Aayush Shah

Batch D1

19BCE245

29 April 2021

Practical 9

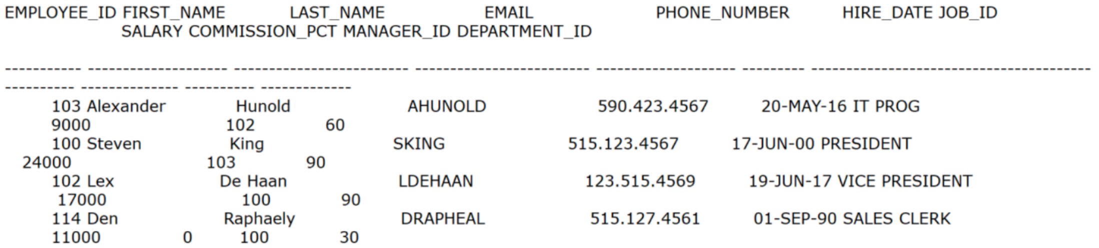
Do as directed.

1. **The organization wants to display only the details of the employees those who are managers [Manager_Record].**
   - SQL> CREATE VIEW manager_record AS SELECT * FROM employees WHERE (employee_id IN (SELECT manager_id FROM employees));
     View created.

2. **The organization wants to display only the details like empno, empname, deptno, deptname of the employees [Employee_Detail].**
   - SQL> CREATE VIEW employee_detail AS SELECT employee_id, first_name, department_id, dname FROM employees INNER JOIN department ON department_id=dept_no;
     View created.

3. **The organization wants to display only the details like empno,empname,deptno,deptname of the all the employees except the managers and [NoManager].**
   - SQL> CREATE VIEW no_manager AS SELECT employee_id, first_name, dept_no, dname FROM employees INNER JOIN department ON department_id=dept_no WHERE (employee_id NOT IN (SELECT manager_id FROM employees));
     View created.

4. **Display all the views generated.**
   - SELECT * FROM manager_record;



```
EMPLOYEE_ID FIRST_NAME          LAST_NAME                EMAIL                    PHONE_NUMBER         HIRE_DATE JOB_ID
                SALARY COMMISSION_PCT MANAGER_ID DEPARTMENT_ID

----------- -------------------- ------------------------- ------------------------- -------------------- --------- ----------------------------------------
---------- -------------- ---------- -------------
       103 Alexander            Hunold                    AHUNOLD                   590.423.4567         20-MAY-16 IT PROG
       9000                    102           60
       100 Steven              King                      SKING                     515.123.4567         17-JUN-00 PRESIDENT
  24000                     103           90
       102 Lex                 De Haan                   LDEHAAN                   123.515.4569         19-JUN-17 VICE PRESIDENT
        17000                   100           90
       114 Den                 Raphaely                  DRAPHEAL                  515.127.4561         01-SEP-90 SALES CLERK
       11000             0       100           30
```

- SQL> SELECT * FROM employee_detail;

```
EMPLOYEE_ID FIRST_NAME DEPARTMENT_ID DNAME

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

100 Steven          90 Sales

101 Neena           90 Sales

102 Lex                     90 Sales
103 Alexander       60 Research

104 Bruce           60 Research

105 David           60 Research

106 Valli           60 Research

114 Den             30 Accounting

119 Karen           30 Accounting

206 William                 110 Marketing
```

10 rows selected.

```
SQL> SELECT * FROM no_manager;

EMPLOYEE_ID FIRST_NAME DEPARTMENT_ID DNAME

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

119 Karen           30 Accounting

105 David           60 Research

106 Valli           60 Research

104 Bruce           60 Research

101 Neena           90 Sales

206 William                 110 Marketin

6 rows selected.
```

5. **Execute the DML commands on the view created.**

Inserting a new row in employees table :

```
SQL> ed

Wrote file afiedt.buf

1 INSERT INTO employees

2* VALUES (222, 'John', 'King', 'JKING', '515.123.4590',
TO_DATE('2003- 09-17','YYYY-MM-DD'), 'IT PROG', 10000.00, NULL,
103, 120)
SQL> /

1 row created.
```

Respective row is also added in the view:

```
SQL> SELECT * FROM employee_detail;


EMPLOYEE_ID FIRST_NAME          DEPARTMENT_ID DNAME

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

100 Steven                    90 SALES

100 Neena                     90 SALES

101 Lex                     90 SALES

102 Alexander                  60 RESEARCH

103 Bruce                     60 RESEARCH

104 David                     60 RESEARCH

105 Valli                    60 RESEARCH

114 Den                      30 ACCOUNTING

119 Karen                     30 ACCOUNTING

206 William                   110 MARKETING

222 John                     90 SALES

10 rows selected.
```

Updating an existing row in employees table

```
SQL> UPDATE employees SET department_id = 90 WHERE employee_id=222;

1 row updated.
```

Respective row is also updated in the view:

```
SQL> SELECT * FROM no_manager;

EMPLOYEE_ID FIRST_NAME           DEPT_NO DNAME

----------- -------------------- ---------- --------------------     119 Karen                30
ACCOUNTING

       105 David                 60 RESEARCH

       106 Valli                60 RESEARCH

     104 Bruce                 60 RESEARCH

     101 Neena                 90 SALES

     222 John                  90 SALES

     206 William                110 MARKETING

7 rows selected.
```

Deleting an existing row in employees table

```
SQL> DELETE FROM employees WHERE employee_id=222;

1 row deleted.
```

Respective row is also deleted from the view:

```
SQL> SELECT * FROM employee_detail;


EMPLOYEE_ID FIRST_NAME          DEPARTMENT_ID DNAME
----------- ------------------- ------------- --------------------
 100 Steven                   90 SALES
        101 Neena                   90 SALES
        102 Lex                   90 SALES
        103 Alexander                 60 RESEARCH
        104 Bruce                   60 RESEARCH
                105  David                  60 RESEARCH
                106 Valli                  60 RESEARCH
        114 Den                   30 ACCOUNTING
        119 Karen                   30 ACCOUNTING
        206 William                  110 MARKETING

10 rows selected.
```

6. **Find the departments of all mangers from Manager_detail.**

```
SQL> SELECT department_id FROM manager_record;
DEPARTMENT_ID

-------------
60 90 90 30
```

7. **Find name along with department name from Employee_detail.**

```
SQL> SELECT first_name, dname FROM employee_detail;

FIRST_NAME          DNAME
------------------- --------------------
Steven              Sales
Neena               Sales
Lex                 Sales
Alexander           Research
Bruce               Research
David               Research
Valli               Research
Den                 Accounting
Karen               Accounting
William             Marketing
```

```
10 rows selected.
```

8. Find Eno, and their corresponding dname from No_Manager.

```
SQL> SELECT employee_id, dname FROM no_manager;
EMPLOYEE_ID DNAME
----------- --------------------
    119 Accounting
      105 Research
      106 Research
      104 Research

      206 Marketing

6 rows selected.
```

9. Add a column Address in Manager_Record.

```
SQL> ALTER TABLE employees ADD address varchar(8);

Table altered.

SQL> CREATE OR REPLACE VIEW manager_record AS SELECT * FROM
employees WHERE (employee_id IN (SELECT manager_id FROM
employees));

View created.
```

10. Change a column name Deptno to D_ID in No_Manager.

```
SQL> CREATE OR REPLACE VIEW no_manager AS SELECT
employee_id, first_name, dept_no as d_id, dname FROM employees
INNER JOIN department ON department_id=dept_no WHERE
(employee_id NOT IN (SELECT manager_id FROM employees));
View created.
```

11. Change size of Empname column to 20 in Employee_Detail

```
SQL> ALTER TABLE employees MODIFY first_name varchar(20);

Table altered.

SQL> DESC employee_detail;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------  EMPLOYEE_ID
NUMBER(6)
```

```
 FIRST_NAME                              VARCHAR2(20)
 DEPARTMENT_ID                              NUMBER(4)  DNAME
VARCHAR2(20)
```

12. **Drop a view.**

```
SQL> DROP VIEW employee_detail;

View dropped.

SQL> DROP VIEW manager_record;

View dropped.

SQL> DROP VIEW no_manager;

View dropped.
```

13. **Create a sequence to insert the data in table person(pid, name, age), which automatically takes the value of pid, which starts with 101 and incremented by 1, and the valid range for pid is 101-199.**

```
SQL> CREATE SEQUENCE seq_pid
2.START WITH 101
3.INCREMENT BY 1
4.MAXVALUE 199 cycle;

Sequence created.
```

14. **Update the sequence created in the above question and increment the value with 2.**

```
SQL> ALTER SEQUENCE seq_pid INCREMENT BY 2;

Sequence altered.
```

15. **What is view?**
    - A view is a virtual table based on the result-set of an SQL statement.A view also contains rows and columns like a real table.

16. **What is Indexed view? How to create it?**
    - Views does not store data. When we convert the views to indexed view, they start to store the data. An indexed view has a unique clustered index. The unique clustered index is stored in SQL Server and updated like any other clustered index.

    We can create indexed view by the following query:

```
CREATE or REPLACE VIEW view_name
WITH SCHEMABINDING
AS
SELECT column_name(s)
FROM table_name WHERE condition GROUP BY
```

17. **What are partitioned views and distributed partitioned views?**
    - A partitioned view is a view defined by a "union all" of member tables structured in the same way, but stored separately as multiple tables in either the same instance of SQL Server or in a group of autonomous instances of SQL Server servers, called federated database servers

    - A distributed partitioned view consists of tables participating in the view residing in different databases which reside on different servers or different instances. They use standard SQL statements, along with the key word "union", to pull data from all the distributed servers.

18. **What functions can a view be used to performed?**

    •Restricting data access –Views provide an additional level of table security by restricting access to a predetermined set of rows and columns of a table.

    •Hiding data complexity –A view can hide the complexity that exists in a multiple table join.

    •Simplify commands for the user –Views allows the user to select information from multiple tables without requiring the users to actually know how to perform a join.

    •Store complex queries –Views can be used to store complex queries.

    •Rename Columns –Views can also be used to rename the columns without affecting the base tables provided the number of columns in view must match the number of columns specified in select statement. Thus, renaming helps to hide the names of the columns of the base tables.

•Multiple view facility –Different views can be created on the same table for different users.

19. Describe the functionalities that views support.

•Views can subset data in a table.
•They can join multiple tables into one virtual table.
•Views can provide security and decrease complexity.
•They save space because only their definition is stored.
•They can also be used to create abstraction.
•Materialized views are commonly used in data warehousing. They represent a snapshot of the data from remote sources.

•Views can create other calculated fields based on values in the real underlying tables.

20. What are the restrictions that views have to follow?

•Since a view is a virtual table – columns of the view cannot be renamed. To change anything in the view, the view must be dropped and create again.

•The select statement on the view cannot contain ORDER BY or INTO TEMP

•When a table or view is dropped, any views in the same database are also dropped.

•It is not possible to create an index on a view
•It is not possible to use DELETE to update a view that is defined as a join.

21. Explain the use of cache option in creation of a sequence.

- The CACHE option of the CREATE SEQUENCE statement is a performance and tuning option that directs sqlplus to preallocate a specified number of sequential values in memory.