- 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;
```

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.

```
creating view employee_vu as select emp_id,
f_name || ' ' || l_name as emp, dept_id from
    emp;
```

2. Display the contents of the EMPLOYEES_VU view.

```
select * from employee_vu;
```

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

select view_name, text from user_views
where view_name = 'EMPLOYEES_VU';

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

select employee, dept_id from emp_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.

create view dept50 as select employee_id as empno,

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

desc dept50; select * from dept50;

7. Attempt to reassign Matos to department 80.

update dept50 set deptno=80 where emp='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.

create view salary_vu as select e.last_name as emp, d.dept_name as dept, e.salary, j.grade_level as grade from employees e join dept d using (dept_id) join job_grades j on e.salary between j.lowest_sal and j.highest_sal;

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.

```
select d.dept_id d.dept_name, l.location_id, l.city
from dept d join locations l using (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.

```
select p.song_id, t.cd_number, c.title, p.comments
from d_play_list_items p join d_track_listings t
using (song_id) join d_cds c 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.

```
select l.city, d.dept_name, d.location_id, d.dept_id
from dept d join locations l using (location_id) where
city = 'Seattle' and dept_id IN (10, 20, 30);
```

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

```
select c.country_name, c.region_id, r.region_name
from countries c join regions r using (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.

```
select e.first_name, e.last_name, e.hiredate,
e.job_id, j.job_title, j.max_salary from
employees e join jobs j using(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.

```
select e.first_name, e.last_name, d.dept_name from
employees e left join departments d using (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.

```
select e.first_name, e.last_name, d.department_name
from employees e right join departments d using
(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.

```
select e.first_name, e.last_name, d.department_name
from employees e full outer join departments d
using (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.

```
select c.first_name, c.last_name, e.event_date, e.
description from d_clients c left join d_events e
using (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.

```
select s.shift_description, a.shift_assignment_da
```

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.

```
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.

```
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.

```
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.

```
select l_name, salary, dept_id, level from employees start with
l_name = 'De Haan' connect by prior employee_id = manager_id;
```

5. What is wrong in the following statement:

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

```
select l_name, dept_id, salary from employees ~~sat~~ start with
l_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.

```
select lpad('-', 2*(level-1)) || l_name as employee employee_id,
manager_id 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.

```
select lpad('-', 2*(level-1)) || l_name as employee from employees
where employee_id from employees start with l_name = 'De Haan'
connect by prior employee_id = manager_id) start with manager_id
is null connect by prior employee_id = manager_id
```

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.

```
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.

```
select * from d_play_list_items p join d_track_listings t on
p.track_id = t.track_id;
```

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

```
select t.title, t.type, t.description, t.artist from d_track_listings
t;
```

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

```
select t.title, t.type, t.description, t.artist from d_track_
listings t where t.track_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.

```
select c.client_name, e.event_title, j.job_id, j.employee_id from
d_clients c joint d_events e on c.client_id = e.client_i
= 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.

AVG calculates average select avg(salary) from employees

COUNT Counts rows select count(*) from employees;

MAX Finds maximum select max(salary) from employees;

MIN Finds minimum select min(salary) from employees;

STDDEV standard deviation select stddev(salary) from employees;

SUM Total sum select sum(salary) from employees;

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

```
select round(avg(event_cost),2) as "Average Event Cost"
From d_events;
```

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

```
Select avg(salary) as "Average Salary" from employees
where manager_ix =19;
```

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

```
Select sum(salary) as "Total Salary" from
employees where emp_id In (12,9);
```

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

```
Select min(sal) as "Lowest Salary", max(hire_da
as "Latest Hire Date", min(last_name) as "first
Alphabetically", MAX(last_name) AS "Last Alphabetically"
```

5. 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;
```

1 row

6. 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?

The report included both salaried and hourly employees. Hourly employees pay rate is annualized incorrectly inflating the average.

7. 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?

July 1, 1980
March 19, 1979
March 30, 1969

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

```
Select avg(total) as "Average orders total"
from orders where order_date between '01-JAN-2002'
and '21-DEC-2002';
```

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

```
select f_name, hire_date from employees where
hire_date = (select Max(hire_date) from employees);
```

10. 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;
```

1 row

Practice Problems -II

COUNT, DISTINCT, NVL

1. How many songs are listed in the DJs on Demand D_SONGS table?

select count(*) as "Number of Songs" from d_songs;

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

select count(distinct loc_type) as "Number of location types" from d_venues;

3. 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?

select count(song_id) as "Total song IDs",
count(distinct cd_number) as "different CD Numbers"
from d_track_listings;

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

select count(email) as "customers with email"
from d_clients where email is not null;

5. Some of the partners in DJs on Demand do not have authorized expense amounts (auth_expense_amt). How many partners do have this privilege?

select count(auth_expense) as "partners" from
d_partners where auth_expense_amt is not null;

6. 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;

count(shoe_color) count(distinct shoe_color)
4 3

7. 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.

select round (Avg (NVL (auth_expense_amt, 100000)),2)
as "Average Expense" from d_partners;

8. Which of the following statements is/are TRUE about the following query?

SELECT AVG(NVL(selling_bonus, 0.10))
FROM bonuses;

_____a. The datatypes of the values in the NVL clause can be any datatype except date data.
_____b. If the selling_bonus column has a null value, 0.10 will be substituted.
_____c. There will be no null values in the selling_bonus column when the average is calculated.
_____ d. This statement will cause an error. There cannot be two functions in the SELECT statement.

b, c

9. Which of the following statements is/are TRUE about the following query?

SELECT DISTINCT colors, sizes
FROM items;

_____a. Each color will appear only once in the results set.
_____b. Each size will appear only once in the results set.
_____c. Unique combinations of color and size will appear only once in the results set.
_____d. Each color and size combination will appear more than once in the results set.

c

Using GROUP BY and HAVING Clauses

1. In the SQL query shown below, which of the following are true about this query?

a. Kimberly Grant would not appear in the results set.

b. The GROUP BY clause has an error because the manager_id is not listed in the SELECT clause.

c. Only salaries greater than 16001 will be in the result set.

d. Names beginning with Ki will appear after names beginning with Ko.

e. Last names such as King and Kochhar will be returned even if they don't have salaries > 16000.

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

a, c, e

2. 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.

a.
```
SELECT manager_id
FROM employees
WHERE AVG(salary) <16000
GROUP BY manager_id;
```

```
select manager_id, avg(salary) from employees
group by manager_id having avg(salary) < 16000;
```

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

```
select cd_number, count(title) from d_cds
where cd_number < 93 group by cd_number;
```

c.
```
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;
```

```
select id, artist, MAX(id) as max_id from d_songs
where duration in ('3 min', '6 min', '10 min') group by id,
artist having id < 50;
```

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

```
select loc_type, rental_fee as fee from d_venues
where id < 100 group by loc_type, rental_fee order
by loc_type;
```

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);
```

select max (distinct 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.

Select max(avg(salary)) as "Max Avg Salary"
min(avg(salary)) as "Min Avg Salary" from
employees group by dept-id

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

select avg(max-salary) as "Avg of Max Salary"
from (select dept-id, max(salary) as max-salary
from employees group by dept-id);

Using Set Operators

1. Name the different Set operators?

union, union all, intersect, minus

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.

```
select emp_id, job_id, hire_date, dept_id from
employees union select emp_id, job_id, start_date,
dept_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.

one extra row (employee 176 - job_id SA_REP)

There is one extra row on employee 176 with a job_id of SA_REP.

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

```
select emp_id, l_name from emp minus select
emp_id, l_name from emp e join_history j on e.employee
= j.employee_id;
```

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

```
select emp_id, l_name from emp e join job_
history j on e.employee_id = j.employee_id;
```

6. 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.

```
select emp_id, job_id, salary from employees union
select emp_id, job_id, nvl(salary, 0) from job_history;
```