#### ****EMPLOYEE\_ID FIRST\_NAME LAST\_NAME EMAIL PHONE\_NUMBER HIRE\_DATE JOB\_ID SALARY COMMISSION\_PCT MANAGER\_ID DEPARTMENT\_ID****

#### ****----------- -------------------- ------------------------- ------------------------- -------------------- --------- ---------- ---------- -------------- ---------- -------------****

#### ****198 Donald OConnell DOCONNEL 650.507.9833 21-JUN-99 SH\_CLERK 2600 124 50****

#### ****199 Douglas Grant DGRANT 650.507.9844 13-JAN-00 SH\_CLERK 2600 124 50****

#### ****200 Jennifer Whalen JWHALEN 515.123.4444 17-SEP-87 AD\_ASST 4400 101 10****

#### ****201 Michael Hartstein MHARTSTE 515.123.5555 17-FEB-96 MK\_MAN 13000 100 20****

#### ****202 Pat Fay PFAY 603.123.6666 17-AUG-97 MK\_REP 6000 201 20****

#### ****203 Susan Mavris SMAVRIS 515.123.7777 07-JUN-94 HR\_REP 6500 101 40****

#### ****204 Hermann Baer HBAER 515.123.8888 07-JUN-94 PR\_REP 10000 101 70****

#### ****205 Shelley Higgins SHIGGINS 515.123.8080 07-JUN-94 AC\_MGR 12000 101 110****

#### ****206 William Gietz WGIETZ 515.123.8181 07-JUN-94 AC\_ACCOUNT 8300 205 110****

#### ****100 Steven King SKING 515.123.4567 17-JUN-87 AD\_PRES 24000 90****

#### ****101 Neena Kochhar NKOCHHAR 515.123.4568 21-SEP-89 AD\_VP 17000 100 90****

#### ****102 Lex De Haan LDEHAAN 515.123.4569 13-JAN-93 AD\_VP 17000 100 90****

#### ****103 Alexander Hunold AHUNOLD 590.423.4567 03-JAN-90 IT\_PROG 9000 102 60****

#### ****104 Bruce Ernst BERNST 590.423.4568 21-MAY-91 IT\_PROG 6000 103 60****

#### ****105 David Austin DAUSTIN 590.423.4569 25-JUN-97 IT\_PROG 4800 103 60****

#### ****106 Valli Pataballa VPATABAL 590.423.4560 05-FEB-98 IT\_PROG 4800 103 60****

#### ****107 Diana Lorentz DLORENTZ 590.423.5567 07-FEB-99 IT\_PROG 4200 103 60****

#### ****108 Nancy Greenberg NGREENBE 515.124.4569 17-AUG-94 FI\_MGR 12000 101 100****

#### ****109 Daniel Faviet DFAVIET 515.124.4169 16-AUG-94 FI\_ACCOUNT 9000 108 100****

#### ****110 John Chen JCHEN 515.124.4269 28-SEP-97 FI\_ACCOUNT 8200 108 100****

#### ****111 Ismael Sciarra ISCIARRA 515.124.4369 30-SEP-97 FI\_ACCOUNT 7700 108 100****

#### ****112 Jose Manuel Urman JMURMAN 515.124.4469 07-MAR-98 FI\_ACCOUNT 7800 108 100****

#### ****113 Luis Popp LPOPP 515.124.4567 07-DEC-99 FI\_ACCOUNT 6900 108 100****

#### ****114 Den Raphaely DRAPHEAL 515.127.4561 07-DEC-94 PU\_MAN 11000 100 30****

#### ****115 Alexander Khoo AKHOO 515.127.4562 18-MAY-95 PU\_CLERK 3100 114 30****

#### ****116 Shelli Baida SBAIDA 515.127.4563 24-DEC-97 PU\_CLERK 2900 114 30****

#### ****117 Sigal Tobias STOBIAS 515.127.4564 24-JUL-97 PU\_CLERK 2800 114 30****

#### ****118 Guy Himuro GHIMURO 515.127.4565 15-NOV-98 PU\_CLERK 2600 114 30****

#### ****119 Karen Colmenares KCOLMENA 515.127.4566 10-AUG-99 PU\_CLERK 2500 114 30****

#### ****120 Matthew Weiss MWEISS 650.123.1234 18-JUL-96 ST\_MAN 8000 100 50****

#### ****121 Adam Fripp AFRIPP 650.123.2234 10-APR-97 ST\_MAN 8200 100 50****

#### ****122 Payam Kaufling PKAUFLIN 650.123.3234 01-MAY-95 ST\_MAN 7900 100 50****

#### ****123 Shanta Vollman SVOLLMAN 650.123.4234 10-OCT-97 ST\_MAN 6500 100 50****

#### ****124 Kevin Mourgos KMOURGOS 650.123.5234 16-NOV-99 ST\_MAN 5800 100 50****

#### ****125 Julia Nayer JNAYER 650.124.1214 16-JUL-97 ST\_CLERK 3200 120 50****

#### ****126 Irene Mikkilineni IMIKKILI 650.124.1224 28-SEP-98 ST\_CLERK 2700 120 50****

#### ****127 James Landry JLANDRY 650.124.1334 14-JAN-99 ST\_CLERK 2400 120 50****

#### ****128 Steven Markle SMARKLE 650.124.1434 08-MAR-00 ST\_CLERK 2200 120 50****

#### ****129 Laura Bissot LBISSOT 650.124.5234 20-AUG-97 ST\_CLERK 3300 121 50****

#### ****130 Mozhe Atkinson MATKINSO 650.124.6234 30-OCT-97 ST\_CLERK 2800 121 50****

#### ****131 James Marlow JAMRLOW 650.124.7234 16-FEB-97 ST\_CLERK 2500 121 50****

#### ****132 TJ Olson TJOLSON 650.124.8234 10-APR-99 ST\_CLERK 2100 121 50****

#### ****133 Jason Mallin JMALLIN 650.127.1934 14-JUN-96 ST\_CLERK 3300 122 50****

#### ****134 Michael Rogers MROGERS 650.127.1834 26-AUG-98 ST\_CLERK 2900 122 50****

#### ****135 Ki Gee KGEE 650.127.1734 12-DEC-99 ST\_CLERK 2400 122 50****

#### ****136 Hazel Philtanker HPHILTAN 650.127.1634 06-FEB-00 ST\_CLERK 2200 122 50****

#### ****137 Renske Ladwig RLADWIG 650.121.1234 14-JUL-95 ST\_CLERK 3600 123 50****

#### ****138 Stephen Stiles SSTILES 650.121.2034 26-OCT-97 ST\_CLERK 3200 123 50****

#### ****139 John Seo JSEO 650.121.2019 12-FEB-98 ST\_CLERK 2700 123 50****

#### ****140 Joshua Patel JPATEL 650.121.1834 06-APR-98 ST\_CLERK 2500 123 50****

#### ****141 Trenna Rajs TRAJS 650.121.8009 17-OCT-95 ST\_CLERK 3500 124 50****

#### ****142 Curtis Davies CDAVIES 650.121.2994 29-JAN-97 ST\_CLERK 3100 124 50****

#### ****143 Randall Matos RMATOS 650.121.2874 15-MAR-98 ST\_CLERK 2600 124 50****

#### ****144 Peter Vargas PVARGAS 650.121.2004 09-JUL-98 ST\_CLERK 2500 124 50****

#### ****145 John Russell JRUSSEL 011.44.1344.429268 01-OCT-96 SA\_MAN 14000 .4 100 80****

#### ****146 Karen Partners KPARTNER 011.44.1344.467268 05-JAN-97 SA\_MAN 13500 .3 100 80****

#### ****147 Alberto Errazuriz AERRAZUR 011.44.1344.429278 10-MAR-97 SA\_MAN 12000 .3 100 80****

#### ****148 Gerald Cambrault GCAMBRAU 011.44.1344.619268 15-OCT-99 SA\_MAN 11000 .3 100 80****

#### ****149 Eleni Zlotkey EZLOTKEY 011.44.1344.429018 29-JAN-00 SA\_MAN 10500 .2 100 80****

#### ****150 Peter Tucker PTUCKER 011.44.1344.129268 30-JAN-97 SA\_REP 10000 .3 145 80****

#### ****151 David Bernstein DBERNSTE 011.44.1344.345268 24-MAR-97 SA\_REP 9500 .25 145 80****

#### ****152 Peter Hall PHALL 011.44.1344.478968 20-AUG-97 SA\_REP 9000 .25 145 80****

#### ****153 Christopher Olsen COLSEN 011.44.1344.498718 30-MAR-98 SA\_REP 8000 .2 145 80****

#### ****154 Nanette Cambrault NCAMBRAU 011.44.1344.987668 09-DEC-98 SA\_REP 7500 .2 145 80****

#### ****155 Oliver Tuvault OTUVAULT 011.44.1344.486508 23-NOV-99 SA\_REP 7000 .15 145 80****

#### ****156 Janette King JKING 011.44.1345.429268 30-JAN-96 SA\_REP 10000 .35 146 80****

#### ****157 Patrick Sully PSULLY 011.44.1345.929268 04-MAR-96 SA\_REP 9500 .35 146 80****

#### ****158 Allan McEwen AMCEWEN 011.44.1345.829268 01-AUG-96 SA\_REP 9000 .35 146 80****

#### ****159 Lindsey Smith LSMITH 011.44.1345.729268 10-MAR-97 SA\_REP 8000 .3 146 80****

#### ****160 Louise Doran LDORAN 011.44.1345.629268 15-DEC-97 SA\_REP 7500 .3 146 80****

#### ****161 Sarath Sewall SSEWALL 011.44.1345.529268 03-NOV-98 SA\_REP 7000 .25 146 80****

#### ****162 Clara Vishney CVISHNEY 011.44.1346.129268 11-NOV-97 SA\_REP 10500 .25 147 80****

#### ****163 Danielle Greene DGREENE 011.44.1346.229268 19-MAR-99 SA\_REP 9500 .15 147 80****

#### ****164 Mattea Marvins MMARVINS 011.44.1346.329268 24-JAN-00 SA\_REP 7200 .1 147 80****

#### ****165 David Lee DLEE 011.44.1346.529268 23-FEB-00 SA\_REP 6800 .1 147 80****

#### ****166 Sundar Ande SANDE 011.44.1346.629268 24-MAR-00 SA\_REP 6400 .1 147 80****

#### ****167 Amit Banda ABANDA 011.44.1346.729268 21-APR-00 SA\_REP 6200 .1 147 80****

#### ****168 Lisa Ozer LOZER 011.44.1343.929268 11-MAR-97 SA\_REP 11500 .25 148 80****

#### ****169 Harrison Bloom HBLOOM 011.44.1343.829268 23-MAR-98 SA\_REP 10000 .2 148 80****

#### ****170 Tayler Fox TFOX 011.44.1343.729268 24-JAN-98 SA\_REP 9600 .2 148 80****

#### ****171 William Smith WSMITH 011.44.1343.629268 23-FEB-99 SA\_REP 7400 .15 148 80****

#### ****172 Elizabeth Bates EBATES 011.44.1343.529268 24-MAR-99 SA\_REP 7300 .15 148 80****

#### ****173 Sundita Kumar SKUMAR 011.44.1343.329268 21-APR-00 SA\_REP 6100 .1 148 80****

#### ****174 Ellen Abel EABEL 011.44.1644.429267 11-MAY-96 SA\_REP 11000 .3 149 80****

#### ****175 Alyssa Hutton AHUTTON 011.44.1644.429266 19-MAR-97 SA\_REP 8800 .25 149 80****

#### ****176 Jonathon Taylor JTAYLOR 011.44.1644.429265 24-MAR-98 SA\_REP 8600 .2 149 80****

#### ****177 Jack Livingston JLIVINGS 011.44.1644.429264 23-APR-98 SA\_REP 8400 .2 149 80****

#### ****178 Kimberely Grant KGRANT 011.44.1644.429263 24-MAY-99 SA\_REP 7000 .15 149****

#### ****179 Charles Johnson CJOHNSON 011.44.1644.429262 04-JAN-00 SA\_REP 6200 .1 149 80****

#### ****180 Winston Taylor WTAYLOR 650.507.9876 24-JAN-98 SH\_CLERK 3200 120 50****

#### ****181 Jean Fleaur JFLEAUR 650.507.9877 23-FEB-98 SH\_CLERK 3100 120 50****

#### ****182 Martha Sullivan MSULLIVA 650.507.9878 21-JUN-99 SH\_CLERK 2500 120 50****

#### ****183 Girard Geoni GGEONI 650.507.9879 03-FEB-00 SH\_CLERK 2800 120 50****

#### ****Basic Questions****

1. Display the full name of all employees by concatenating their first and last names. Use an alias "Full Name" for the column.

Select first\_name || ‘ ‘ || last\_name as full\_name from employees;

1. Show the job ID in uppercase for all employees.

Select lower(job\_id) from employees;

1. Display the first three characters of the last names of all employees.

Select substr(first\_name,1,3) from employees;

1. Find the length of the first names of all employees.

Select length(first\_name) from employees;

1. Replace all occurrences of 'a' in the first names with 'A'.

Select replace(first\_name, ‘a’,’A’) from employees;

1. Convert the hire date of employees to a format: 'DD-Mon-YYYY'.

Select to\_char(hire\_date,’DD-MON-YYYY’) from employees;

1. Extract the year from the hire date for each employee.

select first\_name,to\_char(hire\_date,'YYYY') from employees;

1. Round the salary of employees to the nearest thousand.

select first\_name, Round(salary,-3) from employees;

1. Display the email addresses of employees in lowercase.

select lower(email) from employees;

1. Display the first name, last name, and phone number concatenated with a hyphen ("-").

select first\_name || '-' || last\_name || '-' || phone\_number from employees;

#### ****Intermediate Questions****

1. Calculate the total salary for all employees.
   1. select sum(salary) as total\_salary from employees;
2. Find the average salary of employees in department 50.
   1. select avg(salary) as average\_salary from employees where department\_id = 50;
3. Display the maximum and minimum hire date in the table.
   1. Select max(hire\_date),min(hire\_date) from employees;
4. Count the number of employees in each department.
   1. select count(\*),department\_id from employees group by department\_id;
5. Find the total number of employees whose salaries are greater than 5000.
   1. select count(\*) from employees where salary > 5000;
6. List the department ID and the highest salary in each department.
   1. select max(salary),department\_id from employees group by department\_id;
7. Calculate the total commission earned by employees (use commission\_pct with their salary).
   1. select sum(salary \* nvl(commission\_pct,0)) from employees;
   2. select sum(commission\_pct),sum(salary),sum(commission\_pct+salary) from employees;
8. Show the number of employees hired in each year.
   1. select count(\*),to\_char(hire\_date,'YYYY') from employees group by to\_char(hire\_date,'YYYY');
9. Display the count of employees who have "ST\_CLERK" as their job ID.
   1. select count(\*) from employees where job\_id = 'ST\_CLERK';
10. Find the total salary paid to employees in department 80.
    1. Select sum(salary) from employees where department\_id = 80;

#### ****Advanced Questions****

1. List employees whose hire date is earlier than 01-Jan-1990.
   1. select first\_name from employees where hire\_date < '01-JAN-1990';
2. Show the total number of employees reporting to each manager.
   1. select count(\*),manager\_id from employees group by manager\_id;
3. List employees who have not been assigned to any manager (i.e., manager\_id is null).
   1. select first\_name from employees where manager\_id is null;
4. Display the departments with more than 5 employees.
   1. select count(\*),department\_id from employees group by department\_id having count(\*) > 4;

### String Functions

1. Write a query to display the FIRST\_NAME and LAST\_NAME concatenated with a space between them, along with their EMAIL in uppercase.

select first\_name || ' ' || last\_name as full\_name, upper(email) from employees;

1. Retrieve the FIRST\_NAME of employees where the third letter in their name is 'n'.

select first\_name from employees where instr(first\_name,'n') = 3;

1. Display the EMAIL of all employees, but replace all occurrences of the letter 'a' with '@'.

select replace(email,'A','@') from employees;

1. Extract the first three letters of the JOB\_ID and concatenate them with the last two digits of the HIRE\_DATE (year part).

select substr(job\_id,1,3) || to\_char(hire\_date,'YY') from employees;

1. Display the LAST\_NAME in reverse order for all employees.

select reverse(last\_name) from employees;

### Number Functions

1. Retrieve the SALARY and round it to the nearest thousand for all employees.

select round(salary,-3) from employees;

1. Display the SALARY and its absolute difference from 5000.

select salary-5000 from employees;

1. Write a query to show the SALARY increased by 15% and truncated to two decimal places.

select trunc(salary+salary\*.15,-2) from employees;

1. Find the square root of the SALARY for employees with JOB\_ID as SA\_REP.

select power(salary,2) from employees where job\_id = 'SA\_REP';

1. For employees in department 50, calculate the remainder when their SALARY is divided by 3000.

select mod(salary,3000) from employees where department\_id = 50;

### Date Functions

1. Display the FIRST\_NAME, HIRE\_DATE, and the number of months they have worked for the company till date.

select months\_between(sysdate,hire\_date) from employees;

1. Retrieve the names of employees who were hired in the year 1997.

select first\_name,hire\_Date from employees where to\_char(hire\_date,'YYYY') = 1997;

1. Display the FIRST\_NAME, HIRE\_DATE, and the day of the week they were hired.

select first\_name,to\_char(hire\_date,'DY') from employees;

1. Write a query to calculate the next Friday from the current date for each employee's hire date.
2. Find the HIRE\_DATE of employees and add 100 days to it.

select to\_date(hire\_date)+100 from employees;

### Conditional Expressions

1. Display the FIRST\_NAME and SALARY, and show "Above Average" or "Below Average" based on whether their salary is above or below the company average.

select first\_name,salary,case when salary>6400 then 'Above average'

2 when salary<6400 then 'Below average'

3 end as status

4 from employees;

1. Write a query to display FIRST\_NAME and JOB\_ID, and replace SA\_REP with "Sales Representative" and all other job IDs with "Other".
2. Retrieve the FIRST\_NAME and SALARY of employees who earn a SALARY less than 5000; otherwise, display their salary reduced by 10%.
3. Display the FIRST\_NAME and SALARY, and assign a "Low", "Medium", or "High" label based on the salary range (<5000, 5000–10000, >10000).
4. Display FIRST\_NAME and MANAGER\_ID, but replace any null MANAGER\_ID with "No Manager".

1. Find employees who earn more than any employee in department 50.

select first\_name,salary from employees where salary > (select max(salary) from employees where department\_id = 50);

2. Find employees who earn less than the highest salary in department 30.

select first\_name,salary from employees where salary < (select max(salary) from employees where department\_id = 30);

3. Find employees whose hire date is earlier than all employees in department 60.

select first\_name,hire\_date from employees where hire\_date < (select min(hire\_date) from employees where department\_id = 60);

4. Find employees whose salary is greater than the salary of 'Alexander'.

select first\_name,last\_name,salary from employees where salary > (select avg(salary) from employees where first\_name = 'Alexander');

5. Find employees who share the same job ID as 'Steven King'.

select first\_name,job\_id from employees where job\_id = (select job\_id from employees where first\_name = 'Steven' and last\_name = 'King');

6. Find employees who were hired on the same date as 'Neena Kochhar'.

select first\_name,hire\_date from employees where hire\_date = (select hire\_date from employees where first\_name = 'Neena' and last\_name = 'Kochhar');

7. Find employees who earn more than the employee with ID 101.

select first\_name,salary,department\_id from employees where salary > (select max(salary) from employees where department\_id = 101);

8. Find employees working in the same department as 'Nancy Greenberg'.

select first\_name,department\_id from employees where department\_id = (select department\_id from employees where first\_name = 'Nancy' and last\_name = 'Greenberg');

9. Find employees who have a salary equal to any employee in department 20.

10. Find employees whose job ID is different from all employees in department 100.

Q waqtd the position of char @ in email id

Select instr(email,’@’,1) from customers;

Q waqtd the customers whose mail id contain the word gmail

Select full\_name from customers where instr(email, ‘gmail’, 1) != 0;

Q waqtd the customer details if their mobile number contains the seq 123

Select \* from customer where instr(phno,’123’, 1) != 0;

Q waqtd swap the first name and last name of customer

Q waqtd the text before the symbol @ in the mail\_id column

Select substr(email\_id, 1, instr(email\_id, [‘@’,](mailto:‘@gmail.com’,) 1)-1) from customers;

Q waqtd the digit which is present after the second - in phone number

Select substr(phno,instr(phno,’-’, 1,2)+1) from customers;

### ****Basic Subqueries****

Find the **name of the department** where employee "ALLEN" works.

select dname from dept where deptno = (select deptno from emp where ename = 'ALLEN');

Retrieve the **salary of SMITH's manager**.

SELECT ENAME,SAL FROM EMP WHERE EMPNO = (SELECT MGR FROM EMP WHERE ENAME = 'SMITH');

List all employees who earn **more than JONES**.

SELECT ENAME,SAL FROM EMP WHERE SAL > (SELECT SAL FROM EMP WHERE ENAME = 'JONES');

Find the **location** of the department where employee "WARD" works.

SELECT LOC FROM DEPT WHERE DEPTNO = (SELECT DEPTNO FROM EMP WHERE ENAME = 'WARD');

Display the **highest salary** from the emp table.

SELECT MAX(SAL) FROM EMP;

### ****Single-row Subqueries****

Find the **name of the employee** who earns the **highest salary**.

SELECT ENAME FROM EMP WHERE SAL = (SELECT MAX(SAL) FROM EMP);

Retrieve the **department name** where the highest-paid employee works.

SELECT DNAME FROM DEPT WHERE DEPTNO = (SELECT DEPTNO FROM EMP WHERE SAL = (SELECT MAX(SAL) FROM EMP));

Display the **name of the manager** of employee "SCOTT".

SELECT ENAME FROM EMP WHERE EMPNO = (SELECT MGR FROM EMP WHERE ENAME = 'SCOTT');

Find the **name of the department** where the employee with the lowest salary works.

SELECT DNAME FROM DEPT WHERE DEPTNO = (SELECT DEPTNO FROM EMP WHERE SAL = (SELECT MIN(SAL) FROM EMP));

Find the **hiring date of the employee** with the second-highest salary.

SELECT HIREDATE FROM EMP WHERE SAL = (SELECT MAX(SAL) FROM EMP WHERE SAL < (SELECT MAX(SAL) FROM EMP));

### ****Multi-row Subqueries****

List the **employees** working in the same department as "SCOTT".

SELECT ENAME FROM EMP WHERE DEPTNO = (SELECT DEPTNO FROM EMP WHERE ENAME = 'SCOTT');

Retrieve the names of employees whose salaries are equal to **any salary in department 30**.

SELECT ename FROM emp WHERE sal IN (SELECT sal FROM emp WHERE deptno = 30);E.

Display the **departments** where at least one employee earns more than 3000.

SELECT dname FROM dept WHERE deptno IN (SELECT deptno FROM emp WHERE sal > 3000);

List the **employees** who have the same job as "FORD".

SELECT ENAME,JOB FROM EMP WHERE JOB = (SELECT JOB FROM EMP WHERE ENAME = 'FORD');

Find the **employees** whose salaries are higher than the average salary in their department.

SELECT ENAME,SAL,DEPTNO FROM EMP WHERE SAL > (SELECT AVG(SAL) FROM EMP WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT));

### ****Correlated Subqueries****

Retrieve the **names of employees** who earn more than the average salary of their department.

List the **employees** whose salaries are higher than their managers.

Display the **department name and employee name** for employees earning the highest salary in each department.

Retrieve the **name of the employee** who was hired before their manager.

Find the **departments** where all employees earn less than 2000.

### Basic INSTR + Other String Functions

Write a query to find the employees whose FIRST\_NAME contains the character 'E' as the third letter and convert the result to uppercase using UPPER.

Retrieve all employees where the length of the LAST\_NAME minus the position of 'A' (using INSTR) is greater than 5.

List employees whose reversed EMAIL (using REVERSE) contains the substring 'com' at the third position.

Find employees where the substring extracted (using SUBSTR) from the position of 'N' in their FIRST\_NAME (using INSTR) to the end of the name is more than 3 characters long.

### Combining INSTR and REPLACE

Replace the first occurrence of 'a' in the EMAIL with '#' using REPLACE and find the position of '#' using INSTR.

Retrieve employees where the position of 'e' in the LAST\_NAME (using INSTR) matches the position of a replaced '#' in the EMAIL (using REPLACE).

### Case Sensitivity and Transformation

List employees where the lowercase version of their FIRST\_NAME (using LOWER) contains the string 'john' at any position using INSTR.

Find employees where the uppercase version of their LAST\_NAME (using UPPER) has the letter 'A' as the fifth character using INSTR.

### Using LENGTH with INSTR

Retrieve employees where the position of the last 'o' in their EMAIL (using INSTR) is equal to the total length of the email minus 3 (using LENGTH).

Find employees whose FIRST\_NAME has a substring starting at the position of 'L' (using INSTR) and extending to the end that is exactly 4 characters long.

### Grouping and Aggregation

Group employees by the first letter of their EMAIL (using SUBSTR and INSTR) and display the count of employees in each group.

Find the employee in each JOB\_ID group whose LAST\_NAME has the earliest occurrence of the letter 'S' using INSTR and ORDER BY.

### Sorting with INSTR

Retrieve all employees and sort them in ascending order based on the position of 'K' in their FIRST\_NAME (using INSTR).

Display all employees where the substring extracted from their EMAIL (starting at the position of '.' using INSTR) is sorted alphabetically.

### Advanced Subqueries and Conditions

Find employees whose FIRST\_NAME contains the same position of 'A' as their LAST\_NAME contains 'B' using a subquery with INSTR.

Identify employees where the difference in position between the first 'M' in EMAIL and LAST\_NAME is less than 5 using INSTR.

### Combining Multiple Functions

Retrieve employees where the length of their EMAIL is greater than the position of the first 'X' (using INSTR) plus 5.

List employees where replacing the letter 'R' in LAST\_NAME with '#' results in a string where '#' occurs at the same position as the first 'A' in FIRST\_NAME.

### Nested INSTR

Find employees whose EMAIL has the position of 'E' (using INSTR) nested inside another INSTR to find if 'com' starts after it.

### ****5/1/25****

### ****String Functions****

1. Write a query to find the length of department names in the DEPARTMENTS table.

Select length(department\_name) from departments;

1. Concatenate the first name and last name of employees from the EMPLOYEES table.

select first\_name || ' ' || last\_name from employees;

1. Convert the country name from the COUNTRIES table to uppercase.

select upper(country\_name) from countries;

1. Convert the job titles in the JOBS table to lowercase.

select lower(job\_title) from jobs;

1. Display the job title in the JOBS table with the first letter capitalized.

select initcap(job\_title) from jobs;

1. Reverse the names of employees from the EMPLOYEES table.

select reverse(first\_name || ' ' || last\_name) from employees;

1. Extract the first three characters of department names from the DEPARTMENTS table.

select substr(department\_name,1,3) from departments;

1. Replace the word "Manager" with "Lead" in job titles from the JOBS table.

select replace(job\_title,'Manager','Lead') from jobs;

1. Find the position of the substring "IT" in department names in the DEPARTMENTS table.

select instr(department\_name,'IT') from departments;

1. Pad the department names from the DEPARTMENTS table to a length of 20 characters with asterisks (\*) on the left.

select lpad(department\_name,20,'\*') from departments;

### ****Mathematical Functions****

1. Find the remainder when employee salaries are divided by 1000 in the EMPLOYEES table.

select mod(salary,1000) from employees;

1. Raise the job ID length in the JOBS table to the power of 2.

Select power(length(job\_id),2) from jobs;

1. Round the average salary of employees to 2 decimal places.

select round(avg(salary),-2) from employees;

1. Truncate the maximum salary of employees to the nearest thousand.

select trunc(max(salary),-3) from employees;

### ****NVL and NVL2****

1. Replace NULL commission percentages with 0 in the EMPLOYEES table.

select nvl(commission\_pct,0) from employees;

1. Display "Has Commission" if the commission is not NULL, otherwise display "No Commission" for employees.

select nvl2(commission\_pct,'has commission','no commission') from employees;

### ****CASE Function****

1. Categorize employees into "High Salary" and "Low Salary" based on a salary threshold of 5000.

select salary,case when salary > 5000 then 'High salary' when salary < 5000 then 'low salary' end from employees;

1. Display "Manager" for employees in department 10 and "Staff" for others in the EMPLOYEES table.

select department\_id, case when department\_id = 10 then 'Manager' else 'Staff' end as Role from employees;

### ****Subqueries****

1. Find the name of the employee earning the highest salary.

select first\_name from employees where salary in (select max(salary) from employees);

1. List employees who earn more than the average salary.

select first\_name from employees where salary > (select avg(salary) from employees);

1. Find the department name for the employee with the highest salary.

select department\_name from departments where department\_id in (select department\_id from employees where salary = (select max(salary) from employees));

1. Retrieve job IDs for employees who work in department 20 using a subquery.

select job\_id from jobs where job\_id in (select job\_id from employees where department\_id = 20);

### ****ALL and ANY Operators****

1. List employees whose salary is greater than all employees in department 30.

select \* from employees where salary > all(select max(salary) from employees where department\_id = 30);

1. Find employees whose salary is greater than any employee in department 50.

select \* from employees where salary > any(select max(salary) from employees where department\_id = 50);

### ****Nested Subqueries****

1. Find the second-highest salary in the EMPLOYEES table.

select \* from employees where salary in (select max(salary) from employees where salary < (select max(salary) from employees));

1. Retrieve the name of the employee with the second-lowest salary in department 60.

select \* from employees where salary in (select min(salary) from employees where salary > (select min(salary) from employees where department\_id = 60));

### ****Combination Questions****

1. Find the length of the names of employees earning the highest salary.

select length(first\_name || last\_name) from employees where salary = (select max(salary) from employees);

1. Replace NULL values in the DEPARTMENTS table with "Not Assigned" and display the department names.

select department\_name,manager\_id, case when manager\_id is null then 'not assigned' else 'assigned' end as manager\_id\_status from departments;

select department\_name,manager\_id,nvl2(manager\_id,'not assigned','assigned') from departments;

1. Display job titles where the salary of employees is greater than the average salary for all employees.

select job\_title from jobs where job\_id in (select job\_id from employees where salary > (select avg(salary) from employees));

1. Display the last date of the month when the highest-paid employee was hired.

select last\_day(hire\_date) from employees where salary = (select max(salary) from employees);

### ****Complex Queries****

1. List the departments where the sum of employee salaries exceeds 50,000.

select department\_name from departments where department\_id in (select department\_id from employees group by department\_id having sum(salary) > 50000);

1. Retrieve employee names and their respective department names using a subquery.

Wrong - select first\_name,(select department\_name from departments where department\_id in (select department\_id from employees)) as department\_names from employees;

1. Find employees whose job title contains "Manager" and display their salaries rounded to the nearest hundred.

select first\_name,job\_id,round(salary,-2) from employees where job\_id in (select job\_id from jobs where job\_title like '%Manager');

1. Use a subquery to find the average salary for each job title, and display job titles where the average exceeds 10,000.

select job\_title from jobs where job\_id in (select job\_id from employees group by job\_id having avg(salary) > 10000);

1. Retrieve the first name and last name of employees, and concatenate them with a space between, from the EMPLOYEES table.

select first\_name || ' ' || last\_name from employees;

1. Find the names of employees whose salary is greater than the average salary of all employees.

select first\_name from employees where salary > (select avg(salary) from employees);

1. Display the job title and its length from the JOBS table.

select length(job\_title) from jobs;

1. List the employees whose hire date is earlier than 1st January 2010.

select first\_name,hire\_date from employees where hire\_date<'01-JAN-2010';

1. Replace NULL values in the commission percentage column of the EMPLOYEES table with 0.

select nvl(commission\_pct,0) from employees;

1. Retrieve the department name and the total number of employees in each department.

select department\_name,(select count(\*),department\_id from employees group by department\_id) as total\_employee from departments;

1. Find the second-highest salary in the EMPLOYEES table.

select ename from emp e1 where(select count(distinct e2.sal) from emp e2 where e1.sal < e2.sal) in 1;

1. Display employee names and their respective department names using a subquery.

select e.ename,d.dname from emp e inner join dept d on e.deptno = d.deptno;

1. Find employees who were hired in the year 2020.
2. Display the last name of employees in reverse order from the EMPLOYEES table.

select ename from(select rownum,emp.\* from emp order by rownum desc) where rownum = 1;

1. Calculate the number of months between the hire date and today for each employee.

select months\_between(sysdate,hiredate) from emp;

1. List all employees whose names start with the letter "A" and end with "N".

select ename from emp where ename like 'A%N';

1. Display the job title and its uppercase version from the JOBS table.
2. Retrieve the names of employees whose salary is greater than any employee in department 50.

select first\_name,salary from employees where salary >all (select salary from employees where department\_id = 50);

1. Find the position of the substring "CLERK" in job titles from the JOBS table.

select instr(job\_title,'Clerk',1) from jobs;

1. Pad the department names from the DEPARTMENTS table to 15 characters with underscores (\_) on the right.
2. Find employees whose salary is greater than the salary of employee "James".

select first\_name,salary from employees where salary >all (select salary from employees where first\_name = 'James');

1. Retrieve the department name for the department with the highest total salary.

select department\_name from departments d where d.department\_id in

(select e.department\_id from employees e where salary in

(select max(salary) from employees e where e.department\_id in

(select e.department\_id from employees e)));

1. Calculate the remainder when each employee’s salary is divided by 500.

select mod(salary,500) from employees;

1. Replace the word "Assistant" with "Associate" in job titles from the JOBS table.

select replace(job\_title,'Assistant','Associate') from jobs;

1. Display the names of employees who have been working for more than 5 years.
2. Retrieve the current system date and time.
3. Find the employee with the lowest salary in each department.
4. Display the names of employees who were hired on the last day of the month.
5. Extract the first three characters of each country name from the COUNTRIES table.
6. Find employees whose salary is equal to the highest salary in the EMPLOYEES table.
7. List employees whose hire date is in December.
8. Find the total salary paid to employees in each department.
9. Display employee names along with their salaries, categorized as "High", "Medium", or "Low" using a CASE statement.
10. Retrieve the current timestamp with the time zone.
11. List employees whose first name contains exactly five characters.
12. Display department names where the total number of employees is greater than 10.
13. Find the number of employees hired in each year.
14. Display the last name of employees in lowercase and their first names in uppercase.
15. Find the names of employees whose job title is "Manager" and who have been working for more than 10 years.
16. Retrieve the second-highest salary in each department.
17. List the department name and the number of employees earning more than 5000 in that department.
18. Display the first name, last name, and salary of employees earning more than 10,000, ordered by salary in descending order.
19. Find employees who were hired in the same year as employee "Allen".
20. Retrieve the maximum salary, minimum salary, and average salary of employees in each department.
21. List employees whose job ID starts with "SA" and ends with "REP".
22. Display the first and last names of employees who do not have a manager.
23. Find the department where the highest-paid employee works.
24. Replace leading and trailing spaces in department names with underscores (\_).
25. Retrieve the last name of employees whose last name contains the substring "SON".
26. Find employees whose salaries are greater than all employees in department 20.
27. Display the names of employees who have the same job title as employee "Susan".
28. List the first names of employees hired after "Susan" but earning less than "Susan".
29. Display the names of employees who do not belong to any department.
30. Find employees whose salaries are greater than the average salary of their department.

### ****Easy Questions****

1. Find the name of the manager of the employee with EMPNO 7369.
2. List the employees who do not have a manager.
3. Find the number of employees under manager with EMPNO 7839.
4. Retrieve the names of all managers in the company.
5. Find the details of employees whose salary is greater than their manager's salary.

### ****Moderate Questions****

1. Display the name of the employee and their manager for all employees in department 30.
2. Identify employees whose manager is in the same department.

SELECT ENAME FROM EMP WHERE EMPNO IN (SELECT MGR FROM EMP WHERE DEPTNO IN (SELECT DEPTNO FROM EMP));

1. Retrieve the details of employees where the manager's job is 'MANAGER'.

SELECT \* FROM EMP WHERE MGR IN (SELECT EMPNO FROM EMP WHERE JOB = 'MANAGER');

1. Find the department number where the manager earns the highest salary.

SELECT DEPTNO FROM EMP WHERE SAL > (SELECT MAX(SAL) FROM EMP WHERE JOB='MANAGER');

1. Display the name and salary of employees whose manager earns less than 2000.

select e1.ename,e1.sal,e2.ename,e2.sal from emp e1 join emp e2 on e1.mgr = e2.empno where e2.sal<2000;

### ****Hard Questions****

1. Find the details of managers who manage more than three employees.
2. List the employee names who work for a manager with a salary greater than 3000.
3. Find the department where the average salary of employees is greater than the salary of their manager.
4. Retrieve the names of employees whose managers joined after them.
5. Display the names of employees who have the same job as their manager.

### ****Challenging Questions****

1. Identify employees who are both managers and report to another manager.
2. Find the total salary of all employees under the manager whose name is 'KING'.
3. List employees who are the only subordinate of their manager.
4. Retrieve the details of employees who do not have a manager in their department.
5. Display the details of employees whose manager's name starts with 'B'.

### ****Scenario-Based Questions****

1. Find the employee names and their manager names where the manager is located in 'NEW YORK'.
2. Identify employees who earn more than the average salary of all their manager’s subordinates.
3. Retrieve the details of employees whose manager has a higher commission than them.
4. List employees working under the manager with the maximum subordinates.
5. Display the department name and the number of employees under each manager.

### ****Advanced Analytical Questions****

1. Find employees who are managed by managers with more than two subordinates in different departments.
2. Retrieve the details of employees whose managers were hired after January 1, 1981.
3. List the employee names whose manager’s salary is more than the average salary of all employees in the company.
4. Display the details of employees whose salary is equal to their manager's salary.
5. Find the names of employees whose managers do not have any subordinates other than them.
6. Find the employees who earn more than the average salary in their department.
7. Retrieve the names of employees who have the same job as the employee with EMPNO = 7839.
8. Get the employees who have a higher salary than the average salary of employees in the same department.
9. List employees who were hired before the employee with EMPNO = 7788.
10. Find the employees who earn a salary higher than any employee in department 30.
11. List employees who do not have any subordinates (i.e., no employees reporting to them).
12. Retrieve the employee details where the salary is greater than the salary of any employee in department 10.
13. Find the employee with the second highest salary in the company.
14. Get employees who earn less than the average salary of employees in department 20.
15. Find the employees whose salaries are greater than the highest salary in department 30.
16. List the employees who earn the lowest salary in each department.
17. Retrieve the employees who earn more than the average salary in department 30.
18. Find the employee(s) who earn more than the average salary of all employees.
19. Get the employees who have a salary greater than the salary of employee 7839.
20. List the employees who have the same job as any employee in department 20.
21. Find employees who have been hired after the employee with EMPNO = 7499.
22. Retrieve the names of employees who earn more than 2000 and belong to department 20.
23. List employees who have the highest salary in their department.
24. Retrieve employees whose hire date is after the employee with EMPNO = 7902.
25. Get the names of employees who earn more than the lowest salary in department 30.
26. Find the employees who earn the Nth highest salary in the company.
27. Get the employee(s) whose salary is higher than the average salary of employees in department 40.
28. Find the employee(s) who belong to departments located in 'NEW YORK'.
29. List the employees who have been hired before the employee with the highest salary.
30. Retrieve the employees whose salary is higher than the average salary of employees in the same department.
31. Find employees whose salaries are lower than the salary of employee 7844.
32. Get the names of employees who earn more than 1500 but less than the highest salary in department 30.
33. List employees who work in the same department as employee 7782.
34. Retrieve employees whose hire date is before the hire date of employee 7698.
35. Find the employees who earn more than the salary of the employee with EMPNO = 7369.

### JOIN Queries

1. Write a query to fetch employee names and their department names using the EMPLOYEES and DEPARTMENTS tables.

select first\_name,department\_name from employees inner join departments on employees.department\_id = departments.department\_id;

1. Retrieve job titles and the names of employees holding those jobs using the EMPLOYEES and JOBS tables.

select first\_name,job\_title from employees inner join jobs on employees.job\_id = jobs.job\_id;

1. Fetch department names and their locations by joining DEPARTMENTS and LOCATIONS.

select department\_name,city from departments inner join locations on departments.location\_id = locations.location\_id;

1. List all employees along with their job history using EMPLOYEES and JOB\_HISTORY.

select first\_name,job\_history.\* from employees inner join job\_history on employees.employee\_id = job\_history.employee\_id;

1. Show the region name for each country using REGIONS and COUNTRIES.

select \* from regions inner join countries on regions.region\_id = countries.region\_id;

### Outer JOIN Queries

1. Fetch all departments and their employees, including departments without employees.

select \* from employees right join departments on employees.department\_id = departments.department\_id;

1. List all jobs and employees assigned to them, including jobs with no employees.

select \* from employees right join jobs on employees.job\_id = jobs.job\_id;

1. Retrieve all regions and the countries within them, including regions with no countries.

select \* from regions right join countries on regions.region\_id = countries.region\_id;

### Subqueries

1. Find employees who earn more than the average salary.

select first\_name,salary from employees where salary > (select avg(salary) from employees);

1. List department names where the number of employees is greater than 5.

select department\_name from departments where department\_id > all(select department\_id from employees group by department\_id having count(\*) > 5);

1. Fetch job titles for employees who work in the Sales department.

SELECT JOB\_TITLE FROM JOBS WHERE JOB\_ID IN (SELECT JOB\_ID FROM EMPLOYEES WHERE DEPARTMENT\_ID = (SELECT DEPARTMENT\_ID FROM DEPARTMENTS WHERE DEPARTMENT\_NAME = 'Sales'));

### Advanced Queries

1. Display the name of employees who joined before 2015.

select first\_name from employees where hire\_date < '01-JAN-2015';

1. List all employees and their managers (self-join on the EMPLOYEES table).
2. Retrieve countries where the number of employees exceeds 10.

select country\_name from countries where country\_id in (select country\_id from locations where location\_id in (select location\_id from departments where department\_id in (select department\_id from employees group by department\_id having count(\*) > 10)));

1. Find departments that do not have any employees.

select distinct(employees.department\_id) from employees right join departments on employees.department\_id = departments.department\_id;

### ****Basic to Intermediate JOIN Queries****

1. Retrieve the country name and region name for each country.

select country\_name,region\_name from countries inner join regions on countries.region\_id = regions.region\_id;

1. List all departments along with their location city.

select department\_name,city from departments inner join locations on departments.location\_id = locations.location\_id;

1. Show the job title and department name of all employees.

select first\_name,department\_name,job\_title from employees inner join departments on employees.department\_id = departments.department\_id inner join jobs on employees.job\_id = jobs.job\_id;

1. List the department names and the names of their locations for all departments.

select department\_name,state\_province from departments inner join locations on departments.location\_id = locations.location\_id;

1. Retrieve the names of employees along with the job title and minimum salary for their job.

select first\_name,job\_title,min\_salary from employees inner join jobs on employees.job\_id = jobs.job\_id;

1. Find the names of all employees along with the name of their region.

select first\_name,region\_name from employees inner join departments on employees.department\_id = departments.department\_id inner join locations on departments.location\_id = locations.location\_id inner join countries on locations.country\_id = countries.country\_id inner join regions on countries.region\_id = regions.region\_id;

1. Show the location address and the city for each department.

select department\_name,street\_address,city from departments inner join locations on departments.location\_id = locations.location\_id;

1. Display the job ID, job title, and the count of employees in each job.

select count(\*),employees.job\_id,job\_title from employees inner join jobs on employees.job\_id = jobs.job\_id group by employees.job\_id,job\_title;

1. List all employees who work in the same city as their department's location.

select first\_name,city from employees,departments,locations where employees.department\_id = departments.department\_id and departments.location\_id = locations.location\_id;

### ****Advanced JOIN Queries****

1. Display the total salary of employees grouped by job title and region name.

select sum(salary),job\_title,region\_name from employees inner join jobs on employees.job\_id = jobs.job\_id inner join departments on employees.department\_id = departments.department\_id inner join locations on departments.location\_id = locations.location\_id inner join countries on locations.country\_id = countries.country\_id inner join regions on countries.region\_id = regions.region\_id group by jobs.job\_title,region\_name;

1. Find the names of employees working in departments located in the same country as their office.

SELECT FIRST\_NAME,COUNTRY\_NAME FROM EMPLOYEES INNER JOIN DEPARTMENTS ON EMPLOYEES.DEPARTMENT\_ID = DEPARTMENTS.DEPARTMENT\_ID INNER JOIN LOCATIONS ON DEPARTMENTS.LOCATION\_ID = LOCATIONS.LOCATION\_ID INNER JOIN COUNTRIES ON LOCATIONS.COUNTRY\_ID = COUNTRIES.COUNTRY\_ID;

1. Retrieve the details of employees whose job title is the same as any job in the JOB\_HISTORY table.

select first\_name,employees.job\_id,job\_title from employees inner join jobs on employees.job\_id = jobs.job\_id inner join job\_history on jobs.job\_id = job\_history.job\_id;

1. List all regions along with the names of countries in each region.

select country\_name,region\_name from countries inner join regions on countries.region\_id = regions.region\_id;

1. Show all department names that have employees with salaries above the average salary of their job.

SELECT DISTINCT DEPARTMENT\_NAME FROM EMPLOYEES INNER JOIN DEPARTMENTS ON EMPLOYEES.DEPARTMENT\_ID = DEPARTMENTS.DEPARTMENT\_ID WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES INNER JOIN JOBS ON EMPLOYEES.JOB\_ID = JOBS.JOB\_ID);

1. Retrieve the employee names, their job title, and the location address of their department.

select first\_name,job\_title,street\_address from employees inner join jobs on employees.job\_id = jobs.job\_id inner join departments on employees.department\_id = departments.department\_id inner join locations on departments.location\_id = locations.location\_id;

1. List employees and their job history details (if available); otherwise, show employee details only.

select first\_name,job\_history.\*,salary from employees inner join job\_history on employees.employee\_id = job\_history.employee\_id;

1. Find the names of employees who are currently in the same department as they were in their job history.

select first\_name,employees.department\_id,job\_history.department\_id from employees inner join job\_history on employees.department\_id = job\_history.department\_id;

1. Retrieve the job title and department name for employees who joined after January 1, 2020.

select first\_name,hire\_date,job\_title,department\_name from employees inner join departments on employees.department\_id = departments.department\_id inner join jobs on employees.job\_id = jobs.job\_id where hire\_date > '01-JAN-2020';

### ****Complex Multi-Table JOIN Queries****

1. Display all employee names and their regions, even if the employee’s department or region details are missing.

select e.first\_name, r.region\_name,d.department\_id,c.country\_id,l.location\_id,r.region\_id from employees e right join departments d on e.department\_id = d.department\_id right join locations l on d.location\_id = l.location\_id right join countries c on l.country\_id = c.country\_id right join regions r on c.region\_id = r.region\_id;

1. List the job titles and their corresponding departments for employees who changed departments in the past.

select e.first\_name,j.job\_title,d.department\_id,jh.department\_id from employees e inner join departments d on e.department\_id = d.department\_id inner join jobs j on e.job\_id = j.job\_id inner join job\_history jh on e.employee\_id = jh.employee\_id;

1. Show the department name, city, and the names of employees working in that department.

select e.first\_name,d.department\_name,l.city from employees e inner join departments d on e.department\_id = d.department\_id inner join locations l on d.location\_id = l.location\_id;

1. Retrieve all countries and their employees, even if no employees work in a country.

select e.first\_name,c.country\_name from employees e right join departments d on e.department\_id = d.department\_id right join locations l on d.location\_id = l.location\_id right join countries c on l.country\_id = c.country\_id;

1. Display all jobs and their current employees, even if some jobs have no employees.

select e.first\_name,e.job\_id,j.job\_title

from employees e right join jobs j

on e.job\_id = j.job\_id

where employee\_id not in (select employee\_id from job\_history);

1. Retrieve the names of departments and the cities they are located in, along with the count of employees in each department.

select count(e.first\_name),d.department\_name,l.city from employees e inner join departments d on e.department\_id = d.department\_id inner join locations l on d.location\_id = l.location\_id group by d.department\_name,l.city;

1. Show all employees who earn less than the maximum salary for their job title.

select e.first\_name,e.salary,j.max\_salary from employees e inner join jobs j on e.job\_id = j.job\_id where j.max\_salary>e.salary;

1. List the regions and the total salary of all employees in each region.

SELECT r.region\_name, SUM(e.salary) AS total\_salary

FROM employees e

INNER JOIN departments d ON e.department\_id = d.department\_id

INNER JOIN locations l ON d.location\_id = l.location\_id

INNER JOIN countries c ON l.country\_id = c.country\_id

INNER JOIN regions r ON c.region\_id = r.region\_id

GROUP BY r.region\_name;

1. Retrieve all departments that have had employees working in them in the past but currently have no employees.

SELECT DISTINCT d.department\_name

FROM departments d

INNER JOIN job\_history jh ON d.department\_id = jh.department\_id

WHERE d.department\_id NOT IN (SELECT department\_id FROM employees);

### ****Simple Subqueries****

1. Retrieve the COUNTRY\_NAME and REGION\_ID of countries where the REGION\_ID is greater than the average REGION\_ID of all countries.

select country\_name,region\_id from countries where region\_id > (select avg(region\_id) from countries);

1. Find the names of employees whose SALARY is greater than the average salary of their department.

select first\_name,salary from employees where department\_id in (select department\_id from employees where salary > (select avg(salary) from employees));

1. Display the DEPARTMENT\_NAME of departments with more than 3 employees.

select department\_name from departments where department\_id in (select department\_id from employees group by department\_id having count(\*) > 3);

1. List the JOB\_TITLE of jobs where the MAX\_SALARY exceeds the maximum salary of all employees.

select job\_title from jobs where max\_salary > (select max(salary) from employees);

1. Fetch the details of employees whose MANAGER\_ID exists in the EMPLOYEE\_ID list of employees working in department 90.

select \* from employees where employee\_id in (select manager\_id from employees where department\_id = 60);

### ****Correlated Subqueries****

1. Retrieve the EMPLOYEE\_ID and SALARY of employees whose salary is higher than the average salary of their department.
2. List the JOB\_ID and MIN\_SALARY of jobs where MIN\_SALARY is greater than the minimum salary of employees in those jobs.
3. Display the COUNTRY\_ID of countries with a REGION\_ID that matches the REGION\_ID of the region named "Europe."
4. Fetch the DEPARTMENT\_NAME of departments whose location is in the same city as the location of department 50.
5. Retrieve the EMPLOYEE\_ID and HIRE\_DATE of employees hired later than their respective department managers.

### ****Inner Joins****

1. Display all employees' FIRST\_NAME, LAST\_NAME, and DEPARTMENT\_NAME.
2. List the EMPLOYEE\_ID, SALARY, and JOB\_TITLE of employees earning more than 10000.
3. Retrieve the REGION\_NAME and COUNTRY\_NAME for all countries.
4. Show the LOCATION\_ID, CITY, and DEPARTMENT\_NAME for departments in location 1700.
5. Fetch the EMPLOYEE\_ID, FIRST\_NAME, and MANAGER\_ID of employees along with their manager’s FIRST\_NAME.

### ****Outer Joins****

1. Display all DEPARTMENT\_NAME and their respective LOCATION\_ID even if no departments exist in some locations.
2. List all COUNTRY\_NAME and their REGION\_NAME, even if there are no regions assigned.
3. Retrieve the FIRST\_NAME, LAST\_NAME, and MANAGER\_ID of employees, including employees without managers.
4. Fetch all JOB\_TITLE and their respective employees, including jobs with no employees assigned.
5. Display all CITY and their DEPARTMENT\_NAME, including cities with no departments.

### ****Cross Joins****

1. Generate a Cartesian product of EMPLOYEES and DEPARTMENTS tables showing FIRST\_NAME, LAST\_NAME, and DEPARTMENT\_NAME.
2. Create a Cartesian product of COUNTRIES and REGIONS displaying COUNTRY\_NAME and REGION\_NAME.

### ****Self Joins****

1. Retrieve the EMPLOYEE\_ID and FIRST\_NAME of employees along with their manager’s EMPLOYEE\_ID and FIRST\_NAME.
2. List employees who work in the same department as employee 100.
3. Show all pairs of employees who are working under the same manager.

### ****Natural Joins****

1. Retrieve the DEPARTMENT\_NAME and LOCATION\_ID using a natural join between DEPARTMENTS and LOCATIONS.
2. List the COUNTRY\_NAME and REGION\_NAME using a natural join between COUNTRIES and REGIONS.

### ****Set Operators with Joins/Subqueries****

1. Find the FIRST\_NAME and LAST\_NAME of employees who either have a salary greater than 15000 or work in department 60.
2. Retrieve all unique CITY names where either departments or employees are located.
3. List all REGION\_NAME and COUNTRY\_NAME where countries belong to regions with REGION\_ID less than 3.

Here are 30 mixed questions covering inline queries, correlated subqueries, subqueries, and joins based on the provided `emp` and `dept` tables in Oracle:

---

### \*\*Inline Queries (Scalar Subqueries)\*\*

1. Display the employee name (`ENAME`) and their department name (`DNAME`) using an inline query.

select ename,(select dname from dept where dept.deptno = emp.deptno) as dname from emp;

1. Show the employee name (`ENAME`) and the location (`LOC`) of their department using an inline query.

select ename,(select loc from dept where dept.deptno = emp.deptno) as loc from emp;

1. Display the employee name (`ENAME`) and the name of their manager (`MGR`) using an inline query.

select ename,deptno,(select count(\*) from emp e2 where e2.deptno = emp.deptno) as tot\_emp\_dep from emp;

5. Display the employee name (`ENAME`) and the average salary of their department using an inline query.

select ename,deptno,(select avg(sal) from emp e2 where e2.deptno = emp.deptno) as avg\_sal from emp;

---

### \*\*Correlated Subqueries\*\*

1. Find all employees who earn more than the average salary of their department.

select ename,sal from emp where sal >all (select avg(sal) from emp group by deptno);

1. Display the names of employees who have the same job as their manager.

select ename,job from emp where job in(select job from emp where empno in (select mgr from emp));

1. Find all employees who were hired before their manager.

select ename,hiredate from emp where hiredate <all (select hiredate from emp where empno in (select mgr from emp));

9. Display the names of employees who have at least one subordinate (i.e., they are a manager).

10. Find all employees who earn more than the average salary of their job type.

select ename,sal from emp where sal >all (select avg(sal) from emp group by job);

---

### \*\*Subqueries\*\*

11. Find all employees who work in the same department as `SMITH`.

12. Display the names of employees who earn more than the average salary of all employees.

13. Find all employees who work in departments located in `NEW YORK`.

14. Display the names of employees who do not have any subordinates (i.e., they are not a manager).

15. Find all employees who earn more than the highest salary in department `30`.

---

### \*\*Joins\*\*

16. Display the employee name (`ENAME`), their job (`JOB`), and their department name (`DNAME`) using a join.

17. Show the employee name (`ENAME`), their manager's name (`MGR`), and their department location (`LOC`) using a join.

18. Display the names of employees and their department names for employees who work in `CHICAGO`.

19. Find all employees who work in the `ACCOUNTING` department using a join.

20. Display the employee name (`ENAME`), their salary (`SAL`), and the average salary of their department using a join.

---

### \*\*Mixed Questions\*\*

21. Display the employee name (`ENAME`), their department name (`DNAME`), and the total number of employees in their department using a join and an inline query.

22. Find all employees who earn more than the average salary of their department and work in `DALLAS`.

23. Display the names of employees who have the same job as `KING` using a subquery.

24. Show the employee name (`ENAME`), their department name (`DNAME`), and the location (`LOC`) of their department for employees who earn more than `2000`.

25. Find all employees who work in departments that have more than 5 employees using a subquery.

---

### \*\*Advanced Questions\*\*

26. Display the employee name (`ENAME`), their salary (`SAL`), and the difference between their salary and the average salary of their department using a join and an inline query.

27. Find all employees who earn more than their manager using a correlated subquery.

28. Display the names of employees who work in departments that do not have any managers.

29. Show the employee name (`ENAME`), their department name (`DNAME`), and the highest salary in their department using a join and an inline query.

30. Find all employees who work in departments where the total salary of all employees in that department is greater than `5000` using a subquery.

---

These questions cover a wide range of SQL concepts and will help you practice inline queries, correlated subqueries, subqueries, and joins in Oracle. Let me know if you need solutions or further explanations for any of these!