



Database Management System (DBMS – 204)

Experiment # 10

Creating Views

Student Name: Kabeer Ahmed

Roll Number: SE-19028

Maximum Marks	Performance = 05	Viva = 05	Total = 10
Marks Obtained			
Remarks (if any)			

Experiment evaluated by

Instructor Name: Engr. Adiba Jafar

Signature and Date:

Outcome

After completing this lesson, you should be able to do the following:

1. Describe a view
2. Create, alter the definition of, and drop a view
3. Retrieve data through a view
4. Insert, update, and delete data through a view

VIEW

A view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

Creating table EMPLOYEES

Create table employees AS (select empno AS employee_id,ename AS first_name,sal AS salary,comm AS commission_pct,hiredate AS hire_date,deptno AS department_id from emp);

Creating a View

1. Create a view,EMPVU80, that contains details of employees in department 30.

```
CREATE VIEW empvu80 AS SELECT employee_id, first_name, salary
FROM employees WHERE department_id = 30;
```

View created.

2. Describe the structure of the view by using the iSQL*Plus DESCRIBE command.

```
DESCRIBE empvu80
```

3. Create a view by using column aliases in the subquery.

```
CREATE VIEW salvu50 AS SELECT employee_id, first_name ,salary*12 FROM
employees WHERE department_id = 20;
```

View created.

Retrieving Data from a View

```
SELECT * FROM salvu50;
```

Querying a View

Oracle Server iSQL*Plus USER_VIEWS

```
SELECT * FROM empvu80;
```

```
SELECT employee_id,first_name, salary FROM employees WHERE department_id=50;
```

Modifying a View

1. Modify the EMPVU80 view by using CREATE OR REPLACE VIEW clause. Add an alias for each column name.

```
CREATE OR REPLACE VIEW empvu80
```

```
(id_number, name, sal, department_id)
```

```
AS SELECT employee_id, first_name |,salary, department_id FROM employees WHERE
department_id = 30;
```

View created.

•Column aliases in the CREATE VIEW clause are listed in the same order as the columns in the subquery.

Creating a Complex View

Create a complex view that contains group functions to display values from two tables.

```
CREATE VIEW dept_sum_vu (name, minsal, maxsal, avgsal)
AS SELECT d.department_name, MIN(e.salary),
MAX(e.salary),AVG(e.salary)
FROM employees e, departments d
WHERE e.department_id = d.department_id
GROUP BY d.department_name;
View created.
```

Rules for Performing

DML Operations on a View

1. You can perform DML operations on simple views.
2. You cannot remove a row if the view contains the following:
 - ❖ Group functions
 - ❖ A GROUP BY clause
 - ❖ The DISTINCT keyword
 - ❖ The pseudo column ROWNUM keyword

Denying DML Operations

```
CREATE OR REPLACE VIEW empvu10(employee_number, employee_name, job_title)
AS SELECT employee_id, first_name, job_id FROM employees
WHERE department_id = 10 WITH READ ONLY;
View created.
```

Removing a View

You can remove a view without losing data because a view is based on underlying tables in the database.

```
DROP VIEW view;
DROP VIEW empvu80;
```

View dropped.

LAB# 10

Creating Views

1. Create a view called EMPLOYEES_VU based on the employee numbers, employee names, and department numbers from the EMPLOYEES table.
Change the heading for the employee name to EMPLOYEE.

```
SQL> CREATE OR REPLACE VIEW employees_vu AS
  2  SELECT EMPNO,ENAME EMPLOYEE, DEPTNO
  3  FROM EMP;

View created.
```

2. Display the contents of the EMPLOYEES_VU view.

```
SQL> SELECT * FROM employees_vu;

  EMPNO EMPLOYEE      DEPTNO
-----
  7369 SMITH          20
  7499 ALLEN          30
  7521 WARD           30
  7566 JONES          20
  7654 MARTIN         30
  7698 BLAKE          30
  7782 CLARK          10
  7788 SCOTT          20
  7839 KING           10
  7844 TURNER         30
  7876 ADAMS          20

  EMPNO EMPLOYEE      DEPTNO
-----
  7900 JAMES          30
  7902 FORD           20
  7902 MILLER         10

14 rows selected.
```

3. Select the view name and text from the USER_VIEWS data dictionary view.

```
SQL> SET LONG 600
SQL> SELECT view_name, text
       2 FROM user_views;
```

```
VIEW_NAME
```

```
TEXT
```

```
MVIEW_WORKLOAD
```

```
select
  a.collectionid# as workloadid,
  a.collecttime as import_time,
  a.queryid# as queryid,
  a.application,
  a.cardinality,
  a.resultsized,
  a.qdate as lastuse,
```

```
VIEW_NAME
```

```
TEXT
```

```
SCHEDULER_JOB_ARGS
```

```
SELECT "OWNER", "JOB_NAME", "ARGUMENT_NAME", "ARGUMENT_POSITION", "ARGUMENT_TYPE", "V
ALUE", "ANYDATA_VALUE", "OUT_ARGUMENT" FROM sys.all_scheduler_job_args
```

```
PRODUCT_PRIVS
```

```
SELECT PRODUCT, USERID, ATTRIBUTE, SCOPE,
       NUMERIC_VALUE, CHAR_VALUE, DATE_VALUE, LONG_VALUE
FROM SQLPLUS_PRODUCT_PROFILE
```

```
VIEW_NAME
```

```
TEXT
```

```
WHERE USERID = 'PUBLIC' OR
       USERID LIKE SYS_CONTEXT('USERENV', 'CURRENT_USER')
```

```
EMPLOYEES_VU
```

```
SELECT EMPNO, ENAME EMPLOYEE, DEPTNO
FROM EMP
```

```
11 rows selected.
```

4. Using your EMPLOYEES_VU view, enter a query to display all employee names and department numbers.

```
SQL> SELECT employee, DEPTNO
  2  FROM employees_vu;
```

EMPLOYEE	DEPTNO
SMITH	20
ALLEN	30
WARD	30
JONES	20
MARTIN	30
BLAKE	30
CLARK	10
SCOTT	20
KING	10
TURNER	30
ADAMS	20

EMPLOYEE	DEPTNO
JAMES	30
FORD	20
MILLER	10

```
14 rows selected.
```

5. Create a view named DEPT50 that contains the employee numbers, employee last names, and department numbers for all employees in department 50. Label the view columns EMPNO, EMPLOYEE, and DEPTNO. Do not allow an employee to be reassigned to another department through the view.

```
SQL> CREATE VIEW DEPT50 AS
  2  SELECT EMPNO, ENAME EMPLOYEE, DEPTNO
  3  FROM EMP
  4  WHERE DEPTNO = 50
  5  WITH CHECK OPTION CONSTRAINT emp_dept_50;
```

```
View created.
```

6. Display the structure and contents of the DEPT50 view.

```
SQL> DESCRIBE DEPT50
Name                                         Null?    Type
-----
EMPNO                                         NOT NULL NUMBER
EMPLOYEE                                     VARCHAR2(20)
DEPTNO                                         NUMBER

SQL> SELECT * FROM DEPT50;

no rows selected
```

7. Attempt to reassign Matos to department 80.

```
SQL> UPDATE DEPT50
2 SET DEPTNO = 80
3 WHERE Employee = 'Matos';

0 rows updated.
```

8. Create a view called SALARY_VU based on the employee last names, department names, salaries, and salary grades for all employees. Use the EMPLOYEES, DEPARTMENTS, and JOB_GRADES tables. Label the columns Employee, Department, Salary, and Grade, respectively.

```
SQL> CREATE VIEW salary_vu
2 AS
3 SELECT e.ENAME "Employee",
4        d.DNAME "Department",
5        e.SAL "Salary",
6        j.GRA "Grade"
7 FROM EMP e, DEPT d, SALGRADE j
8 WHERE e.DEPTNO = d.DEPTNO
9 AND e.SAL BETWEEN j.LOWEST_SAL AND j.HIGHEST_SAL;

View created.
```