

EX.NO:5**DATE:18.08.2024****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 |

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 base 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.department_id=d.department_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 the 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;
```

| NAME | NULL? | TYPE |
|----------------|----------|-------------|
| Employee_id | Not null | Number(6) |
| First_Name | | Varchar(20) |
| Last_Name | Not null | Varchar(25) |
| Email | Not null | Varchar(25) |
| Phone_Number | | Varchar(20) |
| Hire_date | Not null | Date |
| Job_id | Not null | Varchar(10) |
| Salary | | Number(8,2) |
| Commission_pct | | Number(2,2) |
| Manager_id | | Number(6) |
| Department_id | | Number(4) |

```
CREATE TABLE Employee (  
    Employee_id    NUMBER(6)    NOT NULL,  
    First_Name     VARCHAR2(20),  
    Last_Name      VARCHAR2(25) NOT NULL,  
    Email          VARCHAR2(25) NOT NULL,  
    Phone_Number   VARCHAR2(20),  
    Hire_date      DATE          NOT NULL,  
    Job_id         VARCHAR2(10) NOT NULL,  
    Salary         NUMBER(8,2),  
    Commission_pct NUMBER(2,2),  
    Manager_id     NUMBER(6),  
    Department_id  NUMBER(4),  
    CONSTRAINT pk_employee PRIMARY KEY (Employee_id)  
);  
  
INSERT INTO Employee (Employee_id, First_Name, Last_Name, Email, Phone_Number,  
Hire_date, Job_id, Salary, Commission_pct, Manager_id, Department_id)  
VALUES (101, 'Amit', 'Sharma', 'amit.sharma@example.com', '9123456789',  
TO_DATE('2021-03-01', 'YYYY-MM-DD'), 'IT_PROG', 60000, NULL, 100, 50);  
  
INSERT INTO Employee (Employee_id, First_Name, Last_Name, Email, Phone_Number,  
Hire_date, Job_id, Salary, Commission_pct, Manager_id, Department_id)  
VALUES (102, 'Sita', 'Patel', 'sita.patel@example.com', '9123456790', TO_DATE('2020-05-12',  
'YYYY-MM-DD'), 'HR_REP', 45000, NULL, 101, 30);  
  
INSERT INTO Employee (Employee_id, First_Name, Last_Name, Email, Phone_Number,  
Hire_date, Job_id, Salary, Commission_pct, Manager_id, Department_id) VALUES (103,  
'Rahul', 'Verma', 'rahul.verma@example.com', '9123456791', TO_DATE('2019-07-23',  
'YYYY-MM-DD'), 'SA_MAN', 70000, 0.1, 101, 80);  
  
INSERT INTO Employee (Employee_id, First_Name, Last_Name, Email, Phone_Number,  
Hire_date, Job_id, Salary, Commission_pct, Manager_id, Department_id) VALUES (104, 'Priya',  
'Singh', 'priya.singh@example.com', '9123456792', TO_DATE('2022-08-17', 'YYYY-MM-DD'),  
'FI_ACCOUNT', 50000, NULL, 102, 50);  
  
INSERT INTO Employee (Employee_id, First_Name, Last_Name, Email, Phone_Number,  
Hire_date, Job_id, Salary, Commission_pct, Manager_id, Department_id) VALUES (105,  
'Rohan', 'Reddy', 'rohan.reddy@example.com', '9123456793', TO_DATE('2020-11-02',  
'YYYY-MM-DD'), 'ST_CLERK', 35000, NULL, 101, 50);  
  
INSERT INTO Employee (Employee_id, First_Name, Last_Name, Email, Phone_Number,  
Hire_date, Job_id, Salary, Commission_pct, Manager_id, Department_id) VALUES (106,  
'Kiran', 'Nair', 'kiran.nair@example.com', '9123456794', TO_DATE('2018-09-25',  
'YYYY-MM-DD'), 'IT_PROG', 65000, NULL, 100, 60);
```

INSERT INTO Employee (Employee_id, First_Name, Last_Name, Email, Phone_Number, Hire_date, Job_id, Salary, Commission_pct, Manager_id, Department_id) VALUES (107, 'Anjali', 'Desai', 'anjali.desai@example.com', '9123456795', TO_DATE('2021-12-05', 'YYYY-MM-DD'), 'HR_REP', 47000, NULL, 102, 30);

| EDIT | EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | PHONE_NUMBER | HIRE_DATE | JOB_ID | SALARY | COMMISSION_PCT | MANAGER_ID | DEPARTMENT_ID |
|------|-------------|------------|-----------|--------------------------|--------------|------------|------------|--------|----------------|------------|---------------|
| | 101 | Amit | Sharma | amit.sharma@example.com | 9123456789 | 03/01/2021 | IT_PROG | 60000 | - | 100 | 50 |
| | 102 | Sita | Patel | sita.patel@example.com | 9123456790 | 05/12/2020 | HR_REP | 45000 | - | 101 | 30 |
| | 103 | Rahul | Verma | rahul.verma@example.com | 9123456791 | 07/23/2019 | SA_MAN | 70000 | .1 | 101 | 80 |
| | 104 | Priya | Singh | priya.singh@example.com | 9123456792 | 08/17/2022 | FL_ACCOUNT | 50000 | - | 102 | 50 |
| | 105 | Rohan | Reddy | rohan.reddy@example.com | 9123456793 | 11/02/2020 | ST_CLERK | 35000 | - | 101 | 50 |
| | 106 | Kiran | Nair | kiran.nair@example.com | 9123456794 | 09/25/2018 | IT_PROG | 65000 | - | 100 | 60 |
| | 107 | Anjali | Desai | anjali.desai@example.com | 9123456795 | 12/05/2021 | HR_REP | 47000 | - | 102 | 30 |

row(s) 1 - 7 of 7

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 VIEW EMPLOYEE_VU AS
SELECT Employee_id,
       First_Name || ' ' || Last_Name AS EMPLOYEE,
       Department_id
FROM Employee;
```

2. Display the contents of the EMPLOYEEES_VU view.

```
SELECT * FROM EMPLOYEE_VU;
```

| EMPLOYEE_ID | EMPLOYEE | DEPARTMENT_ID |
|-------------|--------------|---------------|
| 101 | Amit Sharma | 50 |
| 102 | Sita Patel | 30 |
| 103 | Rahul Verma | 80 |
| 104 | Priya Singh | 50 |
| 105 | Rohan Reddy | 50 |
| 106 | Kiran Nair | 60 |
| 107 | Anjali Desai | 30 |

7 rows returned in 0.03 seconds

[Download](#)

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';
```

| VIEW_NAME | TEXT |
|-------------|---|
| EMPLOYEE_VU | SELECT Employee_id, First_Name ' ' Last_Name AS EMPLOYEE, Department_id FROM Employee |

1 rows returned in 0.02 seconds [Download](#)

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

**SELECT EMPLOYEE, Department_id
FROM EMPLOYEE_VU;**

| EMPLOYEE | DEPARTMENT_ID |
|--------------|---------------|
| Amit Sharma | 50 |
| Sita Patel | 30 |
| Rahul Verma | 80 |
| Priya Singh | 50 |
| Rohan Reddy | 50 |
| Kiran Nair | 60 |
| Anjali Desai | 30 |

7 rows returned in 0.00 seconds

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 VIEW DEPT50 AS
SELECT Employee_id AS EMPNO,
Last_Name AS EMPLOYEE,
Department_id AS DEPTNO
FROM Employee
WHERE Department_id = 50
WITH READ ONLY;**

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

DESC DEPT50;

Object Type **VIEW** Object **DEPT50**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|--------|----------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| DEPT50 | EMPNO | NUMBER | - | 6 | 0 | - | - | - | - |
| | EMPLOYEE | VARCHAR2 | 25 | - | - | - | - | - | - |
| | DEPTNO | NUMBER | - | 4 | 0 | - | ✓ | - | - |
| 1 - 3 | | | | | | | | | |

SELECT * FROM DEPT50;

| EMPNO | EMPLOYEE | DEPTNO |
|-------|----------|--------|
| 101 | Sharma | 50 |
| 104 | Singh | 50 |
| 105 | Reddy | 50 |

3 rows returned in 0.00 seconds

7. Attempt to reassign Matos to department 80.

UPDATE DEPT50

SET DEPTNO = 80

WHERE EMPLOYEE = 'Matos';

This should fail due to the READ ONLY constraint.

ORA-42399: cannot perform a DML operation on a read-only view

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 VIEW SALARY_VU AS

SELECT

e.Last_Name AS Employee,
d.Dept_name AS Department,
e.Salary AS Salary,
kg.Grade_level AS Grade

FROM

Employee e

JOIN

Department d ON e.Department_id = d.Dept_id

JOIN

Job_Grade kg ON e.Salary BETWEEN kg.Lowest_sal AND kg.Highest_sal;

SELECT * FROM SALARY_VU;

| EMPLOYEE | DEPARTMENT | SALARY | GRADE |
|----------|------------|--------|-------|
| Reddy | IT | 35000 | A |
| Patel | HR | 45000 | A |
| Desai | HR | 47000 | A |
| Singh | IT | 50000 | A |
| Sharma | IT | 60000 | B |
| Nair | Finance | 65000 | B |
| Verma | Sales | 70000 | B |

7 rows returned in 0.01 seconds

[Download](#)**RESULT:**

Hence, views are created and executed successfully.