```sql
CREATE DATABASE SQL_ASSIGNMENT2;
show databases;
use SQL_ASSIGNMENT2;
CREATE TABLE tbl_emp_detail_data (
    empid INT,
    firstname VARCHAR(50),
    lastname VARCHAR(50),
    salary FLOAT,
    joiningdate DATE,
    department VARCHAR(50),
    gender VARCHAR(10),
    jobtitle VARCHAR(50)
);
CREATE TABLE tbl_project_deatils (
    project_id INT,
    emp_id_no INT,
    project_name VARCHAR(100),
    start_date DATE,
    end_date DATE,
    status VARCHAR(20)
);

CREATE TABLE tbl_department_heads (
    deptheadid INT,
    deptname VARCHAR(50),
    deptheadname VARCHAR(50)
);

CREATE TABLE tbl_bonus (
    bonusid INT,
    empid INT,
    bonusamount FLOAT,
    bonusyear INT
);
show create table tbl_department_heads;
select * from tbl_department_heads;

#1. Total Salary Paid per Department Ranked Descending
SELECT
    department,
    SUM(salary) AS total_salary,
    RANK() OVER (ORDER BY SUM(salary) DESC) AS salary_rank
FROM
    tbl_emp_detail_data
GROUP BY
    department
ORDER BY
    total_salary DESC;

#2.List Employees Who Have Worked on More Than One Project
SELECT
    emp_id_no,
    COUNT(DISTINCT project_id) AS project_count
FROM
    tbl_project_deatils
GROUP BY
    emp_id_no
```

```sql
HAVING
    COUNT(DISTINCT project_id) > 1;

#3. Employees Working on Ongoing Projects Ordered by Highest Salary
SELECT
    e.empid,
    e.firstname,
    e.lastname,
    e.salary,
    p.project_name,
    p.status
FROM
    tbl_emp_detail_data e
JOIN
    tbl_project_deatils p ON e.empid = 'p.empidno'
WHERE
    p.status = 'Ongoing'
ORDER BY
    e.salary DESC;

#4. Most Experienced Employee in Each Department

SELECT
    department,
    empid,
    firstname,
    lastname,
    joiningdate,
    RANK() OVER (PARTITION BY department ORDER BY joiningdate ASC) AS experience_rank
FROM
    tbl_emp_detail_data
QUALIFY experience_rank>1;
 #5. Employees With Salary Above Department Average
WITH dept_avg AS (
    SELECT department, AVG(salary) AS avg_salary
    FROM tbl_emp_detail_data
    GROUP BY department
)
SELECT
    e.empid,
    e.firstname,
    e.lastname,
    e.salary,
    e.department
FROM
    tbl_emp_detail_data e
JOIN
    dept_avg d ON e.department = d.department
WHERE
    e.salary > d.avg_salary;
#6. Rank Departments Based on Total Salary + Bonus, and Rank Employees Within Departments by Total
Compensation
WITH dept_salary_bonus AS (
    SELECT
        e.department,
        SUM(e.salary) AS total_salary,
        COALESCE(SUM(b.bonusamount), 0) AS total_bonus,
        SUM(e.salary) + COALESCE(SUM(b.bonusamount), 0) AS total_compensation
    FROM tbl_emp_detail_data e
    LEFT JOIN tbl_bonus b ON e.empid = b.empid
    GROUP BY e.department
),
ranked_departments AS (
    SELECT
        department,
        total_salary,
        total_bonus,
        total_compensation,
        RANK() OVER (ORDER BY total_salary + total_bonus DESC) AS dept_rank
    FROM dept_salary_bonus
),
ranked_employees AS (
    SELECT
        e.empid,
        e.firstname,
        e.lastname,
        e.department,
        e.salary,
        COALESCE(b.bonusamount, 0) AS bonus,
        e.salary + COALESCE(b.bonusamount, 0) AS total_comp,
        RANK() OVER (PARTITION BY e.department ORDER BY e.salary + COALESCE(b.bonusamount, 0) DESC) AS
employee_rank
    FROM tbl_emp_detail_data e
    LEFT JOIN tbl_bonus b ON e.empid = b.empid
)
SELECT
    rd.dept_rank,
    rd.department,
    re.empid,
    re.firstname,
    re.lastname,
    re.salary,
    re.bonus,
    re.total_comp,
    re.employee_rank
FROM ranked_departments rd
JOIN ranked_employees re ON rd.department = re.department
ORDER BY rd.dept_rank, re.employee_rank;
```

#7. Rank Employees by Number of Projects and Average Project Duration, and Rank Departments by Average Project Duration

```
WITH project_stats AS (
    SELECT
        p.emp_id_no,
        COUNT(DISTINCT p.project_id) AS project_count,
        AVG(DATEDIF(day, p.start_date, COALESCE(p.end_date, CURRENT_DATE))) AS avg_project_duration
    FROM tbl_project_deatils p
    GROUP BY p.emp_id_no
),
employee_ranks AS (
    SELECT
        e.empid,
        e.firstname,
        e.lastname,
        ps.project_count,
        ps.avg_project_duration,
        RANK() OVER (ORDER BY ps.project_count DESC, ps.avg_project_duration DESC) AS employee_rank
    FROM tbl_emp_detail_data e
    JOIN project_stats ps ON e.empid = ps.emp_id_no
),
department_avg_duration AS (
    SELECT
        e.department,
        AVG(ps.avg_project_duration) AS dept_avg_duration,
        RANK() OVER (ORDER BY AVG(ps.avg_project_duration) DESC) AS department_rank
    FROM tbl_emp_detail_data e
    JOIN project_stats ps ON e.empid = ps.emp_id_no
    GROUP BY e.department
)
SELECT
    er.employee_rank,
    er.empid,
    er.firstname,
    er.lastname,
    er.project_count,
    er.avg_project_duration,
    da.department_rank,
    e.department
FROM employee_ranks er
JOIN tbl_emp_detail_data e ON er.empid = e.empid
JOIN department_avg_duration da ON e.department = da.department
ORDER BY da.department_rank, er.employee_rank;
```

#8. Rank Project Managers Based on Number of Employees Under Them and Employees Within Projects Based on Salary

```
WITH employee_project_counts AS (
    SELECT
        p.project_id,
        p.emp_id_no,
        e.department
    FROM
        tbl_project_deatils p
    JOIN
        tbl_emp_detail_data e ON p.emp_id_no = e.empid
),
manager_employee_counts AS (
    SELECT
        d.deptheadname AS project_manager,
        e.department,
        COUNT(DISTINCT ep.emp_id_no) AS employee_count
    FROM
        employee_project_counts ep
    JOIN
        tbl_department_heads d ON ep.department = d.deptname
    GROUP BY
        d.deptheadname, e.department
),
employee_salary_rank AS (
    SELECT
        e.empid,
        e.firstname,
    e.lastname,
    p.project_id,
    p.status,
    RANK() OVER (PARTITION BY p.project_id ORDER BY e.salary DESC) AS salary_rank
  FROM
    tbl_emp_detail_data e
  JOIN
    tbl_project_deatils p ON e.empid = p.emp_id_no
)
SELECT
  m.project_manager,
  m.department,
  m.employee_count,
  esr.empid,
  esr.firstname,
  esr.lastname,
  esr.project_id,
  esr.salary_rank,
  esr.status
FROM
  manager_employee_counts m
JOIN
  employee_salary_rank esr ON m.department = (SELECT department FROM tbl_emp_detail_data WHERE empid = esr.empid)
ORDER BY
  m.employee_count DESC, esr.salary_rank;
```

#9. Rank Departments by Total Bonus Distributed and Within Each Department Rank Employees Based on Bonus Received

```
WITH department_bonus_totals AS (
  SELECT
    e.department,
    SUM(b.bonusamount) AS total_bonus
  FROM
    tbl_bonus b
  JOIN
    tbl_emp_detail_data e ON b.empid = e.empid
  GROUP BY
    e.department
),
department_ranks AS (
  SELECT
    department,
    total_bonus,
    RANK() OVER (ORDER BY total_bonus DESC) AS dept_rank
  FROM
    department_bonus_totals
),
employee_bonus_ranks AS (
  SELECT
    e.empid,
    e.firstname,
    e.lastname,
    e.department,
    b.bonusamount,
    RANK() OVER (PARTITION BY e.department ORDER BY b.bonusamount DESC) AS bonus_rank
  FROM
```

```
        tbl_emp_detail_data e
    LEFT JOIN
        tbl_bonus b ON e.empid = b.empid
)
SELECT
    dr.dept_rank,
    dr.department,
    ebr.empid,
    ebr.firstname,
    ebr.lastname,
    ebr.bonusamount,
    ebr.bonus_rank
FROM
    department_ranks dr
JOIN
    employee_bonus_ranks ebr ON dr.department = ebr.department
ORDER BY
    dr.dept_rank, ebr.bonus_rank;
```

#10. Rank Employees Based on Years of Experience and Project Count, and Rank Departments Based on Average Experience

```
WITH employee_experience AS (
    SELECT
        empid,
        firstname,
        lastname,
        department,
        DATEDIFF(year, joiningdate, CURRENT_DATE) AS years_experience
    FROM
        tbl_emp_detail_data
),
employee_project_counts AS (
    SELECT
        emp_id_no,
        COUNT(DISTINCT project_id) AS project_count
    FROM
        tbl_project_deatils
    GROUP BY
        emp_id_no
),
employee_ranks AS (
    SELECT
        e.empid,
        e.firstname,
        e.lastname,
        e.department,
        e.years_experience,
        COALESCE(pc.project_count, 0) AS project_count,
        RANK() OVER (ORDER BY e.years_experience DESC, COALESCE(pc.project_count, 0) DESC) AS employee_rank
    FROM
        employee_experience e
    LEFT JOIN
        employee_project_counts pc ON e.empid = pc.emp_id_no
),
department_avg_exp AS (
    SELECT
        department,
        AVG(years_experience) AS avg_experience,
        RANK() OVER (ORDER BY AVG(years_experience) DESC) AS department_rank
    FROM
        employee_experience
    GROUP BY
        department
)
SELECT
    dr.department_rank,
    dr.department,
    er.empid,
    er.firstname,
    er.lastname,
    er.years_experience,
    er.project_count,
    er.employee_rank
FROM
    department_avg_exp dr
JOIN
    employee_ranks er ON dr.department = er.department
ORDER BY
    dr.department_rank, er.employee_rank;

# solutions for the CTE Basic Problems:

#1. Write a CTE that Retrieves Employees with Their Department and Project Details
WITH employee_projects AS (
    SELECT
        e.empid,
        e.firstname,
        e.lastname,
        e.department,
        p.project_name,
        p.status
    FROM
        tbl_emp_detail_data e
    LEFT JOIN
        tbl_project_deatils p ON e.empid = p.emp_id_no
)
SELECT * FROM employee_projects;

# 2. Use a CTE to Find Employees Who Have Worked on More Than One Project
WITH project_counts AS (
    SELECT
        emp_id_no,
        COUNT(DISTINCT project_id) AS project_count
    FROM
        tbl_project_deatils
    GROUP BY
        emp_id_no
)
SELECT
    e.empid,
    e.firstname,
    e.lastname,
    pc.project_count
FROM
```

```
    tbl_emp_detail_data e
JOIN
    project_counts pc ON e.empid = pc.emp_id_no
WHERE
    pc.project_count > 1;
```

#3. Create a CTE to Find Employees Earning More Than the Average Salary of Their Department

```
WITH dept_avg_salary AS (
    SELECT
        department,
        AVG(salary) AS avg_salary
    FROM
        tbl_emp_detail_data
    GROUP BY
        department
)
SELECT
    e.empid,
    e.firstname,
    e.lastname,
    e.salary,
    e.department
FROM
    tbl_emp_detail_data e
JOIN
    dept_avg_salary d ON e.department = d.department
WHERE
    e.salary > d.avg_salary;
```

#4. Use a CTE and JOINs to Fetch Employees Who Joined in the Last Two Years Along with Project Names

```
WITH recent_joins AS (
    SELECT
        *
    FROM
        tbl_emp_detail_data
    WHERE
        joiningdate >= DATE(year, -2, CURRENT_DATE)
)
SELECT
    rj.empid,
    rj.firstname,
    rj.lastname,
    rj.department,
    p.project_name
FROM
    recent_joins rj
LEFT JOIN
    tbl_project_deatils p ON rj.empid = p.emp_id_no;
```

#5. Create a CTE to Calculate Department-wise Salary Statistics (Sum, Avg, Max)

```
WITH dept_salary_stats AS (
    SELECT
        department,
        SUM(salary) AS total_salary,
      AVG(salary) AS average_salary,
      MAX(salary) AS max_salary
   FROM
      tbl_emp_detail_data
   GROUP BY
      department
)
SELECT * FROM dept_salary_stats;
```

#6. Use a CTE with RANK() to Find the Top 5 Highest-paid Employees

```
WITH ranked_employees AS (
   SELECT
      empid,
      firstname,
      lastname,
      salary,
      RANK() OVER (ORDER BY salary DESC) AS salary_rank
   FROM
      tbl_emp_detail_data
)
SELECT * FROM ranked_employees WHERE salary_rank <= 5;
```

#7. Write a CTE to Find Employees with the Longest Tenure in Their Department

```
WITH tenure_rank AS (
   SELECT
      empid,
      firstname,
      lastname,
      department,
      joiningdate,
      RANK() OVER (PARTITION BY department ORDER BY joiningdate ASC) AS tenure_rank
   FROM
      tbl_emp_detail_data
)
SELECT * FROM tenure_rank WHERE tenure_rank = 1;
```

#8. Use a CTE with GROUP BY to Count Employees by Department and Classify Them as Small, Medium, or Large

```
WITH dept_counts AS (
   SELECT
      department,
      COUNT(empid) AS employee_count
   FROM
      tbl_emp_detail_data
   GROUP BY
      department
),
dept_size AS (
   SELECT
      department,
      employee_count,
      CASE
         WHEN employee_count < 5 THEN 'Small'
         WHEN employee_count BETWEEN 5 AND 10 THEN 'Medium'
         ELSE 'Large'
```

```
        END AS size_category
    FROM
        dept_counts
)
SELECT * FROM dept_size;

#10. Write a Query Using a CTE, JOINs, and RANK() to Find the Second-Highest-Paid Employee in Each Department

WITH ranked_salaries AS (
    SELECT
        empid,
        firstname,
        lastname,
        department,
        salary,
        RANK() OVER (PARTITION BY department ORDER BY salary DESC) AS salary_rank
    FROM
        tbl_emp_detail_data
)
SELECT * FROM ranked_salaries WHERE salary_rank = 2;

#Continuing with Advanced CTE Problems solutions:

 #1. Find Departments with Total Compensation (Salary + Bonus) > 300,000 and Rank Employees Within Those
Departments by Compensation

WITH dept_compensation AS (
    SELECT
        e.department,
        SUM(e.salary) + COALESCE(SUM(b.bonusamount), 0) AS total_compensation
    FROM
        tbl_emp_detail_data e
    LEFT JOIN
        tbl_bonus b ON e.empid = b.empid
    GROUP BY
        e.department
    HAVING
        SUM(e.salary) + COALESCE(SUM(b.bonusamount), 0) > 300000
),
employees_compensation AS (
    SELECT
        e.empid,
        e.firstname,
        e.lastname,
        e.department,
        e.salary + COALESCE(b.bonusamount, 0) AS total_comp
    FROM
        tbl_emp_detail_data e
    LEFT JOIN
        tbl_bonus b ON e.empid = b.empid
)
SELECT
    dc.department,
    ec.empid,
    ec.firstname,
    ec.lastname,
    ec.total_comp,
    RANK() OVER (PARTITION BY ec.department ORDER BY ec.total_comp DESC) AS emp_rank
FROM
    dept_compensation dc
JOIN
    employees_compensation ec ON dc.department = ec.department
ORDER BY
    dc.department, emp_rank;
```

#2. Find Departments Where Average Experience > 3 Years and Within Those Departments Rank Employees by Project Count

```
WITH employee_experience AS (
    SELECT
        empid,
        department,
        DATEDIFF(year, joiningdate, CURRENT_DATE) AS experience_years
    FROM
        tbl_emp_detail_data
),
dept_avg_experience AS (
    SELECT
        department,
        AVG(experience_years) AS avg_exp
    FROM
        employee_experience
    GROUP BY
        department
    HAVING
        AVG(experience_years) > 3
),
employee_project_counts AS (
    SELECT
        emp_id_no,
        COUNT(DISTINCT project_id) AS project_count
    FROM
        tbl_project_deatils
    GROUP BY
        emp_id_no
),
employee_ranks AS (
    SELECT
        e.empid,
        e.firstname,
        e.lastname,
        e.department,
        COALESCE(pc.project_count, 0) AS project_count,
        RANK() OVER (PARTITION BY e.department ORDER BY COALESCE(pc.project_count, 0) DESC) AS emp_rank
    FROM
        tbl_emp_detail_data e
    LEFT JOIN
        employee_project_counts pc ON e.empid = pc.emp_id_no
    WHERE
        e.department IN (SELECT department FROM dept_avg_experience)
)
SELECT
    *
FROM
```

```
   employee_ranks
ORDER BY
   department, emp_rank;
```

#3. Identify Project Managers (Department Heads) Whose Department's Total Bonus Exceeds 50,000; Rank Departments and Employees by Bonus

```
WITH department_bonus_totals AS (
   SELECT
      e.department,
      SUM(b.bonusamount) AS total_bonus
   FROM
      tbl_bonus b
   JOIN
      tbl_emp_detail_data e ON b.empid = e.empid
   GROUP BY
      e.department
   HAVING
      SUM(b.bonusamount) > 50000
),
project_managers AS (
   SELECT
      deptname AS department,
      deptheadname AS manager
   FROM
      tbl_department_heads
),
employees_bonus_ranked AS (
   SELECT
      e.empid,
      e.firstname,
      e.lastname,
      e.department,
      COALESCE(b.bonusamount,0) AS bonus,
      RANK() OVER (PARTITION BY e.department ORDER BY COALESCE(b.bonusamount,0) DESC) AS bonus_rank
   FROM
      tbl_emp_detail_data e
   LEFT JOIN
      tbl_bonus b ON e.empid = b.empid
)
SELECT
   dm.manager,
   dbt.department,
   dbt.total_bonus,
   ebr.empid,
   ebr.firstname,
   ebr.lastname,
   ebr.bonus,
   ebr.bonus_rank
FROM
   department_bonus_totals dbt
JOIN
   project_managers dm ON dbt.department = dm.department
JOIN
   employees_bonus_ranked ebr ON dbt.department = ebr.department
ORDER BY
   dbt.total_bonus DESC, ebr.bonus_rank;
```

#4. Find Top 2 Departments Based on Average Project Duration and Rank Employees Within Departments by Joining Date (Experience)

```
WITH project_durations AS (
   SELECT
      p.project_id,
      e.department,
      DATEDIFF(day, p.start_date, COALESCE(p.end_date, CURRENT_DATE)) AS project_duration
   FROM
      tbl_project_deatils p
   JOIN
      tbl_emp_detail_data e ON p.emp_id_no = e.empid
),
department_avg_duration AS (
   SELECT
      department,
      AVG(project_duration) AS avg_duration,
      RANK() OVER (ORDER BY AVG(project_duration) DESC) AS dept_rank
   FROM
      project_durations
   GROUP BY
      department
   HAVING
      RANK() OVER (ORDER BY AVG(project_duration) DESC) <= 2
),
employee_ranked AS (
   SELECT
      e.empid,
      e.firstname,
      e.lastname,
      e.department,
      e.joiningdate,
      RANK() OVER (PARTITION BY e.department ORDER BY e.joiningdate ASC) AS emp_rank
   FROM
      tbl_emp_detail_data e
   WHERE
      e.department IN (SELECT department FROM department_avg_duration)
)
SELECT
   d.department,
   d.avg_duration,
   er.empid,
   er.firstname,
   er.lastname,
   er.joiningdate,
   er.emp_rank
FROM
   department_avg_duration d
JOIN
   employee_ranked er ON d.department = er.department
ORDER BY
   d.dept_rank, er.emp_rank;
```

#5. Find Employees Who Worked on More Than One Completed Project, Belong to Departments with Avg Salary > 55k, and Rank by Salary and Project Count

```
WITH completed_projects AS (
   SELECT
```

```
        emp_id_no,
        COUNT(DISTINCT project_id) AS completed_project_count
    FROM
        tbl_project_deatils
    WHERE
        status = 'Completed'
    GROUP BY
        emp_id_no
),
dept_avg_salary AS (
    SELECT
        department,
        AVG(salary) AS avg_salary
    FROM
        tbl_emp_detail_data
    GROUP BY
        department
    HAVING
        AVG(salary) > 55000
),
qualified_employees AS (
    SELECT
        e.empid,
        e.firstname,
        e.lastname,
        e.department,
        e.salary,
        COALESCE(cp.completed_project_count, 0) AS completed_projects
    FROM
        tbl_emp_detail_data e
    LEFT JOIN
        completed_projects cp ON e.empid = cp.emp_id_no
    WHERE
        e.department IN (SELECT department FROM dept_avg_salary)
        AND COALESCE(cp.completed_project_count, 0) > 1
)
SELECT
    empid,
    firstname,
    lastname,
    department,
    salary,
    completed_projects,
    RANK() OVER (ORDER BY salary DESC, completed_projects DESC) AS emp_rank
FROM
    qualified_employees
ORDER BY
    emp_rank;
```

#6. Departments with More Than 5 Employees; Rank Employees by Total Compensation (Salary + Bonus) & Experience

```
WITH dept_employee_counts AS (
    SELECT
        department,
        COUNT(empid) AS emp_count
    FROM
```

```
      tbl_emp_detail_data
   GROUP BY
      department
   HAVING
      COUNT(empid) > 5
),
employee_compensation_experience AS (
   SELECT
      e.empid,
      e.firstname,
      e.lastname,
      e.department,
      e.salary,
      COALESCE(b.bonusamount, 0) AS bonus,
      e.salary + COALESCE(b.bonusamount, 0) AS total_comp,
      DATEDIFF(year, e.joiningdate, CURRENT_DATE) AS experience_years
   FROM
      tbl_emp_detail_data e
   LEFT JOIN
      tbl_bonus b ON e.empid = b.empid
),
ranked_employees AS (
   SELECT
      empid,
      firstname,
      lastname,
      department,
      total_comp,
      experience_years,
      RANK() OVER (PARTITION BY department ORDER BY total_comp DESC, experience_years DESC) AS emp_rank
   FROM
      employee_compensation_experience
   WHERE
      department IN (SELECT department FROM dept_employee_counts)
)
SELECT * FROM ranked_employees ORDER BY department, emp_rank;
```

#7. Employees with >2 Projects in Departments Where Dept Head Name Starts with 'M', Rank by Salary & Project Count

```
WITH manager_departments AS (
   SELECT deptname
   FROM tbl_department_heads
   WHERE deptheadname LIKE 'M%'
),
employee_project_counts AS (
   SELECT emp_id_no, COUNT(DISTINCT project_id) AS project_count
   FROM tbl_project_deatils
   GROUP BY emp_id_no
),
qualified_employees AS (
   SELECT e.empid, e.firstname, e.lastname, e.department, e.salary, COALESCE(pc.project_count, 0) AS project_count
   FROM tbl_emp_detail_data e
   LEFT JOIN employee_project_counts pc ON e.empid = pc.emp_id_no
   WHERE e.department IN (SELECT deptname FROM manager_departments)
    AND COALESCE(pc.project_count, 0) > 2
)
```

```
SELECT *, RANK() OVER (ORDER BY salary DESC, project_count DESC) AS emp_rank
FROM qualified_employees
ORDER BY emp_rank;
```

#8. Departments with >5 Projects, Calculate Avg Project Duration; Rank Departments & Employees by Project Completion

```
WITH dept_project_counts AS (
   SELECT
      e.department,
      COUNT(DISTINCT p.project_id) AS project_count,
      AVG(DATEDIFF(day, p.start_date, COALESCE(p.end_date, CURRENT_DATE))) AS avg_project_duration
   FROM tbl_project_deatils p
   JOIN tbl_emp_detail_data e ON p.emp_id_no = e.empid
   GROUP BY e.department
   HAVING COUNT(DISTINCT p.project_id) > 5
),
department_ranks AS (
   SELECT
      department,
      avg_project_duration,
      RANK() OVER (ORDER BY avg_project_duration DESC) AS dept_rank
   FROM dept_project_counts
),
employee_completed_projects AS (
   SELECT e.empid, e.firstname, e.lastname, e.department,
         COUNT(CASE WHEN p.status='Completed' THEN 1 END) AS completed_projects
   FROM tbl_emp_detail_data e
   LEFT JOIN tbl_project_deatils p ON e.empid = p.emp_id_no
   GROUP BY e.empid, e.firstname, e.lastname, e.department
),
employee_ranks AS (
   SELECT *,
         RANK() OVER (PARTITION BY department ORDER BY completed_projects DESC) AS emp_rank
   FROM employee_completed_projects
   WHERE department IN (SELECT department FROM dept_project_counts)
)
SELECT dr.department, dr.avg_project_duration, dr.dept_rank,
      er.empid, er.firstname, er.lastname, er.completed_projects, er.emp_rank
FROM department_ranks dr
JOIN employee_ranks er ON dr.department = er.department
ORDER BY dr.dept_rank, er.emp_rank;
```

#9. Employees With Bonuses Greater Than Department Average Bonus, Rank Departments & Employees by Bonus and Total Compensation

```
WITH dept_avg_bonus AS (
   SELECT e.department, AVG(b.bonusamount) AS avg_bonus
   FROM tbl_bonus b
   JOIN tbl_emp_detail_data e ON b.empid = e.empid
   GROUP BY e.department
),
qualified_employees AS (
   SELECT e.empid, e.firstname, e.lastname, e.department, e.salary, b.bonusamount,
         e.salary + b.bonusamount AS total_comp
   FROM tbl_emp_detail_data e
   JOIN tbl_bonus b ON e.empid = b.empid
   JOIN dept_avg_bonus dab ON e.department = dab.department
   WHERE b.bonusamount > dab.avg_bonus
),
department_bonus_totals AS (
   SELECT department, SUM(bonusamount) AS total_department_bonus
   FROM qualified_employees
   GROUP BY department
),
department_ranks AS (
   SELECT department, total_department_bonus,
         RANK() OVER(ORDER BY total_department_bonus DESC) AS dept_rank
   FROM department_bonus_totals
),
employee_ranks AS (
   SELECT *,
         RANK() OVER (PARTITION BY department ORDER BY total_comp DESC) AS emp_rank
   FROM qualified_employees
)
SELECT dr.dept_rank, dr.department, er.empid, er.firstname, er.lastname, er.bonusamount, er.total_comp,
er.emp_rank
FROM department_ranks dr
JOIN employee_ranks er ON dr.department = er.department
ORDER BY dr.dept_rank, er.emp_rank;
```

#10. Departments With Dept Head Name Containing 'a', Avg Employee Experience > 4 Years; Rank Employees by Projects and Compensation

```
WITH qualifying_departments AS (
   SELECT d.deptname
   FROM tbl_department_heads d
   JOIN tbl_emp_detail_data e ON d.deptname = e.department
   GROUP BY d.deptname
   HAVING AVG(DATEDIFF(year, e.joiningdate, CURRENT_DATE)) > 4
     AND d.deptheadname LIKE '%a%'
),
employee_project_counts AS (
   SELECT emp_id_no, COUNT(DISTINCT project_id) AS project_count
   FROM tbl_project_deatils
   GROUP BY emp_id_no
),
employee_compensation AS (
   SELECT e.empid, e.firstname, e.lastname, e.department, e.salary, COALESCE(b.bonusamount,0) AS bonus,
         COALESCE(pc.project_count,0) AS project_count, e.salary + COALESCE(b.bonusamount,0) AS total_comp
   FROM tbl_emp_detail_data e
   LEFT JOIN tbl_bonus b ON e.empid = b.empid
   LEFT JOIN employee_project_counts pc ON e.empid = pc.emp_id_no
   WHERE e.department IN (SELECT deptname FROM qualifying_departments)
),
employee_ranks AS (
   SELECT *,
         RANK() OVER (PARTITION BY department ORDER BY project_count DESC, total_comp DESC) AS emp_rank
   FROM employee_compensation
)
SELECT * FROM employee_ranks ORDER BY department, emp_rank;
```