Shri Ramdeobaba College of Engineering & Management Nagpur-13

Department of Computer Application

Session: 2023-2024

Submission for

Course Name: Database Management Systems Lab

Course Code: MCP545

Name of the Student: Jayesh Lalit Nandanwar

Class Roll No: 26

Semester: MCA II semester

Shift: 2

Batch: 2

Under the Guidance of

Prof. Yoginee Pethe

Date of submission:

List of Experiments

1. Implementation of DDL commands of SQL with suitable examples
   a) Create table b) Alter table c) Truncate table d) Drop table

   Implementation of DML commands of SQL with suitable examples

   a) Select b) Insert c) Update d) Delete
2. Study and implementation of different types of constraints.
3. Implementation of different types of operators in SQL

   a)Arithmetic Operators b) Logical Operators

   c) Comparison Operators d) Set Operators
4. Study and Implementation of
   a) Aggregate functions
   b) Group By & Having clause
   c) Order by clause
5. Study and Implementation of different types of joins like cross join, natural join, inner join, and outer joins.
6. Study and Implementation of Sub queries
7. Study and Implementation of views, and synonyms.
8. Study and Implementation of PL/SQL.

Practical 6

Aim: Study and Implementation of Sub queries

Solution:

Employee (emp_no, emp_name, emp_sal, emp_comm, dept_no, Job_id, location)

Job (job_id, job_title, min_sal, max_sal)

department (dept_no, dept_name, location_id).

employee_manager (emp_no, emp_hiredate, mng_no,mng_name).

Query:
```
CREATE TABLE Employee (
  emp_no INT PRIMARY KEY,
  emp_name VARCHAR(100),
  emp_sal DECIMAL(10, 2),
  emp_comm DECIMAL(10, 2),
  dept_no INT, Job_id INT,
  location VARCHAR(100)
);

CREATE TABLE Job (
  job_id INT PRIMARY KEY,
  job_title VARCHAR(100),
  min_sal DECIMAL(10, 2),
  max_sal DECIMAL(10, 2)
);

CREATE TABLE Department (
  dept_no INT PRIMARY KEY,
  dept_name VARCHAR(100),
  location_id INT
);

CREATE TABLE Employee_Manager (
  emp_no INT PRIMARY KEY,
  emp_hiredate DATE, mng_no INT,
  mng_name VARCHAR(100)
);
```

1. Create a report that displays the employee number, name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary

Query: SELECT emp_no, emp_name, emp_sal FROM Employee
WHERE emp_sal > (SELECT AVG(emp_sal) FROM Employee)
ORDER BY emp_sal ASC;

2. The HR department needs a report that displays the name, department number, and job ID of all employees whose department location ID is 1700.

Query: SELECT e.emp_name, d.dept_no, e.job_id FROM Employee e
JOIN department d ON e.dept_no = d.dept_no
WHERE d.location_id = 1700;

3. Create a report for HR that displays the department number, name, and job ID for every employee in the Executive department

Query: SELECT d.dept_no, d.dept_name, e.job_id
FROM Employee e JOIN department d ON e.dept_no = d.dept_no
WHERE d.dept_name = 'Executive';

4. Create a report that displays a list of all employees whose salary is more than the salary of any employee from department 60..

Query: SELECT emp_no, emp_name, emp_sal FROM Employee
WHERE emp_sal > ANY (SELECT emp_sal FROM Employee
WHERE dept_no = 60
);

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

Table: Customers

order_id	amount	customer_id
1	200	4
2	500	1
3	300	3
4	800	1
5	150	2

Table: Orders

```
CREATE TABLE Customer ( Customer_id INT PRIMARY KEY, First_name
                        VARCHAR(50), Last_name VARCHAR(50), Age INT,
                        Country VARCHAR(50)
);
INSERT INTO Customer (Customer_id, First_name, Last_name, Age, Country)
VALUES(1, 'John', 'Doe', 31,'USA');
INSERT INTO Customer (Customer_id, First_name, Last_name, Age, Country)
VALUES(2, 'Robert', 'Luna', 22, 'USA');
INSERT INTO Customer (Customer_id, First_name, Last_name, Age, Country)
VALUES(3, 'David', 'Robinson', 22, 'UK');
INSERT INTO Customer (Customer_id, First_name, Last_name, Age, Country)
VALUES(4, 'John', 'Reinhardt', 25, 'UK');
INSERT INTO Customer (Customer_id, First_name, Last_name, Age, Country)
VALUES(5, 'Betty', 'Doe', 28, 'UAE');
```

Results | Explain | Describe | Saved SQL | History

CUSTOMER_ID	FIRST_NAME	LAST_NAME	AGE	COUNTRY
1	John	Doe	31	USA
4	John	Reinhardt	25	UK
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
5	Betty	Doe	28	UAE

5 rows returned in 0.01 seconds Download

```
CREATE TABLE Orders (
    Order_id INT PRIMARY KEY,
    Amount INT,
    Customer_id INT,
    FOREIGN KEY (Customer_id) REFERENCES Customer(Customer_id)
);
INSERT INTO Orders (Order_id, Amount, Customer_id) VALUES(1, 200, 4);
INSERT INTO Orders (Order_id, Amount, Customer_id) VALUES(2, 500, 1);
INSERT INTO Orders (Order_id, Amount, Customer_id) VALUES(3, 300, 3);
INSERT INTO Orders (Order_id, Amount, Customer_id) VALUES(4, 800, 1);
INSERT INTO Orders (Order_id, Amount, Customer_id) VALUES(5, 150, 2);
```

Results | Explain | Describe | Saved SQL | History

ORDER_ID	AMOUNT	CUSTOMER_ID
1	200	4
3	300	3
4	800	1
2	500	1
5	150	2

5 rows returned in 0.01 seconds Download

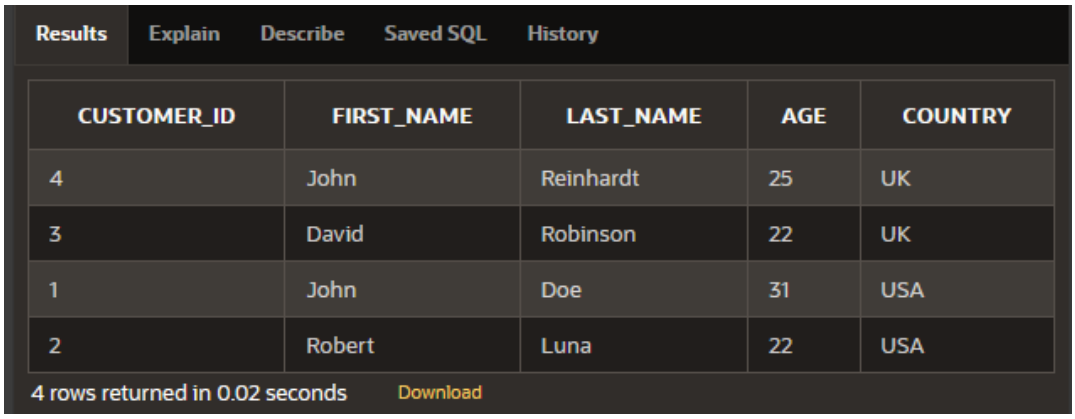
1. Find the details of customers who have placed an order.

Query: SELECT c.* FROM Customer c

WHERE EXISTS (SELECT 1 FROM Orders o

WHERE o.Customer_id = c.Customer_id

);



Results | Explain | Describe | Saved SQL | History

CUSTOMER_ID	FIRST_NAME	LAST_NAME	AGE	COUNTRY
4	John	Reinhardt	25	UK
3	David	Robinson	22	UK
1	John	Doe	31	USA
2	Robert	Luna	22	USA

4 rows returned in 0.02 seconds Download

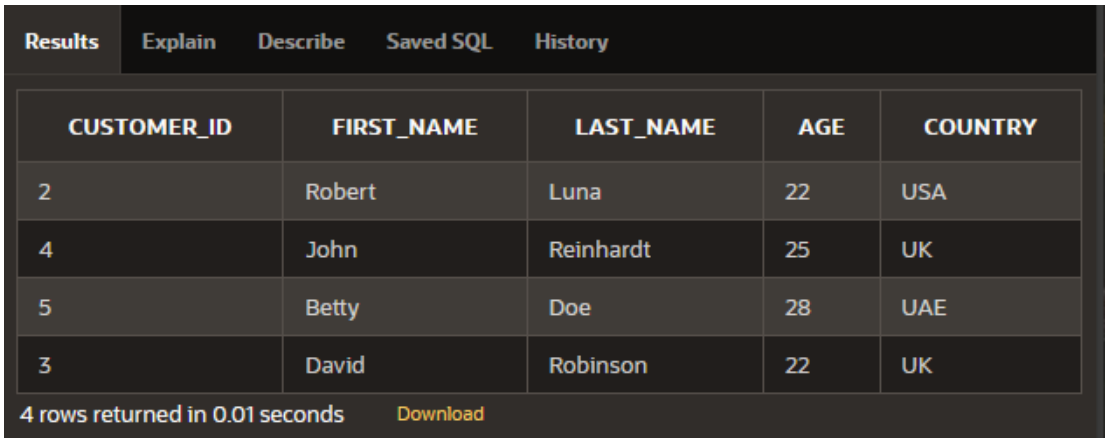
2. Find all customers who do not have order more than $500.

Query: SELECT c.* FROM Customer c

WHERE NOT EXISTS (SELECT 1 FROM Orders o

WHERE o.Customer_id = c.Customer_id

AND o.Amount > 500);



Results | Explain | Describe | Saved SQL | History

CUSTOMER_ID	FIRST_NAME	LAST_NAME	AGE	COUNTRY
2	Robert	Luna	22	USA
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE
3	David	Robinson	22	UK

4 rows returned in 0.01 seconds Download

3. Find the customers who have placed highest order.

Query: SELECT c.* FROM Customer c JOIN Orders o ON c.Customer_id = o.Customer_id
WHERE o.Amount = (SELECT MAX(Amount) FROM Orders);

Results | Explain | Describe | Saved SQL | History

CUSTOMER_ID	FIRST_NAME	LAST_NAME	AGE	COUNTRY
1	John	Doe	31	USA

1 rows returned in 0.01 seconds Download

4. Find all customers whose age is greater than the average age of all customers.

Query: SELECT * FROM Customer
WHERE Age > (SELECT AVG(Age) FROM Customer)

Results | Explain | Describe | Saved SQL | History

CUSTOMER_ID	FIRST_NAME	LAST_NAME	AGE	COUNTRY
1	John	Doe	31	USA
5	Betty	Doe	28	UAE

2 rows returned in 0.01 seconds Download

5. Find the name of customers whose age is equal to age of John.

Query: SELECT First_name, Last_name FROM Customer
WHERE Age = (SELECT MAX(Age) FROM Customer
WHERE First_name = 'John');

Results | Explain | Describe | Saved SQL | History

FIRST_NAME	LAST_NAME
John	Doe

1 rows returned in 0.01 seconds Download