Right Side work

Task 4- **Using Functions in queries and writing subqueries:**

Aim: Perform the advanced query processing and test its heuristics using the designing of optimal correlated and nested subqueries such as finding summary statistics.

Procedures:

Queries:

A query is a request to retrieve specific data from a database. It's a way to ask the database for specific information, and the database returns the results in a structured format.

Syntax:

SELECT * FROM tablename WHERE condition;

1. The SELECT * statement requests that all columns (id, name, country, and age) be returned for the matching rows.
2. The FROM clause specifies the table to query, which is customers.
3. The WHERE clause filters the results to only include rows where the country column is equal to 'USA

Subqueries:

- A Subquery is a query within another SQL query and embedded within the WHERE clause.
- You can use Subquery with SELECT, UPDATE, INSERT, DELETE statements along with the operators like =, <, >, >=, <=, IN, BETWEEN, etc.
- A subquery is **a query within another query**. The **outer query** is known as the main **query,** and the **inner query** is known as a **subquery.**
- Subqueries are on the **right side of the comparison operator.**
- A subquery is **enclosed in parentheses.**
- In the Subquery, ORDER BY command cannot be used. But GROUP BY command can be used to perform the same function as ORDER BY command.

Program:

Create 2 table name as branch and cust2 and apply subqueries

```
SQL> create table branch( b_code number(10) primary key,b_loc varchar(20),manager
varchar(20));

Table created.

SQL> create table cust2 (c_id  number (10),c_name varchar(10),c_age number(10),b_code
number(10)references branch (b_code));

Table created.


SQL> insert into branch values(1001,'chennai','shree');

1 row(s) inserted.

SQL> insert into branch values(1002,'tambaram','raja');

1 row(s) inserted.

SQL> insert into branch values(1003,'chengalpat','uday');

1 row(s) inserted.

SQL> insert into branch values(1004,'banglore','abii');

1 row(s) inserted.

SQL> insert into cust2 values(11,'Ramesh',19,1001);

1 row(s) inserted.

SQL> insert into cust2 values(13,'kiran',25,1001);

1 row(s) inserted.

SQL> insert into cust2 values(14,'kannan',25,1002);

1 row(s) inserted.
SQL> insert into cust2 values(15,'vino',24,1003);
```

Left side work

Output:

Example1:

A) Sub queries

```
SQL> select c_name from cust2 where c_age=(select c_age from cust2 where c_id=15);
```

C_NAME
vino

B) Subqueries that returns several values

```
SQL> select c_name,c_age,c_id from cust2 where c_age<any(select c_age from cust2 where
c_id>=14);
```

C_NAME	C_AGE	C_ID
Raja	19	11
vino	24	15

C) Multiple SubQueries

```
SQL> select b_code,c_age,c_id from cust2 where c_age=any(select c_age from cust2 where
c_id=(select c_id from cust2 where c_name='vino'));
```

B_CODE	C_AGE	C_ID
1003	24	15

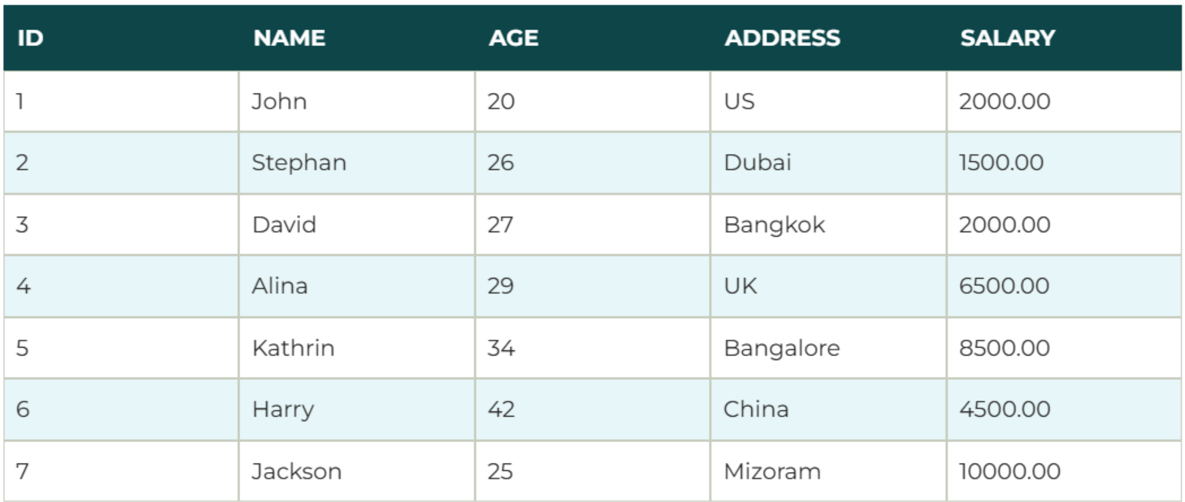
D) Correlated Subquery

SQL> select c_name,c_age,c_id from cust2 where c_age=any(select c.c_age from cust2 c,branch b where b.b_code=c.b_code);

C_NAME	C_AGE	C_ID
Raja	19	11
kannan	25	14
kiran	25	13
vino	24	15

Example: 2

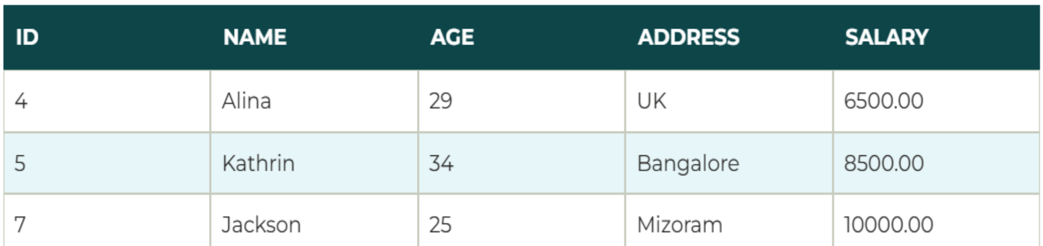
Create table employee(id number(5), name varchar(20), age number(4), address varchar(20), salary varchar(20);



ID	NAME	AGE	ADDRESS	SALARY
1	John	20	US	2000.00
2	Stephan	26	Dubai	1500.00
3	David	27	Bangkok	2000.00
4	Alina	29	UK	6500.00
5	Kathrin	34	Bangalore	8500.00
6	Harry	42	China	4500.00
7	Jackson	25	Mizoram	10000.00

1) **Subqueries with the Select Statement:**

Sql>select * from employee where id in (select id from employee where salary > 4500);



ID	NAME	AGE	ADDRESS	SALARY
4	Alina	29	UK	6500.00
5	Kathrin	34	Bangalore	8500.00
7	Jackson	25	Mizoram	10000.00

Example 3: sample queries : Using IN, Comparison

- **With the IN Operator**:

```
SELECT employee_id, first_name, last_name FROM employees WHERE
department_id NOT IN (SELECT department_id FROM departments WHERE
location_id = 1700) ORDER BY first_name, last_name;
```

- **With Comparison Operators**:

```
SELECT employee_id, first_name, last_name, salary FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);
```