# SQL PRACTICE

DANANG WISNU PRAYOGA

# Question 2

Show the last name, job, salary, and commission of those employees who earn commission. Sort  the data by the salary in descending order.

```
SELECT last_name, job_id AS job, salary, commission_pct
FROM employees
WHERE commission_pct IS NOT NULL
ORDER BY salary DESC;
```

Script Output

Output | Errors | Grid 1 | Environment

| LAST_NAME | JOB | SALARY | COMMISSION_PCT |
|---|---|---|---|
| Abel | SA_REP | 11000 | .3 |
| Zlotkey | SA_MAN | 10500 | .2 |
| Taylor | SA_REP | 8600 | .2 |
| Grant | SA_REP | 7000 | .15 |

# Question 3

Show the employees that have no commission with a 10% raise in their salary (round off the salaries).

```
SELECT emp.employee_id,
       emp.first_name,
       emp.last_name,
       emp.salary,
       emp.commission_pct,
       ROUND(emp.salary * 0.1) + emp.salary AS "RAISE_SAL"
FROM employees emp
WHERE emp.commission_pct IS NULL
ORDER BY emp.salary DESC;
```

Script Output

Output | Errors | Grid 1 | Environment

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | COMMISSION_PCT | RAISE_SAL |
|---|---|---|---|---|---|
| 100 | Steven | King | 24000 | | 26400 |
| 101 | Neena | Kochtar | 17000 | | 18700 |
| 102 | Lex | De Haan | 17000 | | 18700 |
| 201 | Michael | Hartstein | 13000 | | 14300 |
| 205 | Shelley | Higgins | 12000 | | 13200 |
| 103 | Alexander | Hunold | 9000 | | 9900 |
| 206 | William | Gietz | 8300 | | 9130 |
| 202 | Pat | Fay | 6000 | | 6600 |
| 104 | Bruce | Ernst | 6000 | | 6600 |
| 124 | Kevin | Mourgos | 5800 | | 6380 |
| 200 | Jennifer | Whalen | 4400 | | 4840 |
| 107 | Diana | Lorentz | 4200 | | 4620 |
| 141 | Trenna | Rajs | 3500 | | 3850 |
| 142 | Curtis | Davies | 3100 | | 3410 |
| 143 | Randall | Matos | 2600 | | 2860 |
| 144 | Peter | Vargas | 2500 | | 2750 |

# Question 4

Show the last names of all employees together with the number of years and the number of completed months that they have been employed.

```
SELECT emp.last_name,
       FLOOR((SYSDATE - emp.hire_date) / 365) AS "YEARS",
       FLOOR(MOD((SYSDATE - emp.hire_date), 365) / 30) AS "MONTHS"
FROM employees emp;
```

Script Output

Output | Errors | Grid 1 | Environment

| LAST_NAME | YEARS | MONTHS |
|---|---|---|
| King | 37 | 7 |
| Kochtar | 35 | 3 |
| De Haan | 32 | 0 |
| Hunold | 35 | 0 |
| Ernst | 33 | 8 |
| Lorentz | 25 | 11 |
| Mourgos | 25 | 2 |
| Rajs | 29 | 3 |
| Davies | 27 | 11 |
| Matos | 26 | 10 |
| Vargas | 26 | 6 |
| Zlotkey | 24 | 11 |
| Abel | 28 | 8 |
| Taylor | 26 | 9 |
| Grant | 25 | 7 |
| Whalen | 37 | 4 |
| Hartstein | 28 | 11 |
| Fay | 27 | 5 |

# Question 5

Show those employees that have a name starting with J, K, L, or M.

```
SELECT *
FROM employees
WHERE SUBSTR(first_name, 1, 1) IN ('J', 'K', 'L', 'M');
```

Script Output

Output | Errors | Grid 1 | Environment

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | PHONE_NUMBER | HIRE_DATE | JOB_ID | SALARY | COMMISSION_PCT | MANAGER_ID | DEPARTMENT_ID |
|---|---|---|---|---|---|---|---|---|---|---|
| 102 | Lex | De Haan | LDEHAAN | 515.123.4569 | 13-JAN-93 | AD_VP | 17000 | | 100 | 90 |
| 124 | Kevin | Mourgos | KMOURGOS | 650.123.5234 | 16-NOV-99 | ST_MAN | 5800 | | 100 | 50 |
| 176 | Jonathon | Taylor | JTAYLOR | 011.44.1644.429265 | 24-MAR-98 | SA_REP | 8600 | .2 | 149 | 80 |
| 178 | Kimberly | Grant | KGRANT | 011.44.1644.429263 | 24-MAY-99 | SA_REP | 7000 | .15 | 149 | 80 |
| 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 |

# Question 6

Show all employees, and indicate with “Yes” or “No” whether they receive a commission.

```
SELECT employee_id,
       first_name,
       last_name,
       email,
       phone_number,
       job_id,
       hire_date,
       salary,
       DECODE(commission_pct, NULL, 'NO', 'YES') AS COMMISION,
       manager_id,
       department_id
FROM employees;
```

Script Output

Output | Errors | Grid 1 | Environment

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | PHONE_NUMBER | JOB_ID | HIRE_DATE | SALARY | COMMISION | MANAGER_ID | DEPARTMENT_ |
|---|---|---|---|---|---|---|---|---|---|---|
| 100 | Steven | King | SKING | 515.123.4567 | AD_PRES | 17-JUN-87 | 24000 | NO | | 90 |
| 101 | Neena | Kochtar | NKOCHTAR | 515.123.4568 | AD_VP | 21-SEP-89 | 17000 | NO | 100 | 90 |
| 102 | Lex | De Haan | LDEHAAN | 515.123.4569 | AD_VP | 13-JAN-93 | 17000 | NO | 100 | 90 |
| 103 | Alexander | Hunold | AHUNOLD | 515.123.4567 | IT_PROG | 03-JAN-90 | 9000 | NO | 102 | 60 |
| 104 | Bruce | Ernst | BERNST | 515.123.4568 | IT_PROG | 21-MAY-91 | 6000 | NO | 103 | 60 |
| 107 | Diana | Lorentz | DLORENTZ | 515.123.5567 | IT_PROG | 07-FEB-99 | 4200 | NO | 103 | 60 |
| 124 | Kevin | Mourgos | KMOURGOS | 650.123.5234 | ST_MAN | 16-NOV-99 | 5800 | NO | 100 | 50 |
| 141 | Trenna | Rajs | TRAJS | 650.121.8009 | ST_CLERK | 17-OCT-95 | 3500 | NO | 124 | 50 |
| 142 | Curtis | Davies | CDAVIES | 650.121.2994 | ST_CLERK | 29-JAN-97 | 3100 | NO | 124 | 50 |
| 143 | Randall | Matos | RMATOS | 650.121.2874 | ST_CLERK | 15-MAR-98 | 2600 | NO | 124 | 50 |
| 144 | Peter | Vargas | PVARGAS | 650.121.2004 | ST_CLERK | 09-JUL-98 | 2500 | NO | 124 | 50 |
| 149 | Eleny | Zlotkey | EZLOTKEY | 011.44.1344.429018 | SA_MAN | 29-JAN-00 | 10500 | YES | 100 | 80 |
| 174 | Ellen | Abel | EABEL | 011.44.1644.429267 | SA_REP | 11-MAY-96 | 11000 | YES | 149 | 80 |
| 176 | Jonathon | Taylor | JTAYLOR | 011.44.1644.429265 | SA_REP | 24-MAR-98 | 8600 | YES | 149 | 80 |
| 178 | Kimberly | Grant | KGRANT | 011.44.1644.429263 | SA_REP | 24-MAY-99 | 7000 | YES | 149 | 80 |
| 200 | Jennifer | Whalen | JWHALEN | 515.123.4444 | AD_ASST | 17-SEP-87 | 4400 | NO | 101 | 10 |
| 201 | Michael | Hartstein | MHARTSTE | 515.123.5555 | MK_MAN | 17-FEB-96 | 13000 | NO | 100 | 20 |
| 202 | Pat | Fay | PFAY | 603.123.6666 | MK_REP | 17-AUG-97 | 6000 | NO | 201 | 20 |

# Question 7

Show the department names, locations, names, job titles, and salaries of employees who work in location 1800.

```
SELECT dep.department_name,
       dep.location_id,
       emp.first_name,
       job.job_title,
       emp.salary
FROM employees emp
JOIN departments dep ON emp.department_id = dep.department_id
JOIN jobs job ON emp.job_id = job.job_id
WHERE dep.location_id = '1800';
```

Script Output

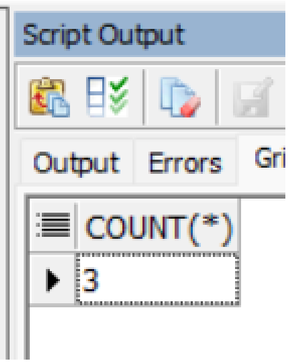
Output | Errors | Grid 1 | Environment

| DEPARTMENT_NAME | LOCATION_ID | FIRST_NAME | JOB_TITLE | SALARY |
|---|---|---|---|---|
| Marketing | 1800 | Michael | Marketing Manager | 13000 |
| Marketing | 1800 | Pat | Marketing Representative | 6000 |

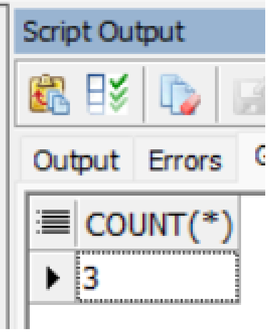
# Question 8

How many employees have a last name that ends with an n? Create two possible solutions.

```
SELECT COUNT(*)
FROM employees
WHERE LOWER(last_name) LIKE '%n';
```




```
SELECT COUNT(*)
FROM employees
WHERE UPPER(last_name) LIKE '%N';
```



# Question 9

Show the names and locations for all departments, and the number of employees working in each department. Make sure that departments without employees are included as well.

```
SELECT dep.department_name,
       loc.street_address || ' ' || loc.city || ' ' || loc.state_province || ' ' ||
       loc.country_id || ' ' || loc.postal_code AS "LOCATION",
       COUNT(emp.first_name)
FROM departments dep
JOIN locations loc ON dep.location_id = loc.location_id
LEFT JOIN employees emp ON dep.department_id = emp.department_id
GROUP BY dep.department_name,
         loc.street_address || ' ' || loc.city || ' ' || loc.state_province || ' ' ||
         loc.country_id || ' ' || loc.postal_code;
```

Script Output

Output | Errors | Grid 1 | Environment

| DEPARTMENT_NAME | LOCATION | COUNT(EMP.FIRST_NAME) |
|---|---|---|
| Accounting | 2004 Charade Rd Seattle Washington U... | 2 |
| IT | 2014 Jabberwocky Rd Southlake Texas ... | 3 |
| Sales | Magdalen Centre, The Oxford Science P... | 4 |
| Administration | 2004 Charade Rd Seattle Washington U... | 1 |
| Executive | 2004 Charade Rd Seattle Washington U... | 3 |
| Marketing | 147 Spadina Ave Toronto Ontario CA M5... | 2 |
| Shipping | 2011 Interiors Blvd South San Francisco ... | 5 |
| Contracting | 2004 Charade Rd Seattle Washington U... | 0 |

# Question 10

Which jobs are found in departments 10 and 20?

```
SELECT DISTINCT job.job_title
FROM jobs job
JOIN employees emp ON emp.job_id = job.job_id
JOIN departments dep ON dep.department_id = emp.department_id
WHERE dep.department_id IN ('10', '20');
```

Script Output

Output | Errors | Grid 1 | Environme

| JOB_TITLE |
| --- |
| Marketing Manager |
| Administration Assistant |
| Marketing Representative |

# Question 11

Which jobs are found in the Administration and Executive departments, and how many employees do these jobs? Show the job with the highest frequency first.

```
SELECT job.job_title,
       COUNT(emp.employee_id) AS employee_count
FROM jobs job
JOIN employees emp ON emp.job_id = job.job_id
JOIN departments dep ON dep.department_id = emp.department_id
WHERE dep.department_name IN ('Administration', 'Executive')
GROUP BY job.job_title
ORDER BY employee_count DESC;
```

Script Output

Output | Errors | Grid 1 | Environment

| JOB_TITLE | EMPLOYEE_COUNT |
|---|---|
| Administration Vice President | 2 |
| Administration Assistant | 1 |
| President | 1 |

# Question 12

Show all employees who were hired in the first half of the month (before the 16th of the month).

```
SELECT *
FROM employees
WHERE TO_CHAR(hire_date, 'DD') < '16';
```

Script Output

Output | Errors | Grid 1 | Environment

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | PHONE_NUMBER | HIRE_DATE | JOB_ID | SALARY | COMMISSION_PCT | MANAGER_ID | DEPARTMENT_ID |
|---|---|---|---|---|---|---|---|---|---|---|
| 102 | Lex | De Haan | LDEHAAN | 515.123.4569 | 13-JAN-93 | AD_VP | 17000 | | 100 | 90 |
| 103 | Alexander | Hunold | AHUNOLD | 515.123.4567 | 03-JAN-90 | IT_PROG | 9000 | | 102 | 60 |
| 107 | Diana | Lorentz | DLORENTZ | 515.123.5567 | 07-FEB-99 | IT_PROG | 4200 | | 103 | 60 |
| 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 |
| 174 | Ellen | Abel | EABEL | 011.44.1644.429267 | 11-MAY-96 | SA_REP | 11000 | .3 | 149 | 80 |
| 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 |

# Question 13

Show the names, salaries, and the number of dollars (in thousands) that all employees earn.

```
SELECT first_name || ' ' || last_name AS name,
       salary,
       salary / 1000 AS earn
FROM employees;
```

Script Output

Output | Errors | Grid 1 | Environment

| NAME | SALARY | EARN |
|---|---|---|
| Steven King | 24000 | 24 |
| Neena Kochtar | 17000 | 17 |
| Lex De Haan | 17000 | 17 |
| Alexander Hunold | 9000 | 9 |
| Bruce Ernst | 6000 | 6 |
| Diana Lorentz | 4200 | 4.2 |
| Kevin Mourgos | 5800 | 5.8 |
| Trenna Rajs | 3500 | 3.5 |
| Curtis Davies | 3100 | 3.1 |
| Randall Matos | 2600 | 2.6 |
| Peter Vargas | 2500 | 2.5 |
| Eleny Zlotkey | 10500 | 10.5 |
| Ellen Abel | 11000 | 11 |
| Jonathon Taylor | 8600 | 8.6 |
| Kimberly Grant | 7000 | 7 |
| Jennifer Whalen | 4400 | 4.4 |
| Michael Hartstein | 13000 | 13 |
| Pat Fay | 6000 | 6 |

# Question 14

Show all employees who have managers with a salary higher than $ 15,000. Show the following data: employee name, manager name, manager salary, and salary grade of the manager.

```
SELECT
    e.first_name || ' ' || e.last_name AS employee_name,
    m.first_name || ' ' || m.last_name AS manager_name,
    m.salary AS manager_salary,
    CASE
        WHEN m.salary BETWEEN g.lowest_salary AND g.highest_salary THEN g.grade_level
        ELSE 'N/A'
    END AS manager_salary_grade
FROM
    employees e
JOIN
    employees m ON e.manager_id = m.employee_id
LEFT JOIN
    job_grades g ON m.salary BETWEEN g.lowest_salary AND g.highest_salary
WHERE
    m.salary > 15000;
```

| EMPLOYEE_NAME | MANAGER_NAME | MANAGER_SALARY | MANAGER_SALARY_G... |
|---|---|---|---|
| Neena Kochtar | Steven King | 24000 | E |
| Lex De Haan | Steven King | 24000 | E |
| Kevin Mourgos | Steven King | 24000 | E |
| Eleny Zlotkey | Steven King | 24000 | E |
| Michael Hartstein | Steven King | 24000 | E |
| Jennifer Whalen | Neena Kochtar | 17000 | E |
| Shelley Higgins | Neena Kochtar | 17000 | E |
| Alexander Hunold | Lex De Haan | 17000 | E |

# Question 15

Show the department number, name, number of employees, and average salary of all departments,  together with the names, salaries, and jobs of the employees working in each department.

```sql
SELECT
    d.department_id AS department_number,
    d.department_name AS department_name,
    COUNT(e.employee_id) AS number_of_employees,
    ROUND(AVG(e.salary), 2) AS average_salary,
    e.first_name || ' ' || e.last_name AS employee_name,
    e.salary AS employee_salary,
    j.job_title AS employee_job
FROM
    departments d
LEFT JOIN
    employees e ON d.department_id = e.department_id
LEFT JOIN
    jobs j ON e.job_id = j.job_id
GROUP BY
    d.department_id, d.department_name, e.first_name, e.last_name, e.salary, j.job_title
ORDER BY
    d.department_id, e.salary DESC;
```

| DEPARTMENT_NUMBER | DEPARTMENT_NAME | NUMBER_OF_EMPLOYEES | AVERAGE_SALARY | EMPLOYEE_NAME | EMPLOYEE_SALARY | EMPLOYEE_JOB |
|---|---|---|---|---|---|---|
| 10 | Administration | 1 | 4400 | Jennifer Whalen | 4400 | Administration Assistant |
| 20 | Marketing | 1 | 13000 | Michael Hartstein | 13000 | Marketing Manager |
| 20 | Marketing | 1 | 6000 | Pat Fay | 6000 | Marketing Representative |
| 50 | Shipping | 1 | 5800 | Kevin Mourgos | 5800 | Stock Manager |
| 50 | Shipping | 1 | 3500 | Trenna Rajs | 3500 | Stock Clerk |
| 50 | Shipping | 1 | 3100 | Curtis Davies | 3100 | Stock Clerk |
| 50 | Shipping | 1 | 2600 | Randall Matos | 2600 | Stock Clerk |
| 50 | Shipping | 1 | 2500 | Peter Vargas | 2500 | Stock Clerk |
| 60 | IT | 1 | 9000 | Alexander Hunold | 9000 | Programmer |
| 60 | IT | 1 | 6000 | Bruce Ernst | 6000 | Programmer |
| 60 | IT | 1 | 4200 | Diana Lorentz | 4200 | Programmer |
| 80 | Sales | 1 | 11000 | Ellen Abel | 11000 | Sales Representative |
| 80 | Sales | 1 | 10500 | Eleny Zlotkey | 10500 | Sales Manager |
| 80 | Sales | 1 | 8600 | Jonathon Taylor | 8600 | Sales Representative |

# Question 16

Show the department number and the lowest salary of the department with the highest average salary.

```sql
SELECT
    department_id AS department_number,
    MIN(salary) AS lowest_salary
FROM
    employees
GROUP BY
    department_id
HAVING
    AVG(salary) = (
        SELECT
            MAX(AVG(salary))
        FROM
            employees
        GROUP BY
            department_id
    );
```

| DEPARTMENT_NUMBER | LOWEST_SALARY |
|---|---|
| 90 | 17000 |

# Question 17

Show the department numbers, names, and locations of the departments where no sales representatives work.

```
3       d.department_name,
4       l.city AS location
5   FROM
6       departments d
7   JOIN
8       locations l ON d.location_id = l.location_id
9   WHERE
10      d.department_id NOT IN (
11          SELECT
12              e.department_id
13          FROM
14              employees e
15          JOIN
16              jobs j ON e.job_id = j.job_id
17          WHERE
```

| DEPARTMENT_NUMBER | DEPARTMENT_NAME | LOCATION |
|---|---|---|
| 60 | IT | Southlake |
| 50 | Shipping | South San Francisco |
| 10 | Administration | Seattle |
| 90 | Executive | Seattle |
| 110 | Accounting | Seattle |
| 190 | Contracting | Seattle |
| 20 | Marketing | Toronto |

# Question 18 (a)

Show the department number, department name, and the number of employees working in each department that:
a. Includes fewer than 3 employees
b. Has the highest number of employees
c. Has the lowest number of employees

```
1  SELECT
2      dep.department_id,
3      dep.department_name,
4      COUNT(emp.first_name) AS employee_count
5  FROM
6      departments dep
7  LEFT JOIN
8      employees emp ON dep.department_id = emp.department_id
9  GROUP BY
10     dep.department_id, dep.department_name
11 HAVING
12     COUNT(emp.first_name) < 3;
13
```

Script Output

Messages | Data Grid | Trace | DBMS Output (disabled) | Query Viewer | Explain Plan

Output | Errors | Grid 1 | Environment

| DEPARTMENT_ID | DEPARTMENT_NAME | EMPLOYEE_COUNT |
|---|---|---|
| 190 | Contracting | 0 |
| 110 | Accounting | 2 |
| 10 | Administration | 1 |
| 20 | Marketing | 2 |

# Question 18 (b)

Show the department number, department name, and the number of employees working in each department that:
a. Includes fewer than 3 employees
b. Has the highest number of employees
c. Has the lowest number of employees

```
SELECT dep.department_id,
       dep.department_name,
       COUNT(emp.first_name) AS employee_count
FROM departments dep
LEFT JOIN employees emp ON dep.department_id = emp.department_id
GROUP BY dep.department_id, dep.department_name
HAVING COUNT(emp.first_name) = (
    SELECT MAX(employee_count)
    FROM (
        SELECT d.department_id,
               COUNT(e.first_name) AS employee_count
        FROM departments d
        LEFT JOIN employees e ON d.department_id = e.department_id
        GROUP BY d.department_id
    )
);
```

Script Output

Messages | Data Grid | Trace | DBMS Output (disabled) | Query Viewer | Explain Plan | Script Out

Output | Errors | Grid 1 | Environment

| DEPARTMENT_ID | DEPARTMENT_NAME | EMPLOYEE_COUNT |
|---|---|---|
| 50 | Shipping | 5 |

# Question 18 (c)

Show the department number, department name, and the number of employees working in each department that:
a. Includes fewer than 3 employees
b. Has the highest number of employees
c. Has the lowest number of employees

```sql
SELECT dep.department_id,
       dep.department_name,
       COUNT(emp.first_name) AS employee_count
FROM departments dep
LEFT JOIN employees emp ON dep.department_id = emp.department_id
GROUP BY dep.department_id, dep.department_name
HAVING COUNT(emp.first_name) = (
    SELECT MIN(employee_count)
    FROM (
        SELECT d.department_id,
               COUNT(e.first_name) AS employee_count
        FROM departments d
        LEFT JOIN employees e ON d.department_id = e.department_id
        GROUP BY d.department_id
    )
);
```

Script Output

Messages | Data Grid | Trace | DBMS Output (disabled) | Query Viewer | Explain Plan | Script Ou

Output | Errors | Grid 1 | Environment

| DEPARTMENT_ID | DEPARTMENT_NAME | EMPLOYEE_COUNT |
|---|---|---|
| 190 | Contracting | 0 |

# Question 19

Show the employee number, last name, salary, department number, and the average salary in their department for all employees

```
SELECT empl.employee_id,
       empl.last_name,
       empl.salary,
       dep2.department_id,
       AVG(emp2.salary) AS avg_salary
FROM employees empl
JOIN departments dep2 ON empl.department_id = dep2.department_id
JOIN employees emp2 ON emp2.department_id = dep2.department_id
GROUP BY empl.employee_id, empl.last_name, empl.salary, dep2.department_id;
```

Script Output

Output | Errors | Grid 1 | Environment

| EMPLOYEE_ID | LAST_NAME | SALARY | DEPARTMENT_ID | AVG_SALARY |
|---|---|---|---|---|
| 143 | Matos | 2600 | 50 | 3500 |
| 144 | Vargas | 2500 | 50 | 3500 |
| 205 | Higgins | 12000 | 110 | 10150 |
| 206 | Gietz | 8300 | 110 | 10150 |
| 100 | King | 24000 | 90 | 19333.3333 |
| 101 | Kochtar | 17000 | 90 | 19333.3333 |
| 102 | De Haan | 17000 | 90 | 19333.3333 |
| 200 | Whalen | 4400 | 10 | 4400 |
| 201 | Hartstein | 13000 | 20 | 9500 |
| 202 | Fay | 6000 | 20 | 9500 |
| 103 | Hunold | 9000 | 60 | 6400 |
| 104 | Ernst | 6000 | 60 | 6400 |
| 107 | Lorentz | 4200 | 60 | 6400 |
| 149 | Zlotkey | 10500 | 80 | 9275 |
| 174 | Abel | 11000 | 80 | 9275 |
| 176 | Taylor | 8600 | 80 | 9275 |
| 178 | Grant | 7000 | 80 | 9275 |

# Question 20

Show all employees who were hired on the day of the week on which the highest number of employees were hired

```
SELECT first_name
FROM employees
WHERE TO_CHAR(hire_date, 'DAY', 'NLS_DATE_LANGUAGE=Indonesian') =
    (
        SELECT TO_CHAR(hire_date, 'DAY', 'NLS_DATE_LANGUAGE=Indonesian')
        FROM employees
        GROUP BY TO_CHAR(hire_date, 'DAY', 'NLS_DATE_LANGUAGE=Indonesian')
        ORDER BY COUNT(*) DESC
        FETCH FIRST 1 ROW ONLY
    );
```

Script Output

Output | Errors | Grid 1 | Environment

```
FIRST_NAME
--------------------
Bruce
Kevin
Trenna
Jonathon
Shelley
William

6 rows selected.
```

# Question 21

Create an anniversary overview based on the hire date of the employees. Sort the anniversaries in ascending order

```sql
SELECT
    employee_id,
    first_name || ' ' || last_name AS full_name,
    TO_CHAR(hire_date, 'DD-MON-YYYY') AS anniversary_date,
    EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM hire_date)
        AS years_of_service
FROM
    employees
ORDER BY
    TO_CHAR(hire_date, 'MMDDYYYY');
```

Script Output

Output | Errors | Grid 1 | Environment

| EMPLOYEE_ID | FULL_NAME | ANNIVERSARY_DATE | YEARS_OF_SERVICE |
|---|---|---|---|
| 103 | Alexander Hunold | 03-JAN-1990 | 35 |
| 102 | Lex De Haan | 13-JAN-1993 | 32 |
| 142 | Curtis Davies | 29-JAN-1997 | 28 |
| 149 | Eleny Zlotkey | 29-JAN-2000 | 25 |
| 107 | Diana Lorentz | 07-FEB-1999 | 26 |
| 201 | Michael Hartstein | 17-FEB-1996 | 29 |
| 143 | Randall Matos | 15-MAR-1998 | 27 |
| 176 | Jonathon Taylor | 24-MAR-1998 | 27 |
| 174 | Ellen Abel | 11-MAY-1996 | 29 |
| 104 | Bruce Ernst | 21-MAY-1991 | 34 |
| 178 | Kimberly Grant | 24-MAY-1999 | 26 |
| 205 | Shelley Higgins | 07-JUN-1994 | 31 |
| 206 | William Gietz | 07-JUN-1994 | 31 |
| 100 | Steven King | 17-JUN-1987 | 38 |
| 144 | Peter Vargas | 09-JUL-1998 | 27 |
| 202 | Pat Fay | 17-AUG-1997 | 28 |
| 200 | Jennifer Whalen | 17-SEP-1987 | 38 |
| 101 | Neena Kochtar | 21-SEP-1989 | 36 |

# Question 22

- Fungsi untuk mengetahui minus atau tidaknya suatu angka
  Inputan : -20
  Output : -1
- Fungsi untuk memaksakan panjang dari suatu karakter , walaupun inputan kurang dari karakter tersebut.
  Inputan : 1
  Output : 000001 atau 100000

```
-- Mengetahui minus atau tidak
VARIABLE input NUMBER;
EXEC :input := -20;
SELECT CASE
           WHEN :input < 0 THEN -1
           ELSE 1
       END AS result
FROM dual;
```

Script Output

Output | Errors

| RESULT |
| --- |
| -1 |

```
-- Memaksakan panjang karakter
VARIABLE input NUMBER;
EXEC :input := 1;
SELECT LPAD(:input, 6, '0') AS result
FROM dual;
```

Script Output

Output | Errors

| RESULT |
| --- |
| 000001 |

# Question 23

- Fungsi untuk mengambil format jam dari inputan date inputan : ’20-AUG-2006 10:30:31’
  output : 10
- Fungsi untuk menghapus spasi dari suatu statement yang mengandung spasi
  inputan : ‘Coba ‘
  output : ‘Coba’

```
-- Memanggil Format Jam
VARIABLE input_date VARCHAR2(20);
EXEC :input_date := '20-AUG-2006 10:30:31';
SELECT TO_CHAR(
    TO_DATE(
        :input_date, 'DD-MON-YYYY HH24:MI:SS'
        ),
        'HH24') AS hour
    FROM dual;
```

Script Output

Output | Errors

| HOUR |
| --- |
| 10 |

```
-- Menghapus Space
VARIABLE input_string VARCHAR2(20);
EXEC :input_string := 'Coba ';
SELECT REPLACE(:input_string, ' ', '') AS result
FROM dual;
```

Script Output

Output | Errors

| RESULT |
| --- |
| Coba |

# Question 24

- Fungsi untuk membesarkan semua huruf dalam suatu statement
  inputan : ‘abc’
  output : ‘ABC’
- Fungsi untuk membulatkan ke bawah inputan desimal
  inputan : 1.1
  output : 1

```
-- Uppercase String
VARIABLE input_string VARCHAR2(20);
EXEC :input_string := 'abc';
SELECT UPPER(:input_string) AS uppercase_result
FROM dual;
```

Script Output

Output | Errors | Grid 1 | Env

| UPPERCASE_RESULT |
| --- |
| ABC |

```
-- Membulatkan desimal ke bawah
VARIABLE input_number NUMBER;
EXEC :input_number := 1.1;
SELECT FLOOR(:input_number) AS rounded_down
FROM dual;
```

Script Output

Output | Errors | Grid 1 | Env

| ROUNDED_DOWN |
| --- |
| 1 |

# Question 25

- Fungsi untuk membuat statement berformat berikut :
  inputan : ABC
  output : Abc
- Fungsi untuk pembulatan ke atas untuk inputan number :
  inputan : 1.1
  output : 2

```
-- Uppercase to Capital First Word
VARIABLE input_string VARCHAR2(20);
EXEC :input_string := 'ABC AJJA';
SELECT INITCAP(:input_string) AS formatted_string
FROM dual;
```

Script Output

Output | Errors | Grid 1 | Enviro

| FORMATTED_STRING |
| --- |
| Abc Ajja |

```
-- Membulatkan Decimal ke Atas
VARIABLE input_number NUMBER;
EXEC :input_number := 1.1;
SELECT CEIL(:input_number) AS rounded_up
FROM dual;
```

Script Output

Output | Errors | Grid

| ROUNDED_UP |
| --- |
| 2 |

# Question 26

- Fungsi untuk memaksakan apapun inputan dari number selalu jadi positif
  Input : 200
  Output : 200
- Fungsi untuk membuat statement berformat berikut :
  inputan : ABC
  output : abc

```
-- Number Positif
VARIABLE input_number NUMBER;
EXEC :input_number := -200;
SELECT ABS(:input_number) AS positive_number
FROM dual;
```

Script Output

Output | Errors | Grid 1 | Envirc

| POSITIVE_NUMBER |
| --- |
| 200 |

```
-- Mengubah huruf kecil
VARIABLE input_string VARCHAR2(20);
EXEC :input_string := 'ABC';
SELECT LOWER(:input_string) AS formatted_string
FROM dual;
```

Script Output

Output | Errors | Grid 1 | Envirc

| FORMATTED_STRING |
| --- |
| abc |

# Question 27

- Fungsi untuk mengetahui jumlah bulan antara dua buah tanggal
  input : 20-JAN-2006 dan 20 –AUG-2006
  output : 7