1. CREATE A TABLE? TABLE NAME EMPLOYEE, THE TABLE CONTAINS 12 COLUMNS?

EMP_ID	FULL_NAME	ROLE	JOIN_DATE	REPORTS_TO	SALARY	BONUS	DEPT_ID	BLOOD_GROUP	EMP_TYPE	GENDER	DOB

2. ADD PRIMARY KEY TO (EMPNO) COLUMN
3. INSERT SOME DATA INTO TABLE

EMP_ID	FULL_NAME	ROLE	JOIN_DATE	REPORTS_TO	SALARY	BONUS	DEPT_ID	BLOOD_GROUP	EMP_TYPE	GENDER	DOB
1127	Vani	Frontend Developer	10-03-2021	1739	80000	5000	1	A+	Full-time	Female	19-07-1993
1182	Sai Kiran	QA Engineer	12-05-2020	1739	70000		2	B+	Intern	Male	01-12-1992
1261	Ravi	DevOps Engineer	25-11-2021	1846	90000	7000	3	AB-	Full-time	Male	09-08-1995
1053	Mani	Backend Developer	15-06-2022	1053	85000	6000	1	O+	Contract	Male	21-03-1999
1344	Sai	Cybersecurity Analyst	01-02-2023	1846	95000		4	O-	Full-time	Male	05-10-1991
1432	Harika	UI/UX Designer	18-10-2020	1739	78000	4000	1	A-	Intern	Female	30-04-1996
1580	Krishna Reddy	System Administrator	01-09-2019	1846	87000	5500	3	B-	Full-time	Male	19-01-1989
1615	Soumya	Database Engineer	05-12-2021	1739	91000		1	O+	Full-time	Female	27-06-1993
1739	Sirisha	Technical Lead	14-05-2018		130000	12000	1	AB+	Full-time	Female	11-07-1985
1846	Tejaswini	Cloud Architect	20-08-2017		140000	15000	3	A+	Full-time	Female	03-02-1987
1957	Vishnu Priya	HR Executive	01-07-2022	1068	65000		5	B+	Contract	Female	14-11-1995
1068	Sireesha	HR Manager	15-10-2020		95000	6000	5	A-	Full-time	Female	08-06-1990
1133	Raju	Network Security Engineer	12-04-2021	1053	88000	7000	4	O+	Full-time	Male	09-01-1993
1209	Charan	Business Analyst	01-10-2020	1739	78000		1	B+	Intern	Male	23-08-1992
1375	Tarun Kumar	Automation Tester	19-01-2023	1182	73000	4000	2	O-	Full-time	Male	12-12-1994
1482	Lakshmi	Recruiter	03-09-2022	1068	60000	2000	5	A+	Full-time	Female	29-03-1996
1571	Srikanth	DevOps Specialist	25-02-2021	1846	92000	6000	3	B-	Full-time	Male	04-10-1990
1677	Pranavi	Penetration Tester	10-05-2023	1344	96000		4	AB+	Full-time	Female	17-01-1995
1763	Pavan	Software Architect	30-07-2019	1127	145000	20000	1	O+	Contract	Male	25-05-1988
1894	Jahnavi	HR Coordinator	05-11-2020	1068	63000	2500	5	B-	Full-time	Female	22-09-1993

4. CREATE A TABLE? TABLE NAME DEPARTMENT, THE TABLE CONTAINS 4 COLUMNS

DEPT_ID	DEPT_NAME	LOCATION	HEAD_ID

5. INSERT SOME DATA INTO DEPARTMENT, TABLE

DEPT_ID	DEPT_NAME	LOCATION	HEAD_ID
1	Development	Hyderabad	1739
2	Testing	Bangalore	1182
3	DevOps	Chennai	1846
4	Cybersecurity	Pune	1344
5	Human Resources	Delhi	1068

6. Now Add the foreign key Between in Employee Table &Department Table

7. Display the full details of all employees working in the organization
8. Get the list of employee names and their corresponding job roles
9. Display the employee names and their joining dates
10. Get the employee names along with their employee IDs.

11. Display the employee names and their department IDs.
12. Find the names and salary details of all employees.
13. Get the names and bonus amounts assigned to each employee.
14. Display the names and blood groups of all employees.
15. Find the names and genders of all employees in the company.
16. Get the names of employees and their birth dates.
17. Find each employee's name and their annual salary (Salary × 12).
18. Display names and total compensation (Salary + Bonus) of each employee.
19. Show employee names with their half-year salary (Salary × 6).
20. Get names and projected salary after a 10% increment.
21. Display names and reduced salary after a 10% deduction.
22. Show the name and a 25% reduced salary for each employee.
23. Display the name and salary after adding a fixed bonus of ₹2000.
24. Get the name and salary after deducting ₹100 as a penalty.
25. Show names with salary hike of ₹500 per month (Annual: Salary + 500×12).
26. Find names with annual bonus of ₹2400 added to salary.
27. Find how many departments we have in emp table
28. Find the How many types of job roles in the company
29. Find how many unique departments are present in the employee table
30. Find how many unique job roles exist in the company.

Where Clause

1. Display the annual salary of the employee whose name is *Sai*.
2. Show the names of employees working as *Backend Developer*.
3. Find the salary of employees who are working as *QA Engineer*.
4. Get the details of employees who earn more than ₹2000 as bonus.
5. Display all the details of the employee whose name is *Ravi*.
6. Find the details of employees who joined after '2021-01-01'.
7. Show the name and salary along with annual salary of employees earning more than ₹120000 per year.
8. Display the EMP_ID of employees who are working in department 3.
9. Show name and join date of employees who joined before '2020-01-01'.
10. Find the details of employees working as *Technical Lead*.
11. Display name and salary of the employee who is receiving ₹6000 as bonus.
12. Show details of employees whose bonus is greater than their salary.
13. Display EMP_IDs of employees who were born before the year 1987.
14. Show details of employees working as *Cybersecurity Analyst*.
15. Get the details of employees earning more than ₹2000 bonus per month.
16. Show the details of employees whose gender is 'Female'.

17. Display the names and roles of employees whose salary is less than 80000.

18. Find the employees who joined in the year 2020.
19. Show the names of employees whose blood group is 'O+'.
20. Display the details of employees whose bonus is not null.
21. Find the names of employees who are working as 'Intern'.
22. Show the employee IDs and full names of employees whose DEPT_ID is 1.
23. Get details of employees whose salary is between 60000 and 90000.
24. Display the details of employees born after the year 1993.
25. Find employees whose EMP_TYPE is 'Contract'.
26. Get the details of employees who report to manager with EMP_ID 1739.
27. Display the full names and DOB of employees born in 1995.
28. Show the names of employees who have a null value in BONUS.
29. Find the details of employees who joined on or after '2022-01-01'.
30. Get the names of employees whose salary is equal to 78000.
31. Display all employees who are not 'Male'.
32. Find details of employees whose blood group is not 'A+'.
33. Get details of employees with DEPT_ID not equal to 2.
34. Find the employees who have both salary and bonus more than 80000.
35. Show the full names of employees whose name length is more than 5 characters.
36. Display the names of employees who joined exactly on '2021-11-25'.
37. Get details of employees who are reporting to themselves.
38. Find the details of employees working as *HR*
39. Display the details of employees born after the year 2001.
40. Find the details of employees who joined on or after '2002'.

AND & OR

1. Display the names and salary of employees who are 'Full-time' and earn more than 80000.
2. Get the details of employees who joined in 2021 and have a bonus more than 5000.
3. Show the employee names and DOB of all female employees born after 1992 and before 1998.
4. Find the employees working in department 1 and reporting to EMP_ID 1739.
5. Get details of employees who are 'Male' and have blood group 'B-'.
6. List employees whose salary is more than 85000 and bonus is not null.
7. Show the names of employees who are working as 'Intern' and are in DEPT_ID 2.
8. Get the list of employees whose EMP_TYPE is 'Contract' and salary is more than 70000.
9. Display full names and join dates of employees hired after '2020-01-01' and with EMP_TYPE 'Intern'.
10. Find employees whose SALARY is greater than BONUS and EMP_TYPE is 'Contract'.
11. Show the full name, DEPT_ID, and ROLE of employees working as 'Technical Lead' in DEPT_ID 1.
12. Get the full name, DEPT_ID, and SALARY of employees working in DEPT_ID 2 and earning less than 80000.

13. Display full name and SALARY of employees earning more than 60000 and less than 90000.
14. Find the full name and DEPT_ID of employees in DEPT_ID 1 and working as ‘Backend Developer’.
15. Find employees whose role is either ‘Backend Developer’ or ‘Frontend Developer’.
16. Display names of employees who joined in 2020 or 2021.
17. Show the employees who either earn a bonus more than 10000 or salary less than 70000.
18. List employees who report to EMP_ID 1739 or 1846.
19. Get names of employees whose EMP_TYPE is ‘Intern’ or ‘Full-time’.
20. List the names and salary of employees earning 85000 or 95000.

IN & NOT IN

1. Display details of employees who are not working in DEPT_ID 1.
2. Get the list of employees who are not 'Female'.
3. Find all employees who are not ‘Interns’.
4. List names of employees who are not born before 1990.
5. Show details of employees whose blood group is not ‘O+’.
6. Find employees who are not working as 'DevOps Engineer'.
7. Display the details of employees not having bonus more than 5000.
8. Get names of employees who do not report to EMP_ID 1739.
9. Show details of employees who are not hired in 2020.
10. List employees whose salary is not in the range of 60000 to 90000.
11. Display employees working in department 1, 3, or 5.
12. Show names of employees whose role is in ('HR Executive', 'Recruiter', 'HR
13. Manager').
14. Find details of employees whose EMP_ID is in (1053, 1133, 1344).
15. Get names of employees whose DEPT_ID is in (2, 4).
16. Display employees whose BLOOD_GROUP is in ('O+', 'B+').
17. Find employees whose EMP_TYPE is in ('Contract', 'Full-time').
18. Get names of employees not in department 1 or 3.
19. Show employees whose REPORTS_TO is in (1739, 1846, 1068).
20. List names of employees whose salary is in (85000, 91000, 95000).
21. Display employees whose role is not in ('Intern', 'Contract').
22. List employees working in DEPT_ID IN (1, 2, 3).
23. Display employees whose BLOOD_GROUP is IN (‘O+’, ‘B+’, ‘A+’).
24. Get employees whose ROLE is IN ('HR Executive', 'HR Manager').
25. Show employees whose DEPT_ID is NOT IN (1, 3).
26. List employees whose EMP_TYPE is IN ('Full-time', 'Intern').
27. Display employees with BONUS IN (4000, 5000, 6000).
28. Get employees working in DEPT_ID NOT IN (2, 4).
29. Find employees whose REPORTS_TO is IN (1068, 1739).

Between

1. Display employees whose salary is between 80000 and 100000.
2. Find employees whose bonus lies between 2000 and 6000.

3. List names and DOBs of employees born between '1992-01-01' and '1995-12-31'.
4. Get names and join dates of employees hired between '2020-01-01' and '2021-12-31'.
5. Display details of employees with EMP_ID between 1100 and 1500.
6. Show names and annual salary of employees whose annual salary is between 900000
7. and 1100000.
8. Find employees with DEPT_ID between 2 and 4.
9. Get names of employees whose bonus is between 3000 and 10000.
10. List employees whose salary falls between 60000 and 85000.
11. Show details of employees whose join date is between '2018-01-01' and '2023-01-01'.
12. Show employees born between '1990-01-01' and '1993-12-31'.
13. Get employees whose BONUS is between 4000 and 10000.

LIKE

1. Show names of employees whose names start with ‘S’.
2. Find employees whose full name ends with ‘a’.
3. Display details of employees whose full name contains the letter ‘i’.
4. Get names of employees where the second character is 'a'.
5. Show names of employees whose names have exactly 6 characters.
6. List employees whose names start with ‘K’ or ‘P’.
7. Find employees whose role contains the word ‘Engineer’.
8. Get names of employees ending with the letter 'n'.
9. Display names of employees having at least two 'a' characters.
10. List names of employees whose name ends with ‘h’ or ‘y’.
11. Display employees whose name is NOT LIKE '%a%'.
12. List names of employees whose FULL_NAME starts with ‘M’ or ‘J’.
13. Display employees whose FULL_NAME is 5 characters long.

IS NULL & IS NOT NULL

1. Show details of employees whose BONUS is NULL.
2. List names of employees who do not have a BONUS.
3. Display full name and salary of employees whose BONUS is NOT NULL.
4. Get names of employees who have a REPORTS_TO value.
5. Find employees who do not have a reporting manager (REPORTS_TO is NULL).
6. List employees whose BONUS is NULL and EMP_TYPE is ‘Intern’.
7. Show full names of employees whose BONUS is NULL but SALARY is greater than 80000.
8. Find employees whose REPORTS_TO is NOT NULL and work in DEPT_ID 1.
9. Get full names of employees whose BONUS and REPORTS_TO are both NULL.
10. List employees with BONUS IS NOT NULL and are hired after 2021.
11. Display names of employees whose BONUS is NULL or REPORTS_TO is NULL.
12. Find all employees who have a BONUS but no REPORTS_TO.
13. Show details of employees who are missing BONUS but have EMP_TYPE as
14. ‘Contract’.
15. List names and roles of employees with NULL BONUS and working in DEPT_ID 2.
16. Get details of employees who don’t have a BONUS and were born after 1993.
17. Show names of employees whose BONUS is NOT NULL and whose ROLE contains ‘Engineer’.

18. Find employees with a NULL BONUS but a salary above 85000.
19. Display employee details where REPORTS_TO is NULL and ROLE is ‘HR
20. Manager’.
21. List full names of employees whose BONUS is NULL and EMP_ID is greater than 300.
22. Show the employees whose BONUS is NULL and also joined after 2022.

Aggregate Functions COUNT (), SUM (), AVG (), MIN (), MAX ()

1. Find the total number of employees in the company.
2. Count how many employees are ‘Full-time’.
3. Count the number of employees who joined after 2020.
4. Count employees with a bonus greater than 5000.
5. Find how many employees are in DEPT_ID 1.
6. Count the number of employees who are female and have a salary above 80000.
7. Count employees who are interns and have no bonus (BONUS IS NULL).
8. Find the total salary paid to all employees.
9. Find the total bonus given to employees in DEPT_ID 3.
10. Calculate the average salary of employees with role ‘Backend Developer’.
11. Get the average bonus for employees who joined after 2021.
12. Find the maximum salary among male employees.
13. Show the highest bonus received by a female employee.
14. Get the minimum salary among employees with role ‘DevOps Engineer’.
15. Find the lowest bonus among employees who are not interns.
16. Find the total salary of employees who report to EMP_ID 1739.
17. Calculate the average salary of employees in DEPT_ID 2.
18. Get the maximum salary of employees whose EMP_TYPE is ‘Contract’.
19. Find the minimum salary among employees born after 1995.
20. Count the number of employees who are not in DEPT_ID 1 or 2.
21. Find the sum of all bonuses where BONUS is NOT NULL.
22. Find the total number of employees whose names start with 'S'.
23. Show the highest salary of employees with blood group ‘A+’.
24. Count the number of employees whose ROLE contains ‘Engineer’.
25. Calculate the average salary of employees whose salary is greater than 90000.

String Functions: UPPER (), LOWER (), LENGTH (), SUBSTR ()

1. Display all employee names in uppercase.
2. Show full names of employees in lowercase.
3. List employees whose full name length is more than 6 characters.
4. Get the first three letters of each employee’s full name.
5. Find employees whose name starts with 'S' using SUBSTR().
6. List employees whose full name has exactly 7 characters.
7. Show full names and their lengths.
8. Display employee names where the third character is 'a'.
9. Show employees where full name ends with 'a' using SUBSTR() and LENGTH().
10. Get employees whose second character is 'a' using SUBSTR().
11. Show full names where the first letter is lowercase using LOWER(SUBSTR(...)).
12. List names where the last character is 'y'.
13. Display employees with names that have more than 1 occurrence of ‘a’ (with nested

SUBSTR() logic).
14. Show employee roles in uppercase.
15. List names of employees whose role has more than 10 characters.
16. Show only the first 2 characters of the ROLE.
17. Display full name and the length difference between full name and role.
18. Get employees where the first character of role is 'C'.
19. Show names and roles, with both displayed in uppercase.
20. Find names with second letter not 'a' using SUBSTR() and condition.

Date Functions **YEAR (), MONTH (), DATEDIFF ()**

1. List employees born in the year 1993.
2. Show employees who joined in 2021.
3. Display names and month of their date of birth.
4. Show full name and how many years old they are (using YEAR(CURDATE()) - YEAR(DOB)).
5. Show e mployees hired in the month of March.
6. Find employees born in June.
7. Get names of employees hired in the same month they were born.
8. Show the number of days since employee joined.
9. Find employees who joined less than 500 days ago.
10. List employees who were born in January.
11. Show employees hired in 2020 and in March.
12. Display full names and the difference in days between DOB and JOIN_DATE.
13. Show employees hired in the same year as EMP_ID > 1500.
14. List employees whose joining year is even (MOD(YEAR(JOIN_DATE), 2) = 0).
15. Show employees who were hired before they turned 30 (DOB + 30 years < JOIN_DATE).
16. Find employees hired in the last 2 years.
17. Display employees born in the 90s (YEAR(DOB) BETWEEN 1990 AND 1999).
18. List employees with same DOB month and JOIN month.
19. Show employees with JOIN_DATE and DOB within the same year.
20. Find employees who were born at least 10000 days ago.

MAX (), MIN ()

1. List the name of the employee with the highest salary.
2. Find the name of the employee drawing the lowest salary.
3. Show the name and joining date of the employee who joined earliest.
4. Display the name and join date of the employee who joined most recently.
5. Fetch the name and bonus of the employee receiving the smallest bonus.
6. Get the name, salary, and bonus of the employee who receives the highest bonus.
7. Retrieve all details of the employee with the highest EMP_ID.

8. Find the employee record with the earliest JOIN_DATE.
9. Show details of the employee with the lowest total annual salary (SALARY * 12).
10. Get the name and annual salary of the employee whose annual salary is higher than all 'Interns'.
11. List the full name of the employee who is the oldest in the company (based on DOB).
12. Find the full name of the employee who is the youngest.
13. Display the employee with the longest full name (maximum LENGTH).
14. List the employee whose bonus is closest to 0 but not NULL.
15. Find the employee with the lowest salary among those working as ‘Backend Developer’.
16. Show the full name of the employee with the latest date of birth.
17. Get the details of the employee whose REPORTS_TO value is highest.
18. Find the employee with the highest annual compensation (SALARY + BONUS).
19. Display the name of the employee who earns more than everyone else in DEPT_ID 1.
20. List the name and salary of the employee whose salary is greater than the average of all 'Contract' employees.

Window Functions

1. Assign a row number to each employee ordered by salary in descending order.
2. Display the top 3 highest-paid employees in each department using RANK().
3. Find the ROW_NUMBER() of each employee ordered by JOIN_DATE.
4. Rank employees by bonus amount within each ROLE.
5. Show employees with their DENSE_RANK() based on SALARY in descending order.
6. List employees and divide them into 4 buckets using NTILE(4) based on their salary.
7. Within each department, assign a ROW_NUMBER() based on the order of JOIN_DATE.
8. Rank employees based on BONUS — ties should receive the same rank (use RANK ()).
9. Display ROW_NUMBER() for employees partitioned by EMP_TYPE and ordered by SALARY.
10. For each gender, show employees and assign ranks based on their salary.
11. Use DENSE_RANK() to find if any two employees in the same department have the same salary.
12. Assign a ROW_NUMBER() to each employee grouped by department and sorted by bonus.
13. Display employees and their position in salary ranking (within their ROLE) using RANK().
14. Divide employees into 3 salary bands using NTILE(3) over SALARY.
15. Assign ROW_NUMBER() to each employee, then filter to get the most recently joined employee in each department.

ORDER BY

1. List all employees sorted by salary in ascending order.
2. Display full names and join dates of employees, ordered by the most recent hire first.
3. Show employee names and bonuses, ordered from highest to lowest bonus.
4. List employees by department, sorted by DEPT_ID in ascending order.
5. Display full names and DOBs, ordered by age (youngest first).
6. Show full names and roles of employees, ordered alphabetically by role.

7. List employees with their salaries, sorted in descending order of salary.
8. Show full name and bonus, ordered by NULL bonuses last and then descending bonuses.
9. List employees ordered by blood group alphabetically.
10. Show full name and salary, sorted by EMP_TYPE ascending and then salary descending.
11. Display names and DOB, ordered by year of birth (oldest first).
12. List employees sorted first by DEPT_ID, then by salary in descending order.
13. Show employee names, ordered by LENGTH of their names in descending order.
14. List full name and join date of employees, sorted by join year then by name.
15. Display employees ordered by GENDER and then by SALARY.
16. Show employees who joined after 2020, ordered by bonus in descending order.
17. List employees by EMP_TYPE in descending order and full name alphabetically.
18. Get employee names sorted by the last character of their name.
19. Display employees ordered by ROLE length (number of characters in role) and then by full name.
20. List employees by annual salary (salary * 12), sorted from highest to lowest.

GROUP BY

1. Show the number of employees in each department, excluding those with the role
2. ‘Penetration Tester’.
3. Display the maximum salary paid for each unique role.
4. Count the noof employees in each role where the full name contains the letter 'a'.
5. Find how many employees in each dept have received a bonus (BONUS IS NOT NULL).
6. List the total number of employees in each department, excluding those with the role '
7. Software Architect'.
8. Show total salary paid to all employees for each job role.
9. Count the number of employees working as ‘Technical Lead’ in each department.
10. Display the average salary paid per department, excluding DEPT_ID 2.
11. Count how many employees have the letter 'a' in their full name grouped by role.
12. For each department, show the number of employees and their average salary were
13. salary is greater than 80000.
14. List total salary paid and number of employees working as 'Intern' in each department.
15. For each role, show the number of employees and the highest salary among them.
16. Display the maximum salary given to employees grouped by department.
17. Show how many times each unique salary value occurs in the employee table.
18. Display the average bonus given to employees in each EMP_TYPE.
19. Show the total number of male and female employees in each department.
20. Count how many employees joined in each year.
21. List the number of employees grouped by their blood group.
22. Display the total bonus amount given in each department.
23. Show the average salary of employees grouped by gender.
24. List the maximum bonus received by employees in each job role.
25. Find how many employees report to each manager (REPORTS_TO).
26. Show the count of employees in each department whose salary is above 85000.
27. Display the number of employees and total salary grouped by EMP_TYPE, excluding
28. interns.

Having Clause

1. Display the roles (designations) where at least 2 employees are working.
2. List the salary values that appear more than once in the employee table.
3. Show department IDs that have at least 2 employees with the letter 'A' or 'S' in them full name.
4. Display each role and its total salary if the total salary is more than 3,45,000.
5. List job titles and their total salary where employees earn more than 80,000 individually.
6. Show job roles where the maximum salary exceeds 90,000.
7. Display the number of employees earning above 85,000 in each job and show only those jobs where the total salary exceeds 3,80,000.
8. List DEPT_IDs and employee counts where there are at least 2 employees with the role 'Recruiter'.
9. Show department IDs where total salary exceeds 4,00,000 and at least 4 employees are present.
10. Show department-wise employee count where at least 2 employees are working as ‘DevOps Specialist’.
11. List job titles and their max salary only if max salary exceeds 95,000.
12. Display salaries that occur more than once among the employees.
13. List the join dates that appear more than once in the employee table.
14. Display department-wise average salary where the average is less than 90,000.
15. List departments that have at least 3 employees whose names include ‘A’ or ‘S’.
16. Show each job role's minimum and maximum salaries where MIN salary > 60,000 and MAX salary < 1,00,000.
17. List EMP_TYPEs where the number of employees is more than 2 and the total bonus exceeds 10,000.
18. List EMP_TYPEs having an average salary greater than 85,000.
19. Show DEPT_IDs where the number of employees is more than 3 and the average bonus is above 4,000.
20. Display GENDER-wise total salary where the count of employees is at least 2.
21. List REPORTS_TO IDs where more than one employee reports to the same manager.
22. Show BLOOD_GROUPs with more than 2 employees.
23. Display DEPT_IDs where the maximum bonus exceeds 6,000.
24. List ROLEs where at least 2 employees have joined after 2020.
25. Show EMP_TYPEs having more than 2 employees and the minimum salary is above 60,000.
26. Display DEPT_IDs where the average of (salary + bonus) is greater than 95,000.
27. List ROLEs where the number of female employees is at least 2.
28. Show DEPT_IDs where more than 2 employees are born after 1993.
29. Display ROLEs where total bonus received is more than 10,000.
30. List JOB ROLES where the average salary is less than 80,000 and more than 1 employee exists.
31. Show DEPT_IDs where the minimum bonus is not NULL and is greater than 2,000.
32. Display BLOOD_GROUPs having employees with total salary exceeding 2,50,000.

Special Question

1. List the full names and departments of employees whose salary is greater than the average salary of their department. Order the result by salary descending.

2. Display the DEPT_ID and total salary of employees, only if the total salary per department exceeds 250000 and there are more than 3 employees in that department.
3. Get the full names, roles, and join dates of employees who joined between '2020-01-01' and '2022-12-31', and have bonuses greater than 4000. Sort by join date.
4. Show EMP_TYPE and average bonus for each type, only if at least 2 employees in that type have bonus not null.
5. List ROLEs and count of employees where the role contains the word ‘Engineer’ and at least 2 employees have salaries greater than 85000.
6. Display names and DOBs of employees whose names start with 'S' or 'P', are born between 1990 and 1996, and work in DEPT_ID 1 or 3.
7. Get the department and count of employees where at least 2 have the letter 'a' in their name and report to EMP_ID 1739.
8. List the roles where the average salary is greater than 90000 and total bonus is not null.
9. Show BLOOD_GROUP and number of employees per group where at least 2 people are ‘Male’ and their salary is more than 75000.
10. Display EMP_IDs and names of employees whose salary is greater than their bonus (bonus not null), and who joined after 2020.
11. Show departments where more than one ‘Full-time’ employee exists and the maximum salary exceeds 95000.
12. Get the name and salary of employees whose name contains exactly 2 occurrences of the letter 'a', and who are not 'Interns'.
13. List ROLEs and count of employees where the max bonus is greater than 5000 and average salary is less than 100000.
14. Display details of employees whose salary is between 80000 and 95000 and whose bonus is either 4000 or 5000. Sort by ROLE alphabetically.
15. List the DEPT_IDs where there are at least 3 employees who joined before 2021 and have a non-null bonus.
16. Show the ROLEs and number of employees where the ROLE ends with ‘Developer’ and count > 1.
17. Display full names and annual salary of employees who earn more than the average of all 'Interns'.
18. List the BLOOD_GROUPs where at least one employee has bonus null and another has bonus more than 4000.
19. Find employees whose names are exactly 6 characters long, have EMP_TYPE
20. 'Contract', and joined after 2020.
21. Show ROLE and max salary of employees grouped by role, where the max salary is greater than the max bonus for that role.

Subqueries

1. Show the names of employees whose salary is less than the salary of 'Sirisha'.
2. Get the details of employees who joined after the employee named 'Soumya'.
3. Find employees whose bonus is more than the bonus of the employee earning the least salary.
4. Display employees whose salary is equal to the maximum salary of all employees.
5. List names and roles of employees who joined before the earliest join date among all 'Full-time' employees.
6. Display the names of employees who earn more than the average salary of their department.
7. Get names of employees who earn more than at least one other employee in the same

role.

8. Show names and join dates of employees who are older than someone else in the same department.
9. List names of employees whose bonus is greater than that of any other employee in the same EMP_TYPE.
10. Display names of employees whose salary is greater than the salary of their manager.
11. Get the full names of employees who work in the same department as ‘Ravi’.
12. Show details of employees who are not in the same role as ‘Mani’.
13. Display the names of employees whose role is shared by at least two other employees.
14. List names of employees working in departments where someone earns more than 140000.
15. Show names of employees whose salary matches someone else's bonus.
16. List departments that have at least one female employee.
17. Get details of employees if there exists someone in the same department with a NULL bonus.
18. Show names of employees who do not report to anyone who earns more than 120000.
19. Display the roles for which at least one intern exists.
20. List names of employees for whom no one else in the same role has a higher salary.
21. Scalar Subqueries
22. Display the names and salary of employees who earn less than the average salary of all ‘Full-time’ employees.
23. Find employees whose bonus is equal to the minimum bonus among all 'Interns'.
24. List employees who are younger than the youngest ‘Backend Developer’.
25. Show names of employees whose salary is higher than the salary of the employee with EMP_ID 1053.
26. Find employees whose joining date is after the joining date of the highest-paid employee.
27. List employees whose salary is greater than the average salary of employees with the same EMP_TYPE.
28. Find employees who are the highest earners in their department.
29. Show names of employees whose DOB is earlier than someone else in the same department.
30. Display employees who are not the oldest in their job role.
31. List employees who earn less than someone they report to.
32. List names of employees working in the same departments as both 'Tejaswini' and 'Pavan'.
33. Show employees who do not work in the same department as any 'UI/UX Designer'.
34. Find names of employees who are in the same EMP_TYPE as ‘Lakshmi’.
35. Display names of employees not in the EMP_TYPE of any intern.
36. List employees who are not reporting to anyone in DEPT_ID 1.
37. List employees who have at least one peer in the same role and department.
38. Show employees for whom no one else in the same department joined earlier.
39. List employees who are the only one with their role in their department.
40. Display details of employees if there's at least one person in the same DEPT_ID with a NULL bonus.
41. Get names of employees who do not share their BLOOD_GROUP with any other employee.

SET OPERATION (UNION, INTERSECT, EXCEPT)

1. List the full names of all employees who are either 'Full-time' or 'Contract'.
2. Find the names of employees who are both 'Interns' and working in DEPT_ID 2.
3. Get the names of employees who are in DEPT_ID 1 or DEPT_ID 3 but exclude those
4. who are in DEPT_ID 2.
5. Display the roles of employees who are either earning a bonus greater than 5000 or
6. having a salary less than 70000.
7. Show the names of employees who are both Female and working in department 1.
8. List EMP_IDs of all employees who are either reporting to 1739 or have the role 'HR
9. Executive'.
10. Get full names of employees who joined after 2021 or were born before 1993.
11. Find names of employees who are in the EMP_TYPE 'Intern' but are not from
12. DEPT_ID 2.
13. Show the ROLEs that are assigned to both Male and Female employees.
14. Display the EMP_IDs of employees who earn a bonus and also have a salary greater than 90000.

Employee–Manager (REPORTS_TO) Relationship

1. What is the full name of the reporting manager of the employee named 'Sai Kiran'?
2. What is the full name of the manager's manager for the employee 'Charan'?
3. What is the department ID of the manager of 'Krishna Reddy'?
4. What is the salary of the manager of 'Lakshmi'?
5. What is the blood group of the manager's manager of 'Raju'?
6. List the names of all employees who report to 'Sirisha'.
7. How many employees are reporting directly to 'Tejaswini'?
8. Show the details of employees who report to 'Soumya'.
9. List the names of employees who report to the same manager as 'Soumya'.
10. Find the number of employees who report to the same manager as 'Tarun Kumar'.

CASE statements

1. Show the FULL_NAME and a column labeled "Salary_Level" which displays 'High' if salary > 90000, 'Medium' if between 70000 and 90000, else 'Low'.
2. Display employee FULL_NAME, ROLE, and a message "Eligible for Bonus" or "Not Eligible" based on whether BONUS is greater than 0.
3. Show all employees' FULL_NAME and a tag 'Senior' if JOIN_DATE is before 2015, else 'Junior'.
4. List employees' FULL_NAME and BLOOD_GROUP, along with a status: 'Critical' if BLOOD_GROUP = 'AB-', 'Normal' otherwise.
5. For each employee, show FULL_NAME and a column "Contract_Status" which displays 'Permanent' for 'Full-time', 'Temporary' for 'Contract' and 'Internship' for 'Intern'.

LIMIT

1. Show the top 5 employees with the highest SALARY.
2. Display the first 10 employees who joined the company (ORDER BY JOIN_DATE).

3. Retrieve the details of the 3 most recently hired employees.
4. Get the top 5 employees with the highest BONUS.
5. Show the first 7 employee records in alphabetical order of FULL_NAME.
6. Which column(s) should be indexed to improve performance for queries filtering by
7. EMP_ID?

INDEX.

1. Create a query that retrieves employees with BONUS > 5000 and explain why creating an index on BONUS could help.
2. Retrieve employees who report to EMP_ID = 1739. Which column should be indexed for faster lookup?
3. Write a query filtering by DEPT_ID and ROLE, and explain which composite index would optimize this.
4. Display employee FULL_NAMEs born after 1995. Would indexing DOB improve performance?

JOINS

INNER JOIN

1. Show the full name and department name for all employees.
2. Display the full name and location of employees working as ‘Manager’.
3. Get the full name, salary, and department name of employees working as ‘Clerk’ in DEPT_ID 2 with salary above 80000.
4. Display employee full name, department name, and location for employees earning more than 90000 in ‘Hyderabad’.
5. Show employee full name and department location of all employees.
6. Show department name and salary of all employees working in the ‘Development’ department.
7. Display department name and annual salary of employees whose salary is more than 95000.
8. List full names and department names of employees if their department name contains the letter 'a'.
9. Show full names and department names of all employees working as ‘Intern’.
10. Get department name and role of all employees where both department and role start with ‘D’

SELF JOIN

1. Display employee full name and their manager's name if the employee works as ‘Clerk’.
2. Get employee full name and manager's role if the manager is in DEPT_ID 1 or 2.
3. Show employee name and manager’s salary if both earn more than 80000.
4. List employee full name and manager’s hire date if the employee joined before 2019.
5. Show employee name and manager’s bonus if the employee is a ‘Sales Executive’ and manager is in DEPT_ID 3.
6. List employee and manager names along with salaries where employee earns more than their manager.
7. Display employee and manager names with their hire dates if manager joined before

employee.
8. List employees and their managers if both work in the same role.
9. Display employee and manager names if manager is working as actual ‘Technical Lead’.
10. Show employee name, manager name, and their annual salaries if employee is in DEPT_ID 1 or 2 and earns less than their manager.
11. Display employee name and manager's designation for all employees.
12. Show employee name and manager's salary if the manager’s salary ends with 50.

LEFT JOIN

1. Display all employee names along with their department names, including employees who are not assigned to any department.
2. Show the employee name, role, and department location even if the department info is missing.
3. List full name and department name of all employees, including those who belong to non-existent DEPT_IDs.
4. Display all employee details along with department head ID (HEAD_ID), even for employees without department mapping.
5. Show all employees and their department names, but only for those with a salary greater than 90000.
6. List employees along with department locations even if the department is NULL.
7. Display employee names and their department names where employees are hired after 2020, including unmatched departments.

RIGHT JOIN

1. Show all departments and the names of employees working in them, including departments that have no employees.
2. List department names and roles of employees, even if no one is assigned to some departments.
3. Display all department names with the total number of employees in each (use GROUP BY), showing 0 if none.
4. Get department names and employee names, including departments where EMP_TYPE is NULL for all.
5. Display department names and salary of employees working there, including departments without any salary data.
6. Show all department names and location with any employee who joined in 2023 or show NULL if no such employees exist.
7. List department names and reporting managers (REPORTS_TO) from the employee table, including departments with no such mappings.

CROSS JOIN

1. Display all combinations of employees and departments (Cartesian product).
2. List all possible employee–department pairs along with employee ROLE and

department LOCATION.
3. Show combinations of employee FULL_NAME and DEPT_NAME to evaluate team assignment options.
4. For every department, list all employee JOIN_DATEs to simulate assigning employees to new branches.
5. Generate a list of all potential assignments of employees to departments with their SALARY and HEAD_ID.
6. Create a report pairing each employee with every department, showing whether the employee's REPORTS_TO matches the department's HEAD_ID.
7. Produce a list of all FULL_NAME and DEPT_NAME combinations where employee's BLOOD_GROUP is 'O+' (filter after the join).

Views

1. Create a view that displays EMP_ID, FULL_NAME, ROLE, and SALARY for all employees with SALARY greater than 80000.
2. Create a view showing employee names, department names, and their JOIN_DATE.
3. Create a view that contains only the employees who are Interns and working in DEPT_ID 2.
4. Create a view listing EMP_ID, FULL_NAME, and ANNUAL_SALARY (SALARY × 12) for all employees.
5. Create a view showing employee details (name, department, role) for those whose manager is EMP_ID 1739.
6. Create a view of employees hired after 2020 and earning a bonus more than 5000.
7. Create a view that joins the employee and department tables and displays EMP_ID, FULL_NAME, DEPT_NAME, and LOCATION.
8. Create a view that displays only the Full-time employees who are earning more than the average salary of all employees.
9. Create a view that includes EMP_ID, FULL_NAME, DOB, and calculates AGE of employees using current date.
10. Create or replace a view to show all employees who report to the same manager as EMP_ID 1846.

LEAD ( ) and LAG ( )

1. Display each employee's name along with the next employee's name (based on salary order).
2. Show each employee’s name and the name of the employee who joined just before them (based on JOIN_DATE using LAG()).
3. Display each employee’s salary and the difference between their salary and the next employee’s salary (ordered by salary).
4. Show employee names along with the previous and next employee’s role using LAG() and LEAD().
5. List each employee's name, join date, and the join date of the next person who joined.
6. Display employees along with the difference between their bonus and the previous employee's bonus (use LAG() on BONUS).
7. Show each employee along with the DEPT_ID of the previous and next employee in the same department, ordered by SALARY.
8. Display the employee name and their salary, along with the name of the employee who earns slightly more than them (use LEAD() on SALARY).

9. List employees and show the change in salary compared to the previous employee in terms of SALARY INCREASE (use LAG()).
10. Display each employee's name and how many days after the previous employee they joined (difference between JOIN_DATE using LAG() and DATEDIFF()).

Cursors

1. Write a cursor to fetch and display all employee full names one by one.
2. Create a cursor to loop through employees and calculate their annual salary (SALARY × 12).
3. Use a cursor to display all employees in the 'Development' department along with their role.
4. Write a cursor that prints names of employees earning more than 90,000.
5. Create a cursor to list employees and mark them as "Senior" if their joining year is before 2018.
6. Use a cursor to go through each department and count the number of employees.
7. Write a cursor to display employees along with their bonus. If bonus is NULL, show "No Bonus".
8. Create a cursor to fetch employees who are 'Interns' and insert their names into a separate log table.
9. Write a cursor that reads employee data and updates a column EMP_TYPE_DESC as 'Permanent' for 'Full-time', 'Temporary' for 'Contract', etc.
10. Develop a cursor to calculate and print the average salary per department by looping through DEPT_IDs.

PL/SQL

Decision-Making (IF / CASE)

1. Create a procedure that checks if an employee's salary is greater than 90000. If yes,
2. display "High Earner", else display "Average Earner".
3. Write a procedure that accepts an EMP_ID and checks if the employee is 'Full-time'.
4. If yes, print their name and salary.
5. Create a procedure to accept a DEPT_ID and display "Remote Team" if the
6. LOCATION is not 'Hyderabad'. Otherwise, print "In-House Team".
7. Write a procedure that checks if an employee has a BONUS greater than 5000. If yes,
8. show full details; else, show only EMP_ID and FULL_NAME.
9. Create a stored procedure that accepts a GENDER and prints "Welcome Sir" for
10. Male, "Welcome Madam" for Female, and "Welcome" otherwise.
11. Develop a procedure to take EMP_ID and print "Reports to CEO" if REPORTS_TO
12. is NULL. Otherwise, show the manager's EMP_ID.
13. Write a stored procedure that checks if an employee's BLOOD_GROUP is 'O+' or
14. 'A+'. If yes, show their name and department. Else print 'N/A'.
15. Create a procedure to accept EMP_TYPE. If it is 'Intern', print "Temporary Access
16. Granted"; else, print "Full Access".
17. Write a procedure that checks if a given EMPLOYEE is working in DEPT_ID 2. If
18. yes, print their role and location; otherwise, say "Not in Testing Department".

19. Create a stored procedure that accepts SALARY. If it's above 100000, print "Eligible
20. for Leadership Program". If it's between 70000 and 100000, print "Eligible for
21. Upskilling". Else, print "Regular Track".

Loops

1. Loop through all employee records and print EMP_ID and FULL_NAME one by one using a WHILE loop.
2. Write a procedure to calculate and display the total salary (SALARY + BONUS) of each employee using a LOOP until all records are covered.
3. Create a loop that prints the EMP_IDs of employees hired in or after the year 2020.
4. Write a procedure that counts how many 'Full-time' and 'Intern' employees are there using a loop.
5. Loop through all employees and print the FULL_NAME of those whose salary is greater than 90000, and stop the loop if an employee is found with salary less than 50000 (use LEAVE).

6. Print the first 10 prime numbers using a `WHILE` loop.
7. Check if a number is a strong number (e.g., 145 = 1! + 4! + 5!) using a `REPEAT` loop.
8. Find the factorial of a number using a `LOOP`.
9. Print Fibonacci series up to N terms using a loop.
10. Reverse a number using a loop and print the result.
11. Check whether a given number is a palindrome using a loop.
12. Print all Armstrong numbers between 1 and 500 using a loop.
13. Find the sum of digits of a number using a loop.
14. Count the number of even and odd digits in a number using a loop.
15. Check if a number is a perfect number (sum of its divisors equals the number) using a loop.

Stored Functions

1. Create a function that takes EMP_ID as input and returns the employee's full name.
2. Write a function that returns the annual salary (SALARY × 12 + BONUS) of an employee given their EMP_ID.
3. Create a function that takes a DEPT_ID and returns the number of employees in that department.
4. Create a function that returns TRUE if an employee is eligible for bonus
5. (BONUS > 0), otherwise FALSE.
6. Write a function to return the age of the employee (based on DOB) given the EMP_ID.
7. Create a function to check whether the FULL_NAME contains more than one word.
8. Write a function that takes GENDER as input and returns the count of employees of that gender.
9. Create a function that returns the average salary of employees in a specific department.
10. Write a function that takes EMP_TYPE and returns the count of employees with that type.
11. Create a function to return the length of the FULL_NAME for a given employee.

Stored Procedures

1. Create a procedure that displays the full details of an employee given their EMP_ID.
2. Write a procedure that inserts a new employee into the table with parameters like
3. name, salary, role, etc.
4. Create a procedure that increases the salary of all employees in a department by 10%.
5. Write a procedure that deletes all employees of type 'Intern'.
6. Create a procedure that prints all employees who joined after a given date.
7. Write a procedure that displays the highest and lowest salary in each department.
8. Create a procedure that prints the number of employees with salary greater than a
9. given threshold.
10. Write a procedure to update BONUS to 0 for all employees whose salary is below
11. 60000.
12. Create a procedure that accepts a department name and returns all employee names in
13. that department.
14. Write a procedure that prints all employees who report to a given EMP_ID.

Exception Handling

1. Write a block to fetch the salary of an employee based on EMP_ID.
   If the employee does not exist, handle the exception and print: "Employee not found."
2. Create a PL/SQL block to divide an employee's salary by BONUS.
   Handle division by zero if BONUS is NULL or 0.
3. Write a block that attempts to insert a new employee with a duplicate EMP_ID.
   Catch the duplicate key exception and display a custom message.
4. Write a block that tries to update the BONUS for a given EMP_ID.
   If EMP_ID doesn't exist, handle the "no data found" exception and notify the user.
5. Create a block to delete employees with salary below 10000.
   Handle any data manipulation exceptions, and print the number of rows deleted or an error message if none were deleted.

Views

1. Create a view to display only the EMP_ID, FULL_NAME, SALARY, and DEPT_ID
2. of employees earning more than 80,000.
3. Create a view named high_bonus_employees that shows employees whose BONUS is
4. greater than their SALARY * 0.1.
5. Create a view to display employee full name, role, and department name using a join
6. with the department table.
7. Create a view showing only the employees who are 'Full-time' and belong to
8. DEPT_ID = 1.
9. Create a view named female_employees_details to display full name, DOB, and blood
10. group of all female employees.
11. Create a view to show the EMP_ID, FULL_NAME, SALARY, and
12. ANNUAL_SALARY (calculated as SALARY * 12) of all employees.
13. Create or replace a view to display employee details with a calculated column
14. showing "Seniority" as 'Senior' if salary > 90000, else 'Junior'.

15. Create a view named dept_summary to show DEPT_ID and number of employees in
16. each department.
17. Create a view that includes employees who report to the same manager (i.e., group by
18. REPORTS_TO) with the count of direct reports.
19. Create a view for all employees working in 'Hyderabad' location with their
20. department names.

Triggers

1. Create a trigger to automatically set the BONUS to 0 if it is inserted as NULL in the employee table.
2. Create a trigger to log any deletion of an employee by inserting their EMP_ID, FULL_NAME, and current timestamp into a separate employee_audit table.
3. Create a trigger that prevents inserting employees with a SALARY less than 20000.
4. Create a trigger to update the EMP_TYPE to 'Contract' if an employee is inserted with a BONUS less than 2000.
5. Create a trigger to automatically update a field called LAST_UPDATED (add this field in your employee table) whenever an employee's salary is updated.