Create database practical1;

Use practical1;

CREATE TABLE Students (student_id INT AUTO_INCREMENT PRIMARY KEY, first_name VARCHAR(50), last_name VARCHAR(50), department VARCHAR(50), age INT);

Insert into students
value(1,”AAA”,”aaa”,”IT”,19),(2,”BBB”,”bbb”,”CS”,20),(3,”CCC”,”ccc”,”ENTC”,21),(4,”DDD”,”ddd”,”MECH”,22),(5,”EEE”,”eee”,”IT”,21);

Select * from students;

DELIMITER //

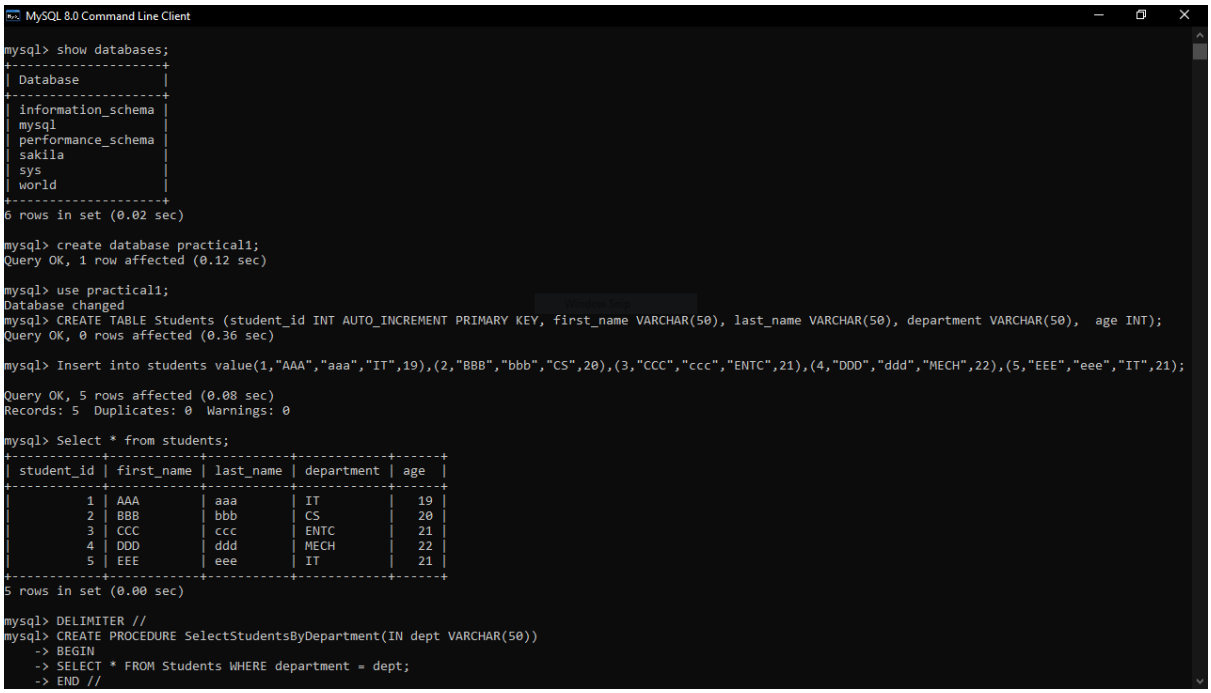
CREATE PROCEDURE SelectStudentsByDepartment(IN dept VARCHAR(50))

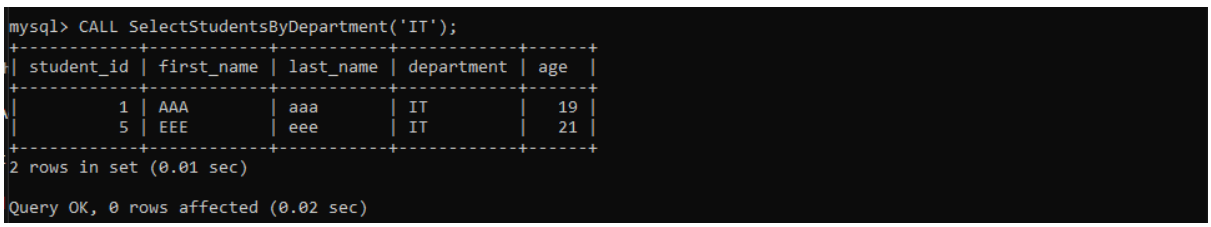
BEGIN

SELECT * FROM Students WHERE department = dept;

END //

CALL SelectStudentsByDepartment('IT');

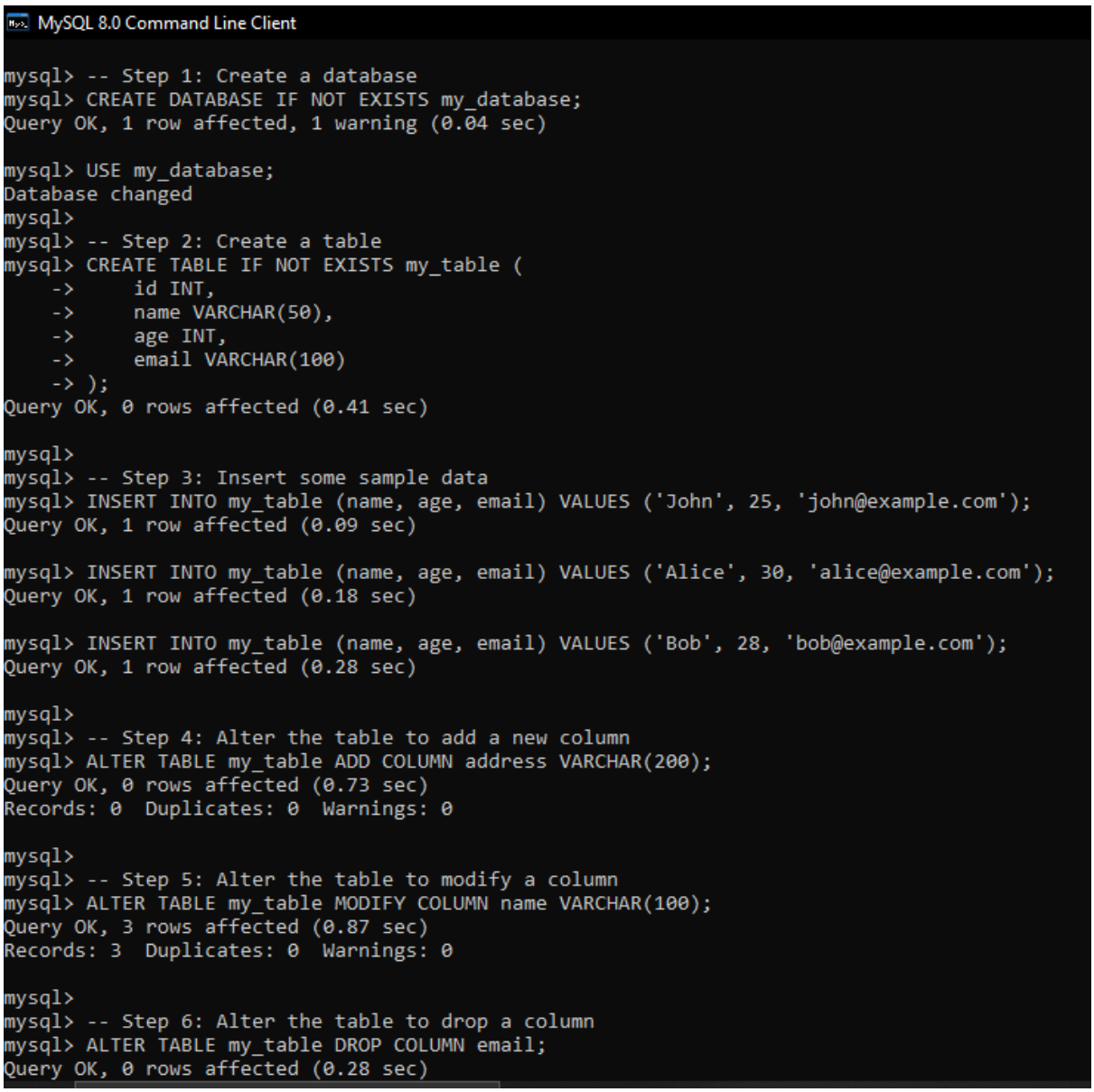


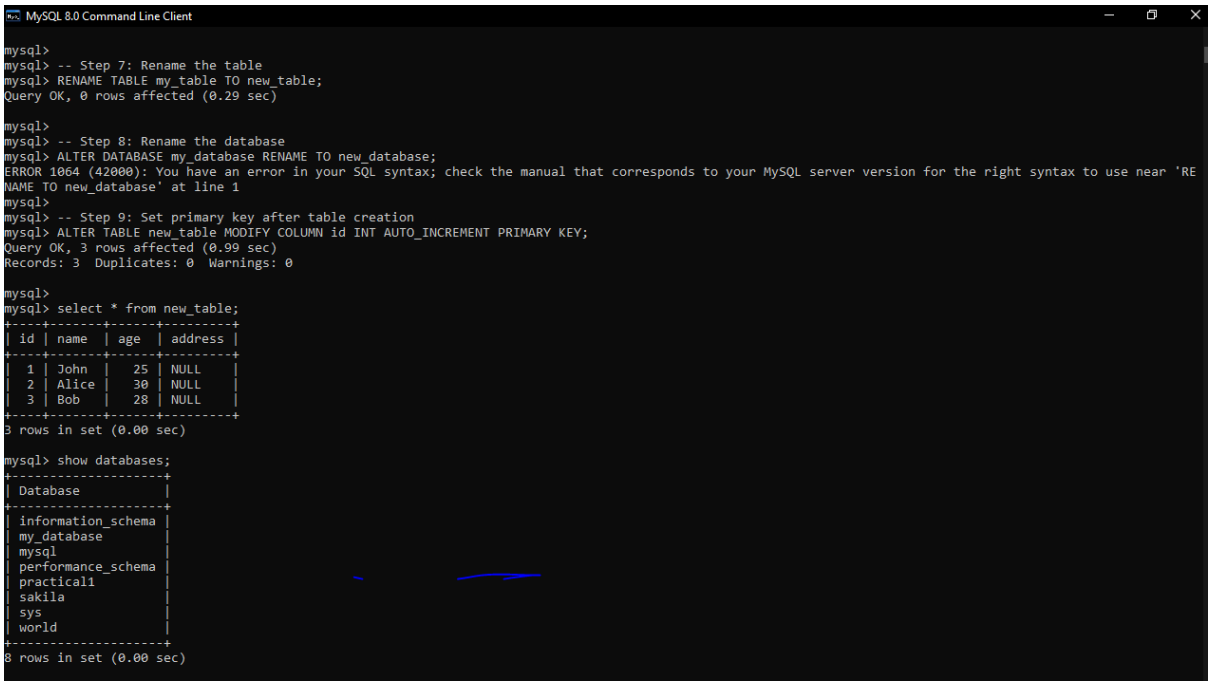


```
-- Step 1: Create a database
CREATE DATABASE IF NOT EXISTS my_database;
USE my_database;

-- Step 2: Create a table
CREATE TABLE IF NOT EXISTS my_table (
    id INT,
    name VARCHAR(50),
    age INT,
    email VARCHAR(100)
);

-- Step 3: Insert some sample data
INSERT INTO my_table (name, age, email) VALUES ('John', 25, 'john@example.com');
INSERT INTO my_table (name, age, email) VALUES ('Alice', 30, 'alice@example.com');
INSERT INTO my_table (name, age, email) VALUES ('Bob', 28, 'bob@example.com');

-- Step 4: Alter the table to add a new column
ALTER TABLE my_table ADD COLUMN address VARCHAR(200);

-- Step 5: Alter the table to modify a column
ALTER TABLE my_table MODIFY COLUMN name VARCHAR(100);

-- Step 6: Alter the table to drop a column
ALTER TABLE my_table DROP COLUMN email;

-- Step 7: Rename the table
RENAME TABLE my_table TO new_table;

-- Step 8: Rename the database
ALTER DATABASE my_database RENAME TO new_database;
```

-- Step 9: Set primary key after table creation

ALTER TABLE new_table MODIFY COLUMN id INT AUTO_INCREMENT PRIMARY KEY;




MySQL 8.0 Command Line Client
mysql>
mysql> -- Step 7: Rename the table
mysql> RENAME TABLE my_table TO new_table;
Query OK, 0 rows affected (0.29 sec)
mysql>
mysql> -- Step 8: Rename the database
mysql> ALTER DATABASE my_database RENAME TO new_database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'RE
NAME TO new_database' at line 1
mysql>
mysql> -- Step 9: Set primary key after table creation
mysql> ALTER TABLE new_table MODIFY COLUMN id INT AUTO_INCREMENT PRIMARY KEY;
Query OK, 3 rows affected (0.99 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql>
mysql> select * from new_table;
+----+-------+------+---------+
| id | name | age | address |
+----+-------+------+---------+
1	John	25	NULL
2	Alice	30	NULL
3	Bob	28	NULL
+----+-------+------+---------+
3 rows in set (0.00 sec)
mysql> show databases;
+--------------------+
| Database |
+--------------------+
| information_schema |
| my_database |
| mysql |
| performance_schema |
| practical1 |
| sakila |
| sys |
| world |
+--------------------+
8 rows in set (0.00 sec)

Set 3

Create a table to store employee details. Define input parameters within the CREATE PROCEDURE statement and pass them in the CALL statement

Answer:-

```
-- Create the table to store employee details
CREATE TABLE Employee (
    EmployeeID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Department VARCHAR(50),
    Position VARCHAR(50),
    Salary DECIMAL(10, 2)
);

-- Create a stored procedure to insert data into the Employee table
DELIMITER //
CREATE PROCEDURE InsertEmployee(
    IN p_FirstName VARCHAR(50),
    IN p_LastName VARCHAR(50),
    IN p_Department VARCHAR(50),
    IN p_Position VARCHAR(50),
    IN p_Salary DECIMAL(10, 2)
)
```

```
BEGIN
    INSERT INTO Employee (FirstName, LastName, Department, Position, Salary)
    VALUES (p_FirstName, p_LastName, p_Department, p_Position, p_Salary);
END //
DELIMITER ;

-- Call the stored procedure to insert data into the Employee table
CALL InsertEmployee('John', 'Doe', 'IT', 'Software Engineer', 60000.00);
```

Create a database, create a table, demonstrate all DML commands. Add column to the existing table, set primary key to any of the column, remove the primary key.

Answer :-

1. **Create a Database**:

```sql
CREATE DATABASE CompanyDB;
```

2. **Use the Database**:

```sql
USE CompanyDB;
```

3. **Create a Table**:

```sql
CREATE TABLE Employees (
    EmployeeID INT AUTO_INCREMENT,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Department VARCHAR(50),
    Position VARCHAR(50),
    Salary DECIMAL(10, 2),
    PRIMARY KEY (EmployeeID)
);
```

4. **Insert Data**:

```sql
INSERT INTO Employees (FirstName, LastName, Department, Position, Salary)
VALUES ('John', 'Doe', 'IT', 'Software Engineer', 60000.00);
```

5. **Select Data**:

```sql
SELECT * FROM Employees;
```

6. **Update Data**:

```sql
UPDATE Employees
SET Salary = 65000.00
WHERE EmployeeID = 1;
```

7. **Delete Data**:

```sql
DELETE FROM Employees
WHERE EmployeeID = 1;
```

8. **Add Column to Existing Table**:

```sql
ALTER TABLE Employees
```

ADD Email VARCHAR(100);

```

9. **Set Primary Key**:

```sql
ALTER TABLE Employees
ADD PRIMARY KEY (EmployeeID);
```

10. **Remove Primary Key**:

```sql
ALTER TABLE Employees
DROP PRIMARY KEY;
```
```

Create Database:

```
CREATE DATABASE database_name;
```

Use Database:

```
USE database_name;
```

Create Table:

```
CREATE TABLE table_name (
        column1 datatype,
        column2 datatype,
        ...
);
```

Insert Records:

```
INSERT INTO table_name VALUES
(value1, value2, ...),
(value1, value2, ...),
...;
```

Add Another Column:

```
ALTER TABLE table_name ADD COLUMN new_column datatype;
```

Change Column Name:

```
ALTER TABLE table_name CHANGE COLUMN old_column new_column datatype;
```

Delete a Record:

```
DELETE FROM table_name WHERE condition;
```

Demonstrate Logical Operator (AND, OR, NOT):

SELECT * FROM table_name WHERE condition1 AND/OR/NOT condition2;

Demonstrate Pattern Matching (LIKE):

SELECT * FROM table_name WHERE column_name LIKE 'pattern';

8.A.a)Add a primary key for a combination of columns location_id and country_id

create database Dipali;

use Dipali;

create table locations(location_id int,street_address varchar(40),pin_code varchar(12),city varchar(30),state varchar(25),country_id varchar(2));

ALTER TABLE locations ADD PRIMARY KEY(location_id,country_id);

show columns from locations;

8.A.b)Drop the existing primary from the table locations on a combination of columns location_id and country_id

ALTER TABLE locations DROP PRIMARY KEY;

show columns from locations;

8.A.c)Add a foreign key on job_id column of job_history table referencing to the primary key job_id of jobs table

ALTER TABLE jobs ADD PRIMARY KEY(job_id);

alter table job_history add foreign key(job_id) references jobs(job_id);

show columns from job_history;

9.A)Change salary of employee to 8000 whose ID is 105, if the existing salary is less than 5000

```
create database Dipali;

use Dipali;

create table employee(emp_id int , first_name varchar(25), last_name varchar(25), salary int
 ,job_title varchar(40));

insert into employee values(105 , "DIPALI" , "KHAIRNAR" , 4000 , 'CEO' );

update employee set salary = 8000 where emp_id = 105 and salary < 5000;\

select * from employee;
```

9.B) Change job ID of employee of ID is 118, to SH_CLERK if the employee in the department_id 30 and the job ID not started with SH

```
create database Dipali;

use Dipali;

create table employee(employee_id int,first_name varchar(25), last_name varchar(25),salary
int ,job_id varchar(40) , department_id int);

insert into employee values(118 , "DIPALI" , "KHAIRNAR" , 4000 , 'jn_clerk' ,30);

select * from employee;

update employee set job_id = "SH_CLERK" WHERE employee_id = 118 AND department_id =
30  AND NOT job_id LIKE 'SH%';

select * from employee;
```

- **Create table EMPLOYEE with attributes E_id, E_name, E_dept, E_salary, E_pno, E_city. Create view having E_id, E_name, E_dept, E_salary. create another table Employee details with some attributes and create another view from both the tables**

answer :

1) CREATE TABLE employee (
e_id INT PRIMARY KEY,
e_name VARCHAR(70),
e_dept VARCHAR(70),
e_salary int ,
e_pno VARCHAR(15),
e_city VARCHAR(30)
);

2) INSERT INTO employee (e_id, e_name, e_dept, e_salary, e_pno, e_city) VALUES (1, 'John Doe', 'IT', 50000.00, '123-456-7890', 'New York'), (2, 'Jane Smith', 'HR', 45000.00, '987-654-3210', 'Los Angeles'), (3, 'Michael Johnson', 'Finance', 55000.00, '555-555-5555', 'Chicago');

3) CREATE VIEW employee_view AS SELECT e_id, e_name, e_dept, e_salary FROM employee;

4) CREATE TABLE employee_details (
e_id INT PRIMARY KEY,
e_doj DATE,
e_age int,
e_pf int
);

5) INSERT INTO employee_details (e_id, e_doj, e_tpo, e_pf) VALUES (1, '2022-01-15', 2000.00, 1500.00), (2, '2021-05-20', 1800.00, 1200.00), (3, '2023-03-10', 2200.00, 1600.00);

6) CREATE VIEW employee_details_view ASSELECT e_id, e_doj, e_tpo, e_pf FROM employee_details;

SET 17 :

Display name, credit_rating, sales_rep_id from S_customer table of those customer who either satisfies the condition that credit_rating is greater than 5 out of 10 and sales_rep_id is equal to 4232. Demonstrate pattern matching and logical operator.

```
CREATE TABLE S_customer ( name VARCHAR(50), credit_rating INT, sales_rep_id INT);

INSERT INTO S_customer VALUES('Arun', 7, 4231), ('Atharva', 6, 4231);

SELECT name, credit_rating, sales_rep_id
FROM S_customer
WHERE (credit_rating > 5 AND sales_rep_id = 4232);
```

SET 18 :

Display the id, name and phone number of the customer

1) Whose id falls in the range 303 to 306

2)Whose id is greater than 300 and customer belongs to Pune

3)display the id, names of employee whose names contains fourth and fifth leters are 'sh' followed by anything and also belogs to pune city.

```
CREATE TABLE customers (
id INT PRIMARY KEY,
name VARCHAR(100),
phone_number VARCHAR(15),
city VARCHAR(50)
);

CREATE TABLE employees (
id INT PRIMARY KEY,
name VARCHAR(100),
city VARCHAR(50)
);
```

```
INSERT INTO customers (id, name, phone_number, city)
VALUES
 (301, 'John Doe', '123-456-7890', 'Pune'),
 (302, 'Jane Smith', '987-654-3210', 'Mumbai'),
 (303, 'Alice Wonderland', '555-123-4567', 'Pune'),
 (304, 'Bob Builder', '777-888-9999', 'Pune'),
 (305, 'Charlie Chaplin', '444-555-6666', 'Pune'),
 (306, 'David Beckham', '111-222-3333', 'Delhi');

INSERT INTO employees (id, name, city)
VALUES
 (101, 'Ashley Johnson', 'Pune'),
 (102, 'Michelle Sharma', 'Mumbai'),
 (103, 'Joshua Smith', 'Pune'),
 (104, 'Nisha Shah', 'Pune'),
 (105, 'Rajesh Patel', 'Mumbai'),
 (106, 'Rakesh Kumar', 'Pune');

SELECT * FROM customers
WHERE id BETWEEN 303 AND 306;

SELECT * FROM customers
WHERE id > 300 AND city = 'Pune';

SELECT * FROM employees
WHERE name LIKE '__sh%' AND city = 'Pune';
```

Set : 20

```
mysql> create table EmployeeDetails
    -> (
    -> EmpId int primary key,
    -> FullName varchar(50),
    -> ManagerId int,
    -> DateOfJoining date,
    -> City varchar(20)
    -> );
Query OK, 0 rows affected (0.20 sec)

mysql> insert into EmployeeDetails values
    -> (121,'John Snow',321,'2019-01-31','Toronto'),
    -> (321,'Walter White',986,'2020-01-30','California'),
    -> (421,'Kuldeep Rana',876,'2021-11-27','New Delhi');
Query OK, 3 rows affected (0.05 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

- Reference Table to solve following SQL queries : -

```
mysql> select * from EmployeeDetails;
+——+————+———+————+————+
| EmpId | FullName     | ManagerId | DateOfJoining | City       |
+——+————+———+————+————+
|   121 | John Snow    |       321 | 2019-01-31    | Toronto    |
|   321 | Walter White |       986 | 2020-01-30    | California |
|   421 | Kuldeep Rana |       876 | 2021-11-27    | New Delhi  |
+——+————+———+————+————+
3 rows in set (0.01 sec)
```

A) Write an SQL query to fetch the EmpId and FullName of all the employees working under the Manager Id – '986'.

```
mysql> select EmpId,FullName from EmployeeDetails where ManagerId = '986';
+——+————+
| EmpId | FullName     |
+——+————+
|   321 | Walter White |
+——+————+
1 row in set (0.06 sec)
```

B) write an sql query to fetch the employees whose name begins with any two characters,followed by text 'hn' and ends with any sequence of characters.

```
mysql> select * from EmployeeDetails
   -> where FullName like 'hn%';
Empty set (0.05 sec)
```

C) write an sql query to fetch the employees full names and replace the space with '-'

```
mysql> select replace(FullName, ' ','-')as modified_full_name
   -> from EmployeeDetails;
+——————————+
| modified_full_name |
+——————————+
| John-Snow          |
| Walter-White       |
| Kuldeep-Rana       |
+——————————+
3 rows in set (0.01 sec)
```

OR

```
mysql> DELIMITER //

mysql>

mysql> CREATE PROCEDURE AddTwoNumbers (IN num1 INT, IN num2 INT, OUT
result INT)

   -> BEGIN

   ->     SET result = num1 + num2;

   -> END //

Query OK, 0 rows affected (0.02 sec)


mysql>

mysql> DELIMITER ;

mysql> CALL AddTwoNumbers(10, 5, @sum);

Query OK, 0 rows affected (0.05 sec)


mysql> SELECT @sum AS SumResult;

+————+

| SumResult |

+————+

|       15 |

+————+

1 row in set (0.00 sec)
```