

Title: Stored procedure and Views

Objective:

- To be familiar with concept of “stored procedure” and implement it.
- To be familiar with concept of views and analyze the role of views for data security.

Problem:

Stored Procedure

1) Create any database.

```
MariaDB [(none)]> create database views;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| customer_db |
| eemc_db |
| information_schema |
| joinlab |
| labfour |
| mysql |
| performance_schema |
| phpmyadmin |
| sak_db |
| sakwheelsdb |
| test |
| views |
+-----+
12 rows in set (0.030 sec)

MariaDB [(none)]> use views;
Database changed
```

2) Create two tables with following columns where underlined attributes represent primary key.

Employee(emp_id,emp_name,postion,salary,dept_id)

Department(dep_id,dept_name,location,budget)

```
MariaDB [views]> create table department (dept_id int primary key ,dept_name varchar(30), location varchar(30), budget int);
Query OK, 0 rows affected (0.656 sec)
```

LAB-6 DBMS (Stored procedure and Views)

```
MariaDB [views]> describe department;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| dept_id | int(11)   | NO   | PRI | NULL    |       |
| dept_name | varchar(30) | YES  |     | NULL    |       |
| location | varchar(30) | YES  |     | NULL    |       |
| budget  | int(11)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.033 sec)

MariaDB [views]> create table employee (emp_id int primary key ,emp_name varchar(30),
-> position varchar(30), salary int, dept_id int, foreign key (dept_id) references department (dept_id));
Query OK, 0 rows affected (0.413 sec)

MariaDB [views]> show tables;
+-----+
| Tables_in_views |
+-----+
| department       |
| employee         |
+-----+
2 rows in set (0.001 sec)

MariaDB [views]> describe employee;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_id | int(11)   | NO   | PRI | NULL    |       |
| emp_name | varchar(30) | YES  |     | NULL    |       |
| position | varchar(30) | YES  |     | NULL    |       |
| salary | int(11)   | YES  |     | NULL    |       |
| dept_id | int(11)   | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.006 sec)
```

3) Insert at least 5 rows in each table.

```
MariaDB [views]> insert into department
-> values(1,"Computer","Sanepa",23453);
Query OK, 1 row affected (0.350 sec)

MariaDB [views]> insert into department
-> values(2,"civil","Jhamsikhel",42521);
Query OK, 1 row affected (0.165 sec)

MariaDB [views]> insert into department
-> values(2,"IT","Bhaisipati",53632);
ERROR 1062 (23000): Duplicate entry '2' for key 'PRIMARY'
MariaDB [views]> insert into department
-> values(3,"IT","Bhaisipati",53632);
Query OK, 1 row affected (0.109 sec)

MariaDB [views]> insert into department
-> values(4,"Automobile","Palpa",35626);
Query OK, 1 row affected (0.360 sec)

MariaDB [views]> insert into department
-> values(5,"Aerospace","Pulchok",67221);
Query OK, 1 row affected (0.102 sec)
```

```

MariaDB [views]> insert into employee
-> values(1,"Sakchyaam","Manager",67221,2);
Query OK, 1 row affected (0.358 sec)

MariaDB [views]> insert into employee
-> values(2,"Sujan","Technician",65321,1);
Query OK, 1 row affected (0.088 sec)

MariaDB [views]> insert into employee
-> values(3,"Sujit","Designer",21611,4);
Query OK, 1 row affected (0.096 sec)

MariaDB [views]> insert into employee
-> values(4,"Ravi","Guff Garne",36364,5);
Query OK, 1 row affected (0.083 sec)

MariaDB [views]> insert into employee
-> values(5,"Rame","Toilet Cleaner",57325,1);
Query OK, 1 row affected (0.330 sec)

```

```

MariaDB [views]> select * from employee;
+-----+-----+-----+-----+-----+
| emp_id | emp_name | position | salary | dept_id |
+-----+-----+-----+-----+-----+
| 1 | Sakchyaam | Manager | 67221 | 2 |
| 2 | Sujan | Technician | 65321 | 1 |
| 3 | Sujit | Designer | 21611 | 4 |
| 4 | Ravi | Guff Garne | 36364 | 5 |
| 5 | Rame | Toilet Cleaner | 57325 | 1 |
+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)

```

```

MariaDB [views]> select * from department;
+-----+-----+-----+-----+
| dept_id | dept_name | location | budget |
+-----+-----+-----+-----+
| 1 | Computer | Sanepa | 23453 |
| 2 | civil | Jhamsikhel | 42521 |
| 3 | IT | Bhaisipati | 53632 |
| 4 | Automobile | Palpa | 35626 |
| 5 | Aerospace | Pulchok | 67221 |
+-----+-----+-----+-----+

```

4) Create stored procedure without using parameters to

i) To Find all the information of employee.

ii) To find emp_name, position, salary, dept_name of employee.

5) Execute the above created stored procedure.

```
MariaDB [views]> Delimiter //
MariaDB [views]> Create procedure Sak()
-> Begin
-> select * from employee;
-> end //
Query OK, 0 rows affected (0.306 sec)
```

```
MariaDB [views]> delimiter ;
MariaDB [views]> call Sak();
```

emp_id	emp_name	position	salary	dept_id
1	Sakchyam	Manager	67221	2
2	Sujan	Technician	65321	1
3	Sujit	Designer	21611	4
4	Ravi	Guff Garne	36364	5
5	Rame	Toilet Cleaner	57325	1

6) Create stored procedure to with using parameters

i) To find the information of employee of specified department.

```
MariaDB [views]> Delimiter //
MariaDB [views]> Create procedure emp_deptart(dept varchar(30))
-> Begin
-> select employee.*
-> from employee,department
-> where employee.dept_id = department.dept_id
-> and department.dept_name=dept;
-> end //
Query OK, 0 rows affected (0.246 sec)

MariaDB [views]> delimiter ;
```

```

MariaDB [views]> call emp_deptart("civil");
+-----+-----+-----+-----+-----+
| emp_id | emp_name | position | salary | dept_id |
+-----+-----+-----+-----+-----+
|      1 | Sakchyam | Manager  | 67221  |      2  |
+-----+-----+-----+-----+-----+
1 row in set (0.013 sec)

Query OK, 0 rows affected (0.023 sec)

MariaDB [views]> call emp_deptart("IT");
Empty set (0.001 sec)

Query OK, 0 rows affected (0.001 sec)

MariaDB [views]> call emp_deptart("computer");
+-----+-----+-----+-----+-----+
| emp_id | emp_name | position          | salary | dept_id |
+-----+-----+-----+-----+-----+
|      2 | Sujana   | Technician        | 65321  |      1  |
|      5 | Rame     | Toilet Cleaner    | 57325  |      1  |
+-----+-----+-----+-----+-----+

```

ii) To find the information of employee of specified employee name and employee position.

```

MariaDB [views]> Create procedure emp_d(dept varchar(30), posi varchar(30))
-> Begin
-> select employee.*
-> from employee,department
-> where employee.dept_id = department.dept_id
-> and department.dept_name=dept
-> and employee.position=posi;
-> end //
Query OK, 0 rows affected (0.214 sec)

MariaDB [views]> delimiter ;
MariaDB [views]> call emp_d('sujan','technician');
Empty set (0.001 sec)

Query OK, 0 rows affected (0.002 sec)

MariaDB [views]> call emp_d('Sujan','Technician');
Empty set (0.001 sec)

Query OK, 0 rows affected (0.001 sec)

```

Views

- 1) Create view for display the emp_id ,emp_name, position of employee.
- 2) Create view for display emp_id, emp_name, position, dept_name, location.

```
MariaDB [views]> create view emp_view as
-> select emp_id,emp_name,position from employee;
Query OK, 0 rows affected (0.396 sec)
```

```
MariaDB [views]> select * from emp_view;
```

emp_id	emp_name	position
1	Sakchya	Manager
2	Sujan	Technician
3	Sujit	Designer
4	Ravi	Guff Garne
5	Rame	Toilet Cleaner

5 rows in set (0.002 sec)

```
MariaDB [views]> select * from emp_view
-> where position="manager";
```

emp_id	emp_name	position
1	Sakchya	Manager

1 row in set (0.001 sec)

```
MariaDB [views]> create view emp_view2 as
-> select emp_name,position,dept_name,location
-> from employee natural join department;
Query OK, 0 rows affected (0.099 sec)
```

```
MariaDB [views]> select * from emp_view2;
```

emp_name	position	dept_name	location
Sakchya	Manager	civil	Jhamsikhel
Sujan	Technician	Computer	Sanepa
Sujit	Designer	Automobile	Palpa
Ravi	Guff Garne	Aerospace	Pulchok
Rame	Toilet Cleaner	Computer	Sanepa

5 rows in set (0.002 sec)

Discussion:

In this DBMS Lab, we delved into the practical aspects of Stored Procedures and Views, two essential components in managing and manipulating data within a database.

LAB-6 DBMS (Stored procedure and Views)

In the lab, we learned how to create, modify, and execute stored procedures. By doing so, we gained the ability to encapsulate intricate logic into a single function that we can easily reuse. This not only saves time but also maintains consistency in data manipulation.

Our lab exercises involved creating views to simplify data retrieval. For instance, if we often needed a specific subset of data from a table, creating a view made it convenient to access that information without writing complex queries repeatedly.

Conclusion:

Hence, in this DBMS Lab on Stored Procedures and Views provided a hands-on understanding of optimizing data management. Stored Procedures simplify complex tasks, promoting efficiency and security, while Views offer a clear window to relevant data without altering the original. These skills are key for building robust and streamlined database solutions in real-world applications.