EXERCISE-11

CREATING VIEWS

After the completion of this exercise, students will be able to do the following:

- Describe a view
- Create, alter the definition of, and drop a view
- Retrieve data through a view
- Insert, update, and delete data through a view
- Create and use an inline view

View

A view is a logical table based on a table or another view. A view contains no data but is like a window through which data from tables can be viewed or changed.The tables on which a view is based are called base tables.

Advantages of Views

- To restrict data access
- To make complex queries easy
- To provide data independence
- To present different views of the same data

Classification of views

1. Simple view
2. Complex view

Feature	Simple	Complex
No. of tables	One	One or more
Contains functions	No	Yes
Contains groups of data	No	Yes
DML operations thr' view	Yes	Not always

Creating a view

Syntax

CREATE OR REPLACE FORCE/NOFORCE VIEW view_name AS Subquery WITH CHECK OPTION CONSTRAINT constraint WITH READ ONLY CONSTRAINT constraint;

FORCE - Creates the view regardless of whether or not the base tables exist.

NOFORCE - Creates the view only if the ase table exist.

WITH CHECK OPTION CONSTRAINT-specifies that only rows accessible to the view can be inserted or updated.

WITH READ ONLY CONSTRAINT-ensures that no DML operations can be performed on the view.

Example: 1 (**Without using Column aliases)**
Create a view EMPVU80 that contains details of employees in department80.

Example 2:

CREATE VIEW empvu80 AS SELECT employee_id, last_name, salary FROM employees
WHERE department_id=80;

Example:1 (**Using column aliases)**

CREATE VIEW salvu50
AS SELECT employee_id,id_number, last_name NAME, salary *12 ANN_SALARY
FROM employees
WHERE department_id=50;

Retrieving data from a view

Example:
SELECT * from salvu50;

Modifying a view

A view can be altered without dropping, re-creating.

Example: (Simple view)

Modify the EMPVU80 view by using CREATE OR REPLACE.

CREATE OR REPLACE VIEW empvu80 (id_number, name, sal, department_id)
AS SELECT employee_id,first_name, last_name, salary, department_id
FROM employees
WHERE department_id=80;

Example: **(complex view)**

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, department d
WHERE e.deparment_id=d.deparment_id
GROUP BY d.department_name;

Rules for performing DML operations on view

- Can perform operations on simple views
- Cannot remove a row if the view contains the following:

- Group functions
- Group By clause
- Distinct keyword

- Cannot modify data in a view if it contains
  - Group functions
  - Group By clause
  - Distinct keyword
  - Columns contain by expressions
  -
- Cannot add data thr' a view if it contains
  - Group functions
  - Group By clause
  - Distinct keyword
  - Columns contain by expressions
  - NOT NULL columns in the base table that are not selected by the view

Example: (Using the WITH CHECK OPTION clause)

```
CREATE OR REPLACE VIEW empvu20
 AS      SELECT *
 FROM employees
 WHERE department_id=20
 WITH CHECK OPTION CONSTRAINT empvu20_ck;
```

Note: Any attempt to change the department number for any row in the view fails because it violates the WITH CHECK OPTION constraint.

Example – (Execute this and note the error)

```
 UPDATE empvu20 SET department_id=10 WHERE employee_id=201;
```

Denying DML operations

Use of WITH READ ONLY option.
Any attempt to perform a DML on any row in the view results in an oracle server error.

Try this code:

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

NEIL DANIEL A - 240701356 - DBMS RECORD

Find the Solution for the following:

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

```
CREATE OR REPLACE VIEW employee_vu AS
SELECT employee_id, last_name
AS employee, department_id
FROM employees;
```

```
View EMPLOYEE_VU created.

Elapsed: 00:00:00.010
```

2. Display the contents of the EMPLOYEES_VU view.

```
SELECT * FROM employee_vu;
```

Download Execution time: 0.007 seconds

	EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
1	1001	Doe	90
2	1002	Smith	60
3	1003	Brown	40
4	1004	White	50

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

```
SELECT view_name, text
FROM user_views
WHERE view_name = 'EMPLOYEE_VU';
```

Download Execution time: 0.032 seconds

	VIEW_NAME	TEXT
1	EMPLOYEE_VU	SELECT employee_id, last_name AS employee, department_id FROM employees

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

```
SELECT employee, department_id
FROM employee_vu;
```

Download Execution time: 0.004 seconds

	EMPLOYEE	DEPARTMENT_ID
1	Doe	90
2	Smith	60
3	Brown	40
4	White	50

5. Create a view named DEPT50 that contains the employee number, 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.

```
CREATE OR REPLACE VIEW dept50 (empno, employee, deptno) AS
SELECT employee_id, last_name, department_id
FROM employees
WHERE department_id = 50
WITH CHECK OPTION CONSTRAINT dept50_ck;
```

```
View DEPT50 created.

Elapsed: 00:00:00.014
```

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

```
SELECT column_name, data_type, data_length
FROM user_tab_columns
WHERE table_name = 'DEPT50';

SELECT * FROM dept50;
```

Download Execution time: 0.003 seconds

	EMPNO	EMPLOYEE	DEPTNO
1	1004	White	50

Download Execution time: 0.195 seconds

	COLUMN_NAME	DATA_TYPE	DATA_LENGTH
1	EMPNO	NUMBER	22
2	EMPLOYEE	VARCHAR2	25
3	DEPTNO	NUMBER	22

7. Attempt to reassign Matos to department 80.

```
UPDATE dept50
SET deptno = 80
WHERE employee = 'Matos';
```

```
0 rows updated.

Elapsed: 00:00:00.005
```

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_GRADE tables. Label the column Employee, Department, salary, and Grade respectively.

```
CREATE OR REPLACE VIEW salary_vu AS
SELECT e.last_name AS Employee,
       d.dept_name AS Department,
       e.salary AS Salary,
       j.grade_level AS Grade
FROM employees e
JOIN department d ON e.department_id = d.dept_id
JOIN job_grade j ON e.salary BETWEEN j.lowest_sal AND j.highest_sal;
```

```
View SALARY_VU created.

Elapsed: 00:00:00.012
```

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	