

Lab Report 2 DATABASE MANAGEMENT SYSTEM (CSEC-321)

SUBMITTED BY	SUBMITTED TO
Abrar Agmain Shazid	Md. Harun-Ar-Rashid
ID: 2222081036	Lecturer,
Batch: 57A Day	Department of CSE,
Department of CSE,	Uttara University
Uttara University	

Submission Date – 06/11/2024

GITHUB Link: https://github.com/AbrarShazid/DBMS/tree/main/Lab%20report%202

Task 1:

Using WHERE Clause

- Write a query to select all records from a table where a specific condition is met.
- Example: Retrieve all employees with a salary greater than 50,000.

Code:

```
1
    CREATE DATABASE db;
 2
    USE db;
 3
 4
    CREATE TABLE employee(
 5
    id INT PRIMARY KEY,
 6
    NAME VARCHAR(50),
 7
    designation VARCHAR (50),
     salary INT
 8
 9
    );
10
11
     INSERT INTO employee VALUES
    (1, 'Alice', 'Junior Engineer', 40000),
12
    (2, 'Bob', 'Junior Engineer', 44000), (3, 'Eve', 'Engineer', 48000),
13
14
     (4, 'John', 'Senior Engineer', 60000),
15
     (5,'Paul','Manager',55000),
16
     (6, 'Mofizzz', 'HR', 10);
17
18
     SELECT * FROM employee
19
    WHERE salary>50000;
20
21
22
```



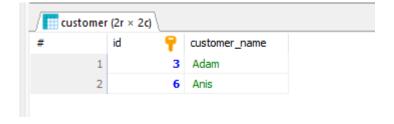
Task 2:

Using LIKE Operator

- Write a query to find records where a column's value matches a specified pattern.
- Example: Retrieve all customers whose names start with 'A'.

Code:

```
CREATE DATABASE db;
1
2
   USE db;
3
4
   CREATE TABLE customer(
5
   id INT PRIMARY KEY,
6
   customer_name VARCHAR(50)
7
   );
8
9
   INSERT INTO customer VALUES
   (1, 'Eric'),
0
   (2, 'John'),
1
2
   (3, 'Adam'),
   (4, 'Eve'),
3
   (5, 'Mike'),
4
5
   (6, 'Anis'),
6
   (7, 'James Aron');
7
8
   SELECT * FROM customer
9
   WHERE customer_name LIKE 'A%';
0
```



Task 3:

Using GROUP BY Clause

- Write a query to group records by a specific column and perform aggregate functions.
- Example: Group employees by department and calculate the average salary.

Code:

```
1
    CREATE DATABASE db;
 2
    USE db;
 3
    CREATE TABLE emp(
 4
 5
    id INT PRIMARY KEY,
    designation VARCHAR(50),
 6
 7
    department VARCHAR(50),
8
    salary int
9
    );
10
11
    INSERT INTO emp VALUES
    (1, 'Junior Engineer', 'Development', 40000),
12
    (2, 'Engineer', 'Development', 50000),
13
14
    (3, 'Senior Engineer', 'Development', 55000),
    (4, 'Manager', 'Management', 52000),
15
    (5, 'Accountant', 'Finance', 35000),
16
    (6, 'CEO', 'Management', 900000),
17
    (7, 'Senior Accountant', 'Finance', 65000);
18
19
    SELECT department, AVG(salary) FROM emp
20
21
    GROUP BY department;
```

```
# department AVG(salary)

1 Development 48,333.3333
2 Finance 50,000.0
3 Management 476,000.0
```

Task 4:

Using HAVING Clause

- Write a query to filter groups based on a condition.
- Example: Retrieve departments with an average salary greater than 60,000.

Code:

```
1
    CREATE DATABASE db;
 2
    USE db;
 3
 4
    CREATE TABLE emp(
 5
    id INT PRIMARY KEY,
 6
    designation VARCHAR(50),
 7
    department VARCHAR(50),
 8
    salary int
 9
    );
10
    INSERT INTO emp VALUES
11
    (1, 'Junior Engineer', 'Development', 40000),
12
    (2, 'Engineer', 'Development', 50000),
13
    (3, 'Senior Engineer', 'Development', 55000),
14
    (4, 'Manager', 'Management', 52000),
15
16
    (5, 'Accountant', 'Finance', 55000),
    (6, 'CEO', 'Management', 100000),
17
    (7, 'Senior Accountant', 'Finance', 67000);
18
19
20
    SELECT department, AVG(salary) FROM emp
21
    GROUP BY department
22
    HAVING avg(salary)>60000;
```



Task 5:

Using IN Operator

- Write a query to select records where a column's value is within a specified set of values.
- Example: Retrieve all orders placed by customers with IDs 1, 2, and 3.

Code:

```
create database db;
 2
    USE db;
 3
   CREATE TABLE customer(
 5
   id INT,
   order_item VARCHAR (100),
 6
   price INT);
 7
 8
 9
    INSERT INTO customer VALUES
    (1, 'Shirt', 1200),
10
   (2,'Watch',2200),
11
   (3,'Pant',1700),
12
   (4,'SSD',10000),
13
   (1,'Earphone',1900),
14
   (1, 'Medicine', 765),
15
   (5,'Phone',120000),
16
   (3, 'Laptop', 100000);
17
18
19
    SELECT * FROM customer
20
    WHERE id IN (1,2,3);
```

customer (6r × 3c)					
#	id	order_item	price		
1	1	Shirt	1,200		
2	2	Watch	2,200		
3	3	Pant	1,700		
4	1	Earphone	1,900		
5	1	Medicine	765		
6	3	Laptop	100,000		

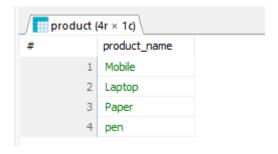
Task 6:

Using NOT IN Operator

- Write a query to select records where a column's value is not within a specified set of values.
- Example: Retrieve all products that are not in categories 1, 2, and 3.

Code:

```
create database db;
 1
 2
    USE db;
 3
 4
    CREATE TABLE product (
 5
        product id INT PRIMARY KEY,
        product name VARCHAR(100),
 6
        category id INT);
 7
 8
    INSERT INTO product VALUES
 9
    (1, 'Shirt', 1),
10
    (2, 'Pant',1),
11
    (3, 'Earphone', 2),
12
    (4, 'Cable', 2),
13
14 (5, 'Chips', 3),
   (6,'Mobile',4),
15
    (7, 'Laptop',4),
16
    (8, 'Paper',5),
17
    (9, 'pen', 5);
18
19
    SELECT product name FROM product
20
    WHERE category_id NOT IN (1,2,3);
21
22
```



Task 7:

Using Sub queries

- Write a query that includes a sub query to perform a more complex selection.
- Example: Retrieve all employees who work in a department with more than 10 employees.

Code:

```
create database db;
 2
       USE db;
 3
     CREATE TABLE emp (
             emp_id INT PRIMARY key,
 5
             designation VARCHAR(100),
 6
            department VARCHAR(100));
 7
       INSERT INTO emp VALUES
       (1, 'Junior Engineer', 'Development'),
(2, 'Junior Engineer', 'Development'),
(3, 'Junior Engineer', 'Development'),
(4, 'Junior Engineer', 'Development'),
(5, 'Junior Engineer', 'Development'),
 9
10
11
       (6, 'Engineer', 'Development'),
(7, 'Engineer', 'Development'),
(8, 'Engineer', 'Development'),
(9, 'Senior Engineer', 'Development'),
(10, 'Lead Engineer', 'Development'),
(11, 'Senior Engineer', 'Development'),
13
14
15
16
17
18
       (12, 'Accountant', 'Finance'),
19
       (13, 'Senior Accountand', 'Finance'),
20
       (14, 'Manager', 'Management'),
21
22
       (15, 'Product Manager', 'Management'),
        (16, 'Digital Marketing', 'Marketing'),
23
       (17,'CEO','Management'), (18,'MD','Management'),
24
25
       (19,'Marketing Head','Management'),
26
27
       (20, 'Finance Director', 'Management');
28
29
       SELECT emp_id, designation, department
       FROM emp
30
31
       WHERE department IN (
32
             SELECT department
33
             FROM emp
34
             GROUP BY department
35
             HAVING COUNT(emp_id) > 10
36 );
```

emp (11r × 3c)				
#	emp_id 🥊	designation	department	
1	1	Junior Engineer	Development	
2	2	Junior Engineer	Development	
3	3	Junior Engineer	Development	
4	4	Junior Engineer	Development	
5	5	Junior Engineer	Development	
6	6	Engineer	Development	
7	7	Engineer	Development	
8	8	Engineer	Development	
9	9	Senior Engineer	Development	
10	10	Lead Engineer	Development	
11	11	Senior Engineer	Development	