

## 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 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;
```

#### 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;
```

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.

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

2. Display the contents of the EMPLOYEES\_VU view.

Ans: `SELECT * FROM employee_vu;`

3. Select the view name and text from the USER\_VIEWS data dictionary views.

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

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

Ans: `SELECT employee, department_id FROM employee_vu;`

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.

Ans: `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;`

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

Ans: `DESC dept50; SELECT * FROM dept50;`

7. Attempt to reassign Matos to department 80.

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

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.

Ans: `CREATE OR REPLACE VIEW salary_vu (employee, department, salary, grade) AS
SELECT e.last_name, d.department_name, e.salary, j.grade FROM employees e JOIN
departments d ON e.department_id = d.department_id JOIN job_grade j ON e.salary
BETWEEN j.lowest_sal AND j.highest_sal;`

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

## Practice Problems -I Join

### Clauses

Use the Oracle database for problems 1-6.

1. Join the Oracle database locations and departments table using the location\_id column. Limit the results to location 1400 only.

Ans: `SELECT l.location_id, l.city, d.department_id, d.department_name  
FROM locations l JOIN departments d ON l.location_id = d.location_id  
WHERE l.location_id = 1400;`

2. Join DJs on Demand d\_play\_list\_items, d\_track\_listings, and d\_cds tables with the JOIN USING syntax. Include the song ID, CD number, title, and comments in the output.

Ans: `SELECT song_id, cd_number, title, comments FROM d_play_list_items  
JOIN d_track_listings USING (song_id) JOIN d_cds USING (cd_number);`

3. Display the city, department name, location ID, and department ID for departments 10, 20, and 30 for the city of Seattle.

Ans: `SELECT l.city, d.department_name, l.location_id, d.department_id FROM  
departments d JOIN locations l ON d.location_id = l.location_id WHERE  
d.department_id IN (10, 20, 30)  
AND l.city = 'Seattle';`

4. Display country name, region ID, and region name for Americas.

Ans: `SELECT c.country_name, c.region_id, r.region_name FROM countries c  
JOIN regions r ON c.region_id = r.region_id WHERE r.region_name = 'Americas';`

5. Write a statement joining the employees and jobs tables. Display the first and last names, hire date, job id, job title, and maximum salary. Limit the query to those employees who are in jobs that can earn more than \$12,000.

Ans: `SELECT e.first_name, e.last_name, e.hire_date, e.job_id, j.job_title, j.max_salary  
FROM employees e JOIN jobs j ON e.job_id = j.job_id WHERE j.max_salary > 12000;`

## Inner versus Outer Joins

Use the Oracle database for problems 1-7.

1. Return the first name, last name, and department name for all employees including those employees not assigned to a department.

Ans: `SELECT e.first_name, e.last_name, d.department_name FROM employees e  
LEFT JOIN departments d ON e.department_id = d.department_id;`

2. Return the first name, last name, and department name for all employees including those departments that do not have an employee assigned to them.

**Ans:** `SELECT e.first_name, e.last_name, d.department_name FROM departments d  
LEFT JOIN employees e ON e.department_id = d.department_id;`

3. Return the first name, last name, and department name for all employees including those departments that do not have an employee assigned to them and those employees not assigned to a department.

**Ans:** `SELECT e.first_name, e.last_name, d.department_name FROM employees e  
FULL OUTER JOIN departments d ON e.department_id = d.department_id;`

4. Create a query of the DJs on Demand database to return the first name, last name, event date, and description of the event the client held. Include all the clients even if they have not had an event scheduled.

**Ans:** `SELECT c.first_name, c.last_name, e.event_date, e.description FROM d_clients c  
LEFT JOIN d_events e ON c.client_id = e.client_id;`

5. Using the Global Fast Foods database, show the shift description and shift assignment date even if there is no date assigned for each shift description.

**Ans:** `SELECT s.shift_description, a.assignment_date FROM shifts s LEFT JOIN  
shift_assignments a ON s.shift_id = a.shift_id;`

### Self Joins and Hierarchical Queries

For each problem, use the Oracle database.

1. Display the employee's last name and employee number along with the manager's last name and manager number. Label the columns: Employee, Emp#, Manager, and Mgr#, respectively.

**Ans:** `SELECT e.last_name AS Employee, e.employee_id AS Emp#, m.last_name AS  
Manager, m.employee_id AS Mgr# FROM employees e JOIN employees m ON  
e.manager_id = m.employee_id;`

2. Modify question 1 to display all employees and their managers, even if the employee does not have a manager. Order the list alphabetically by the last name of the employee.

**Ans:** `SELECT e.last_name AS Employee, e.employee_id AS Emp#, m.last_name AS  
Manager, m.employee_id AS Mgr# FROM employees e LEFT JOIN employees m ON  
e.manager_id = m.employee_id ORDER BY e.last_name;`

3. Display the names and hire dates for all employees who were hired before their managers, along with their managers' names and hire dates. Label the columns Employee, Emp Hired, Manager, and Mgr Hired, respectively.

**Ans:** `SELECT e.last_name AS Employee, e.hire_date AS "Emp Hired", m.last_name  
AS Manager, m.hire_date AS "Mgr Hired" FROM employees e JOIN employees m  
ON e.manager_id = m.employee_id WHERE e.hire_date < m.hire_date;`

4. Write a report that shows the hierarchy for Lex De Haans department. Include last name, salary, and department id in the report.

**Ans:** `SELECT last_name, salary, department_id FROM employees START WITH  
last_name = 'De Haan' CONNECT BY PRIOR employee_id = manager_id;`

5. What is wrong in the following statement:

Ans: `SELECT last_name, department_id, salary FROM employees START WITH last_name = 'King' CONNECT BY PRIOR manager_id = employee_id;`

6. Create a report that shows the organization chart for the entire employee table. Write the report so that each level will indent each employee 2 spaces. Since Oracle Application Express cannot display the spaces in front of the column, use - (minus) instead.

Ans: `SELECT LPAD('-', LEVEL * 2, '-') || last_name AS "Employee Chart" FROM employees START WITH manager_id IS NULL CONNECT BY PRIOR employee_id = manager_id;`

7. Re-write the report from 6 to exclude De Haan and all the people working for him.

Ans: `SELECT LPAD('-', LEVEL * 2, '-') || last_name AS "Employee Chart" FROM employees START WITH manager_id IS NULL CONNECT BY PRIOR employee_id = manager_id AND last_name != 'De Haan';`

### Oracle Equijoin and Cartesian Product

1. Create a Cartesian product that displays the columns in the d\_play\_list\_items and the d\_track\_listings in the DJs on Demand database.

Ans: `SELECT * FROM d_play_list_items, d_track_listings;`

2. Correct the Cartesian product produced in question 1 by creating an equijoin using a common column.

Ans: `SELECT * FROM d_play_list_items p JOIN d_track_listings t ON p.song_id = t.song_id;`

3. Write a query to display the title, type, description, and artist from the DJs on Demand database.

Ans: `SELECT title, type, description, artist FROM d_track_listings;`

4. Rewrite the query in question 3 to select only those titles with an ID of 47 or 48.

Ans: `SELECT title, type, description, artist FROM d_track_listings WHERE song_id IN (47, 48);`

5. Write a query that extracts information from three tables in the DJs on Demand database, the d\_clients table, the d\_events table, and the d\_job\_assignments table.

Ans: `SELECT c.first_name, c.last_name, e.event_date, e.description, j.job_title FROM d_clients c JOIN d_events e ON c.client_id = e.client_id JOIN d_job_assignments j ON e.event_id = j.event_id;`

## Group Functions

1. Define and give an example of the seven group functions: AVG, COUNT, MAX, MIN, STDDEV, SUM, and VARIANCE.

**Ans: AVG()**

Example: SELECT AVG(salary) FROM employees;

**COUNT()**

• Example: SELECT COUNT(\*) FROM employees;

**MAX()**

• Example: SELECT MAX(salary) FROM employees;

**MIN()**

• Example: SELECT MIN(salary) FROM employees;

**STDDEV()**

• Example: SELECT STDDEV(salary) FROM employees;

**SUM()**

• Example: SELECT SUM(salary) FROM employees;

**VARIANCE()**

• Example: SELECT VARIANCE(salary) FROM employees;

2. Create a query that will show the average cost of the DJs on Demand events. Round to two decimal places.

**Ans: SELECT ROUND(AVG(event\_cost), 2) AS avg\_event\_cost FROM d\_events;**

3. Find the average salary for Global Fast Foods staff members whose manager ID is 19.

**Ans: SELECT AVG(salary) AS avg\_salary FROM staff WHERE manager\_id = 19;**

4. Find the sum of the salaries for Global Fast Foods staff members whose IDs are 12 and 9.

**Ans: SELECT SUM(salary) AS total\_salary FROM staff WHERE staff\_id IN (12, 9);**

5. Using the Oracle database, select the lowest salary, the most recent hire date, the last name of the person who is at the top of an alphabetical list of employees, and the last name of the person who is at the bottom of an alphabetical list of employees. Select only employees who are in departments 50 or 60

**Ans: SELECT MIN(salary) AS lowest\_salary, MAX(hire\_date) AS latest\_hire, MIN(last\_name) AS first\_alpha, MAX(last\_name) AS last\_alpha FROM employees WHERE department\_id IN (50, 60);**

6. Your new Internet business has had a good year financially. You have had 1,289 orders this year. Your customer order table has a column named total\_sales. If you submit the following query, how many rows will be returned? SELECT sum(total\_sales) fFROM orders;

**Ans: 1**

7. You were asked to create a report of the average salaries for all employees in each division of the company. Some employees in your company are paid hourly instead of by salary. When you ran the report, it seemed as though the averages were not what you expected—they were much higher than you thought! What could have been the cause?

**Ans:** Hourly employees may have been excluded or misrepresented in the salary column, or their hourly rates were treated as monthly salaries, inflating the average.

8. Employees of Global Fast Foods have birth dates of July 1, 1980, March 19, 1979, and March 30, 1969. If you select MIN(birthdate), which date will be returned?

**Ans:** 1969-03-30

9. Create a query that will return the average order total for all Global Fast Foods orders from January 1, 2002, to December 21, 2002.

**Ans:** `SELECT AVG(order_total) AS avg_order FROM orders WHERE order_date BETWEEN TO_DATE('01-JAN-2002', 'DD-MON-YYYY') AND TO_DATE('21-DEC-2002', 'DD-MON-YYYY');`

10. What was the hire date of the last Oracle employee hired?

**Ans:** `SELECT MAX(hire_date) AS last_hire_date FROM employees;`

11. Your new Internet business has had a good year financially. You have had 1,289 orders this year. Your customer order table has a column named total\_sales. If you submit the following query, how many rows will be returned? `SELECT sum(total_sales) FROM orders;`

**Ans:** 1Row

Evaluation Procedure	Marks awarded
Practice Evaluation (5)	5
Viva(5)	5
Total (10)	10
Faculty Signature	<i>RJM</i>

## Practice Problems -II

### COUNT, DISTINCT, NVL

- How many songs are listed in the DJs on Demand D\_SONGS table?

Ans: `SELECT COUNT(*) FROM d_songs;`

- In how many different location types has DJs on Demand had venues?

Ans: `SELECT COUNT(DISTINCT loc_type) FROM d_venues;`

- The d\_track\_listings table in the DJs on Demand database has a song\_id column and a cd\_number column. How many song IDs are in the table and how many different CD numbers are in the table?

Ans: `SELECT COUNT(song_id) AS total_songs, COUNT(DISTINCT cd_number) AS unique_cds FROM d_track_listings;`

- How many of the DJs on Demand customers have email addresses?

Ans: `SELECT COUNT(email) FROM d_clients WHERE email IS NOT NULL;`

- Some of the partners in DJs on Demand do not have authorized expense amounts (auth\_expense\_amt). How many partners do have this privilege?

Ans: `SELECT COUNT(auth_expense_amt) FROM d_partners WHERE auth_expense_amt IS NOT NULL;`

- What values will be returned when the statement below is issued?

ID	type	shoe_color
456	oxford	brown
463	sandal	tan
262	heel	black
433	slipper	tan

`SELECT COUNT(shoe_color),  
COUNT(DISTINCT shoe_color)  
FROM shoes;`

Ans: 4, 3 [ 4 total entries, 3 unique colors (brown, tan, black)]

- Create a query that will convert any null values in the auth\_expense\_amt column on the DJs on Demand D\_PARTNERS table to 100000 and find the average of the values in this column. Round the result to two decimal places.

Ans: `SELECT ROUND(AVG(NVL(auth_expense_amt, 100000)), 2) AS avg_expense  
FROM d_partners;`

- Which of the following statements is/are TRUE about the following query? `SELECT AVG(NVL(selling_bonus, 0.10))  
FROM bonuses;`

- The datatypes of the values in the NVL clause can be any datatype except date data. **True**
- If the selling\_bonus column has a null value, 0.10 will be substituted. **True**
- There will be no null values in the selling\_bonus column when the average is calculated. **True**
- This statement will cause an error. There cannot be two functions in the SELECT statement. **False**

9. Which of the following statements is/are TRUE about the following query? `SELECT DISTINCT colors, sizes FROM items;`
- Each color will appear only once in the results set. False
  - Each size will appear only once in the results set. False
  - Unique combinations of color and size will appear only once in the results set. True
  - Each color and size combination will appear more than once in the results set. False

### Using GROUP BY and HAVING Clauses

- In the SQL query shown below, which of the following are true about this query?
- Kimberly Grant would not appear in the results set.**
  - The GROUP BY clause has an error because the manager\_id is not listed in the SELECT clause.
  - Only salaries greater than 16001 will be in the result set.
  - Names beginning with Ki will appear after names beginning with Ko.**
  - Last names such as King and Kochhar will be returned even if they don't have salaries > 16000.

**Ans:** `SELECT last_name, AVG(salary) FROM employees WHERE last_name LIKE 'K%' GROUP BY manager_id, last_name HAVING MAX(salary) > 16000 ORDER BY last_name DESC;`

- Each of the following SQL queries has an error. Find the error and correct it. Use Oracle Application Express to verify that your corrections produce the desired results.

- `SELECT manager_id ,FROM employees WHERE AVG(salary) < 16000 GROUP BY manager_id;`

**Ans:** `SELECT manager_id, AVG(salary) FROM employees GROUP BY manager_id HAVING AVG(salary) < 16000;`

- `SELECT cd_number, COUNT(title) FROM d_cds WHERE cd_number < 93;`

**Ans:** `SELECT cd_number, COUNT(title) FROM d_cds WHERE cd_number < 93 GROUP BY cd_number;`

- `SELECT ID, MAX(ID), artist AS Artist FROM d_songs WHERE duration IN('3 min', '6 min', '10 min') HAVING ID < 50 GROUP BY ID;`

**Ans:** `SELECT ID, MAX(ID), artist AS Artist FROM d_songs WHERE duration IN ('3 min', '6 min', '10 min') GROUP BY ID, artist HAVING ID < 50;`

- `SELECT loc_type, rental_fee AS Fee FROM d_venues WHERE id < 100 GROUP BY "Fee" ORDER BY 2;`

**Ans:** `SELECT loc_type, rental_fee AS Fee FROM d_venues WHERE id < 100 GROUP BY loc_type, rental_fee ORDER BY rental_fee;`

3. Rewrite the following query to accomplish the same result:

`SELECT DISTINCT MAX(song_id) FROM d_track_listings WHERE track IN (1, 2, 3);`

**Ans:** `SELECT MAX(song_id) FROM d_track_listings WHERE track IN (1, 2, 3);`

4. Indicate True or False

a. If you include a group function and any other individual columns in a SELECT clause, then each individual column must also appear in the GROUP BY clause. [T]

b. You can use a column alias in the GROUP BY clause. [F]

c. The GROUP BY clause always includes a group function. [F]

5. Write a query that will return both the maximum and minimum average salary grouped by department from the employees table.

**Ans:** `SELECT MAX(avg_salary) AS max_avg, MIN(avg_salary) AS min_avg  
FROM ( SELECT department_id, AVG(salary) AS avg_salary FROM  
employees GROUP BY department_id);`

6. Write a query that will return the average of the maximum salaries in each department for the employees table.

**Ans:** `SELECT AVG(max_salary) AS avg_max_salary  
FROM (SELECT department_id, MAX(salary) AS max_salary FROM employees GROUP  
BY department_id);`

### Using Set Operators

1. Name the different Set operators?

#### **UNION**

- Description: Combines results of two queries and removes duplicates.

#### **UNION ALL**

- Description: Combines results of two queries and retains duplicates.

#### **INTERSECT**

- Description: Returns rows common to both queries.

#### **MINUS**

- Description: Returns rows from the first query that are not in the second.

2. Write one query to return the employee\_id, job\_id, hire\_date, and department\_id of all employees and a second query listing employee\_id, job\_id, start\_date, and department\_id from the job\_history table and combine the results as one single output. Make sure you suppress duplicates in the output.

**Ans:** `SELECT employee_id, job_id, hire_date AS start_date, department_id FROM employees  
UNION SELECT employee_id, job_id, start_date, department_id  
FROM job_history;`

3. Amend the previous statement to not suppress duplicates and examine the output. How many extra rows did you get returned and which were they? Sort the output by employee\_id to make it easier to spot.

**Ans:** `SELECT employee_id, job_id, hire_date AS start_date, department_id FROM  
employees UNION ALL SELECT employee_id, job_id, start_date, department_id`

**FROM job\_history ORDER BY employee\_id;**

4. List all employees who have not changed jobs even once. (Such employees are not found in the job\_history table)

**Ans: SELECT employee\_id, first\_name, last\_name FROM employees WHERE employee\_id NOT IN (SELECT employee\_id FROM job\_history);**

5. List the employees that HAVE changed their jobs at least once.

**Ans: SELECT DISTINCT e.employee\_id, e.first\_name, e.last\_name FROM employees e JOIN job\_history j ON e.employee\_id = j.employee\_id;**

1. Using the UNION operator, write a query that displays the employee\_id, job\_id, and salary of ALL present and past employees. If a salary is not found, then just display a 0 (zero) in its place.

**Ans: SELECT employee\_id, job\_id, salary FROM employees UNION SELECT employee\_id, job\_id, NVL(salary, 0) FROM job\_history;**

Evaluation Procedure	Marks awarded
Practice Evaluation (5)	5
Viva(5)	5
Total (10)	10
Faculty Signature	P. S. M