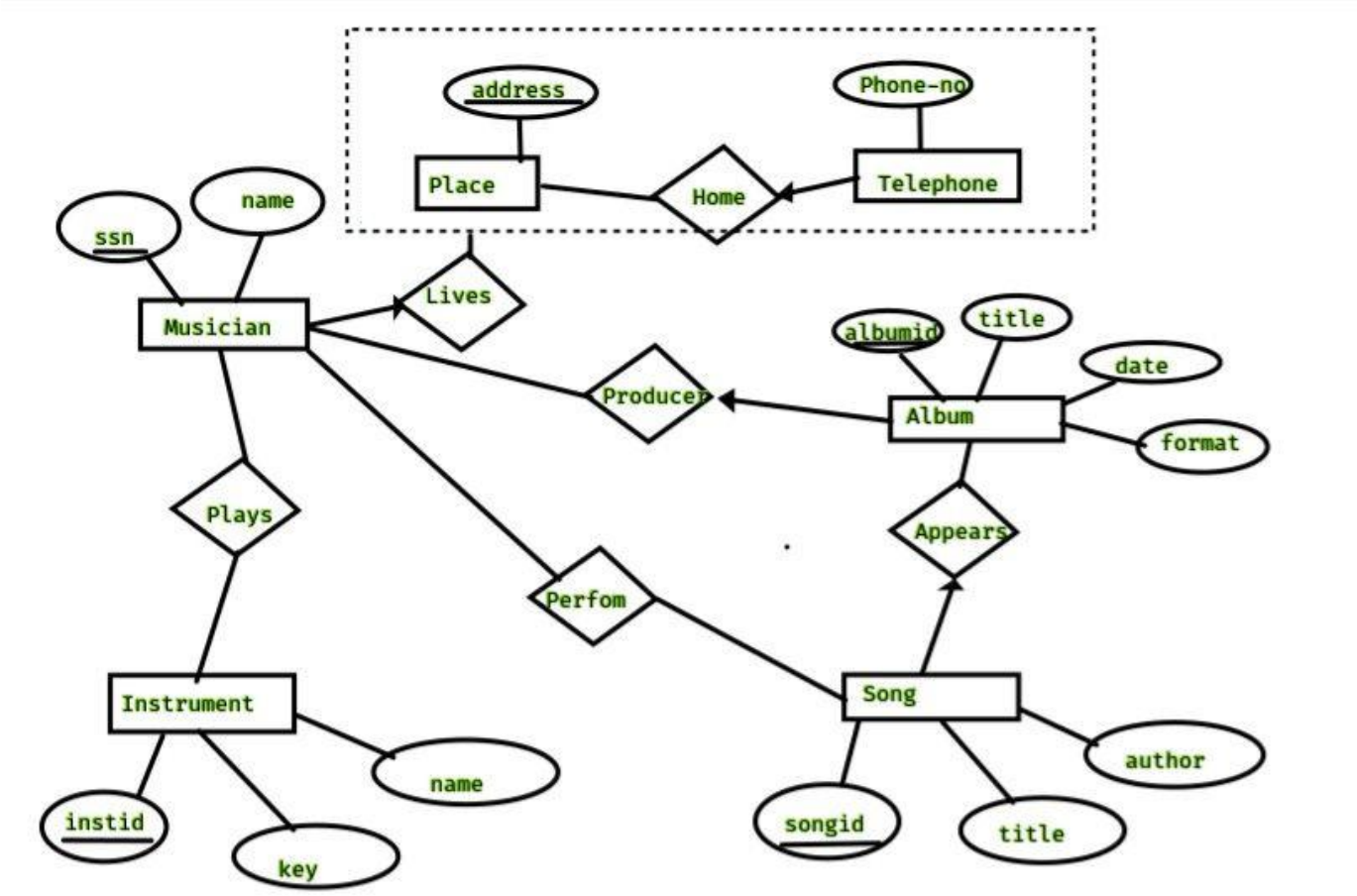

Figure 1: musician er-diagram

Now, let's convert the above *ER-Diagram* to *Relational Database*. I'll perform this operation on `mysql`.

10.2 Relation Database schema

Creation of *relational database schema* for the above *er-diagram*.

```
-- creation of Musician
MariaDB [csb]> create table Musician(ssn varchar(10), name varchar(30),
 primary key(ssn));
Query OK, 0 rows affected (0.175 sec)

MariaDB [csb]> desc Musician;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| ssn   | varchar(10) | NO   | PRI | NULL    |       |
| name  | varchar(30) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.002 sec)

-- creation of Insturment
-- note: changed name of attriubute key to instkey
MariaDB [csb]> create table Instrument(instid varchar(10) primary key,
name varchar(30), instkey varchar(5));
Query OK, 0 rows affected (0.122 sec)

MariaDB [csb]> desc Instrument;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| instid  | varchar(10) | NO   | PRI | NULL    |       |
| name    | varchar(30) | YES  |     | NULL    |       |
| instkey | varchar(5)  | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.002 sec)

-- creating Plays relation
MariaDB [csb]> create table Plays
    -> (ssn varchar(10), instid varchar(10),
    -> primary key(ssn, instid),
    -> foreign key(ssn) references Musician(ssn),
    -> foreign key(instid) references Instrument(instid));
Query OK, 0 rows affected (0.180 sec)

MariaDB [csb]> desc Plays;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| ssn    | varchar(10) | NO   | PRI | NULL    |       |
| instid | varchar(10) | NO   | PRI | NULL    |       |
+--------+-------------+------+-----+---------+-------+
2 rows in set (0.002 sec)

-- song 'Appears' relation
MariaDB [csb]> create table Song(
    -> songid int primary key, title varchar(30), author varchar(20));
```

```
Query OK, 0 rows affected (0.124 sec)

-- create Song table;
MariaDB [csb]> desc Song;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| songid | int(11)     | NO   | PRI | NULL    |       |
| title  | varchar(30) | YES  |     | NULL    |       |
| author | varchar(20) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.002 sec)

-- create Perform relation
MariaDB [csb]> create table Perform
    -> (ssn varchar(10), songid int, performdate date,
    -> primary key(ssn, songid),
    -> foreign key(ssn) references Musician(ssn),
    -> foreign key(songid) references Song(songid));
Query OK, 0 rows affected (0.145 sec)

MariaDB [csb]> desc Perform;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| ssn         | varchar(10) | NO   | PRI | NULL    |       |
| songid      | int(11)     | NO   | PRI | NULL    |       |
| performdate | date        | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.002 sec)

-- create Album table
MariaDB [csb]> create table Album
    -> (albumid varchar(10), title varchar(30),
    -> releasedate date, format varchar(12), primary key(albumid));
Query OK, 0 rows affected (0.125 sec)

MariaDB [csb]> desc Album;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| albumid     | varchar(10) | NO   | PRI | NULL    |       |
| title       | varchar(30) | YES  |     | NULL    |       |
| releasedate | date        | YES  |     | NULL    |       |
| format      | varchar(12) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.001 sec)

-- create Appears relation
MariaDB [csb]> create table Appears
    -> (albumid varchar(10), songid int, so ngnumberinalbum int,
    -> primary key(albumid, songid),
    -> foreign key(albumid) references Album(albumid),
    -> foreign key(songid) references Song(songid));
Query OK, 0 rows affected (0.140 sec)
```

```
MariaDB [csb]> desc Appears;
+------------------+-------------+------+-----+---------+-------+
| Field            | Type        | Null | Key | Default | Extra |
+------------------+-------------+------+-----+---------+-------+
| albumid          | varchar(10) | NO   | PRI | NULL    |       |
| songid           | int(11)     | NO   | PRI | NULL    |       |
| songnumberinalbum | int(11)     | YES  |     | NULL    |       |
+------------------+-------------+------+-----+---------+-------+
3 rows in set (0.002 sec)

-- create 'Producer' relation
MariaDB [csb]> create table Producer (ssn varchar(10), albumid varchar(10), productionstarted date, primar
) references Musician(ssn), foreign key(albumid) references Album(album
id));
Query OK, 0 rows affected (0.274 sec)

MariaDB [csb]> desc Producer;
+------------------+-------------+------+-----+---------+-------+
| Field            | Type        | Null | Key | Default | Extra |
+------------------+-------------+------+-----+---------+-------+
| ssn              | varchar(10) | NO   | PRI | NULL    |       |
| albumid          | varchar(10) | NO   | PRI | NULL    |       |
| productionstarted | date        | YES  |     | NULL    |       |
+------------------+-------------+------+-----+---------+-------+
3 rows in set (0.002 sec)

-- Place table
MariaDB [csb]> create table Place(address varchar(30) primary key);
Query OK, 0 rows affected (0.150 sec)

MariaDB [csb]> desc Place;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| address | varchar(30) | NO   | PRI | NULL    |       |
+---------+-------------+------+-----+---------+-------+
1 row in set (0.002 sec)

-- Telephone table
MariaDB [csb]> create table Telephone(phoner_no int primary key);
Query OK, 0 rows affected (0.122 sec)

MariaDB [csb]> desc Telephone;
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| phoner_no | int(11) | NO   | PRI | NULL    |       |
+-----------+---------+------+-----+---------+-------+
1 row in set (0.002 sec)

-- Home relation
MariaDB [csb]> create table Home
    -> (phone_no int, address varchar(30),
    -> primary key(phone_no),
    -> foreign key(address) references Place(address));
Query OK, 0 rows affected (0.189 sec)
```

```
MariaDB [csb]> desc Home;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| phone_no | int(11)     | NO   | PRI | NULL    |       |
| address  | varchar(30) | YES  | MUL | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.002 sec)

-- relation Lives
MariaDB [csb]> create table Lives
    -> (ssn varchar(10), phone_no int,
    -> foreign key(phone_no) references Home(phone_no));
Query OK, 0 rows affected (0.147 sec)

MariaDB [csb]> desc Lives;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| ssn      | varchar(10) | YES  |     | NULL    |       |
| phone_no | int(11)     | YES  | MUL | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.002 sec)
```

10.2.1 Tools used in creating this practical(pdf)

- **OS :** 5.4.85-1-MANJARO
- **WM :** DWM
- **Pdf(markup) convertor:** Pandoc(2.11.2)
- **Pdf engine :** xelatex
- **Source File Format :** Markdown(md)
- **Text Editor :** Neovim-nightly(v0.5.0-dev+1000-g84d08358b)
- **DB used:** Mariadb(Ver 15.1 Distrib 10.5.8-MariaDB)

--* THE END --*