Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act, 1956) School of Computing

B

Technology

VTR UGE2021- (CBCS)

Academic Year: 2025–2026

SUMMER SEMESTER - SS2526

Course Code : 10211IT201

Course Name : Database System Concepts

Slot No : S12L5

DBMS TASK - 4 REPORT

Submitted by:

VTUNO	REGISTER NUMBER	STUDENT NAME
VTU29010	24UEIT0040	P.Manikanta

ABSTRACT

The objective of this task is to demonstrate the use of nested queries (subqueries) and different types of joins in SQL.

Subqueries enable users to retrieve data by embedding one query inside another, which helps in performing advanced filtering and data comparison. Joins are used to combine rows from two or more tables based on related columns.

This task uses examples of simple subqueries, multiple subqueries, correlated subqueries, and join operations such as Equi-Join, Outer Join, and Self Join on the branch and cust2 tables.

1. Create Tables

```
CREATE TABLE Branch ( b_code
  NUMBER(10) PRIMARY KEY, b_loc
  VARCHAR2(20), manager
  VARCHAR2(20)
);
SQL> DESC BRANCH;
Name                          Null?    Type
B_CODE                        NOT NULL NUMBER(10)
B_LOC                                  VARCHAR2(20)
MANAGER                                VARCHAR2(20)
CREATE TABLE Cust2 ( c_id
  NUMBER(10) PRIMARY KEY,
  c_name VARCHAR2(20), c_age
  NUMBER(10), b_code
  NUMBER(10),
  CONSTRAINT fk_branch FOREIGN KEY (b_code) REFERENCES Branch(b_code)
```

);

SQL> DESC CUST2;

Name	Null?	Type
C_ID	NOT NULL	NUMBER(10)
C_NAME		VARCHAR2(20)
C_AGE		NUMBER(10)
B_CODE		NUMBER(10)

2. Insert Records

```
INSERT INTO Branch VALUES (1001, 'Chennai', 'Shree');
INSERT INTO Branch VALUES (1002, 'Tambaram', 'Raja');
INSERT INTO Branch VALUES (1003, 'Chengalpat', 'Uday');
INSERT INTO Branch VALUES (1004, 'Bangalore', 'Abii');
```

SQL> SELECT*FROM BRANCH;

B_CODE	B_LOC	MANAGER
1001	Chennai	Shree
1002	Tambaram	Raja
1003	Chengalpat	Uday
1004	Bangalore	Abii

```
INSERT INTO Cust2 VALUES (11, 'Ramesh', 19, 1001);
```

INSERT INTO Cust2 VALUES (13, 'Kiran', 25, 1001);

INSERT INTO Cust2 VALUES (14, 'Kannan', 25, 1002);

INSERT INTO Cust2 VALUES (15, 'Vino', 24, 1003);

SQL> SELECT*FROM CUST2;

C_ID	C_NAME	C_AGE	B_CODE
11	Ramesh	19	1001
13	Kiran	25	1001
14	Kannan	25	1002
15	Vino	24	1003

NESTED QUERIES

a. Simple Subquery

SELECT c_name

FROM Cust2

WHERE c_age = (SELECT c_age FROM Cust2 WHERE c_id = 15);

C_NAME

Vino

b. Subquery Returning Multiple Values

```
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
--------------------- ----------- -----------
Ramesh                19          11
Vino                  24          15
```

c. Multiple Subqueries

```
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

```
SELECT c_name, c_age, c_id
FROM Cust2 c
WHERE c_age = ANY
```

```
   ( SELECT c2.c_age
   FROM Cust2 c2, Branch b
   WHERE b.b_code = c2.b_code
);
```

```
C_NAME                 C_AGE       C_ID
----------------------------------------
Ramesh                  19          11
Kiran                   25          13
Kannan                  25          14
Vino                    24          15
```

JOINS

a. Simple (Equi) Join

```
SELECT c.c_id, c.c_name, c.c_age, c.b_code,
      b.b_loc, b.manager
FROM Cust2 c
JOIN Branch b ON c.b_code = b.b_code;
```

```
    C_ID C_NAME                 C_AGE     B_CODE B_LOC
---------- ---------------------- ---------- ---------- --------------------
MANAGER
      11 Ramesh                  19       1001 Chennai
Shree
```

25 1001 Chennai

13 Kiran

Shree

25 1002 Tambaram

14 Kannan

Raja

15 Vino 24 1003 Chengalpat

Uday

b. Outer Join

SELECT c.c_id, c.c_name, c.c_age, c.b_code,

b.b_loc, b.manager

FROM Cust2 c

RIGHT OUTER JOIN Branch b ON c.b_code = b.b_code;

C_ID C_NAME C_AGE B_CODE B_LOC

---------- ---------- ---------- ---------- ----------

MANAGER

---

11 Ramesh 19 1001 Chennai

Shree

13 Kiran 25 1001 Chennai

Shree

14 Kannan 25 1002 Tambaram

Raja

15 Vino 24 1003 Chengalpat

Uday

Bangalore

Abii

c. Left Outer Join (Alternative Example)

SELECT c.c_id, c.c_name, c.c_age, c.b_code,

b.b_loc, b.manager

FROM Cust2 c

LEFT OUTER JOIN Branch b ON c.b_code = b.b_code; C_ID C_NAME C_AGE B_CODE B_LOC

MANAGER

---

11 Ramesh 19 1001 Chennai

Shree

13 Kiran 25 1001 Chennai

Shree\

14 Kannan 25 1002 Tambaram

Raja

15 Vino 24 1003 Chengalpat
Uday

e. Self Join

```
SELECT a.c_id AS Customer_ID, a.c_name AS Customer_Name,
    b.c_id AS Ref_ID, b.c_name AS Ref_Name
FROM Cust2 a, Cust2 b
WHERE a.b_code = b.b_code AND a.c_id > 13;
```

CUSTOMER_ID CUSTOMER_NAME REF_ID REF_NAME

---------- ---------- ---------- ----------

14 Kannan 14
Kannan

15 Vino 15 Vino

RESULT:- The queries are executed successfully