

İlahə Vəliyeva SQL təkrar

Sual 1:

Departament üzrə üçüncü ən çox əmək haqqı alan və məlumatında hər hansı dəyişiklik edilmiş(job_history) əməkdaşların ad, soyad, departament nömrəsi(department_id), işlədiyi vəzifə(job_id), əmək haqqı, əmək haqqının dərəcəsinə(job_grades) departamentin adı, şəhərin adı, ölkə kodu, ölkə adı haqqında məlumatı ekrana çıxardan sorğunu yazın.

```
SELECT e.*,
       d.*
FROM employees e
JOIN departments d
ON e.department_id = d.department_id
WHERE e.job_history_change = 'YES'
AND e.salary = (
    SELECT salary
    FROM (
        SELECT e2.salary,
               ROW_NUMBER()
                   OVER (PARTITION BY e2.department_id
                        ORDER BY e2.salary DESC)
                   AS rn
        FROM employees e2
    ) AS ranked_salaries
    WHERE rn = 3
    AND department_id = e.department_id
);
```

Sual 2:

Aşağıdakı candidates cədvəlindən istifadə edərək hər qrup üçün ən yüksək əmək haqqını qaytaran sorğu yazın.

```
SELECT
    MAX(CASE WHEN POSITION = 'senior' THEN SALARY END)
    AS senior,
    MAX(CASE WHEN POSITION = 'junior' THEN SALARY END)
    AS junior
FROM candidates;
```

Sual 3:

Aşağıdakı candidates cədvəlindən istifadə edərək əmək haqqı gözləntisini göstərən sorğunu yazın. Əmək haqqı gözləntisini göstərən məlumatı ayrıca sütun şəklində göstərməlidir. Əmək haqqı gözləntisi 0 - 10000 arasında 10k, 10001 - 20000 arası 20k, 20001 - 30000 arası 30k, 30001 - 40000 arası 40k və 40000 dən yuxarı isə 50k olaraq qiymətləndirilsin.

```
SELECT
    ID,
    POSITION,
    SALARY,
    CASE
        WHEN SALARY BETWEEN 0 AND 10000 THEN '10k'
        WHEN SALARY BETWEEN 10001 AND 20000 THEN '20k'
        WHEN SALARY BETWEEN 20001 AND 30000 THEN '30k'
        WHEN SALARY BETWEEN 30001 AND 40000 THEN '40k'
        WHEN SALARY > 40000 THEN '50k'
    END AS salary_expectation
FROM candidates;
```

Sual 4:

Aşağıdakı candidates cədvəlindən istifadə edərək hər vəzifə üçün kümülatif cəm (cumulative sum) məlumatını ekrana çıxardan sorğunu yazın.

```
SELECT
    ID,
    POSITION,
```

```

        SALARY,
        SUM(SALARY) OVER
        (PARTITION BY POSITION ORDER BY ID)
        AS cumulative_salary
FROM candidates
ORDER BY POSITION, ID;

```

Sual 5:

Aşağıdaki natural join vasitəsilə yazılan sorğunun nəticəsi ilə eyni nəticəni əldə edən alternativ sorğunu yazın.

```

select emp.first_name,
emp.last_name,
his.end_date
from employees emp natural join job_history his;

```

```

SELECT emp.first_name,
       emp.last_name,
       his.end_date
FROM employees emp
INNER JOIN job_history his
ON emp.employee_id = his.employee_id;

```

Sual 6:

ADDRESS və CUST_INF cədvəllərini join edərək aşağıdakı məlumatları əldə edən sorğunu yazın.

```

SELECT
    ci.CUST_ID,
    MAX(CASE WHEN a.TYPE_ID = 1 THEN a.NAME END)
    AS NAME_QEYDIYYAT,
    MAX(CASE WHEN a.TYPE_ID = 2 THEN a.NAME END)
    AS NAME_FAKTIKI,
    MAX(CASE WHEN a.TYPE_ID = 3 THEN a.NAME END)
    AS NAME_DAIMI
FROM

```

```
CUST_INF ci
JOIN
ADDRESS a ON ci.CUST_ID = a.CUST_ID
GROUP BY
ci.CUST_ID;
```

Sual 7:

Student və Record cədvəlləri vasitəsilə qeyd olunan birləşmələrin nəticərini yazın.

Inner join

```
SELECT s.name, r.id, r.class, r.city
FROM Student s
INNER JOIN Record r
ON s.class = r.class AND s.city = r.city;
```

Left join

```
SELECT s.name, r.id, r.class, r.city
FROM Student s
LEFT JOIN Record r
ON s.class = r.class AND s.city = r.city;
```

Right Join

```
SELECT s.name, r.id, r.class, r.city
FROM Student s
RIGHT JOIN Record r
ON s.class = r.class AND s.city = r.city;
```

Full outer join

```
SELECT s.name, r.id, r.class, r.city
FROM Student s
```

```
FULL OUTER JOIN Record r
ON s.class = r.class AND s.city = r.city;
```

Non-Equi Join

```
SELECT s.name, r.id, r.class, r.city
FROM Student s
JOIN Record r
ON s.class <> r.class;
```

Sual 8:

1. Sifarişlərin sayı və hər bir müştərinin ən son sifariş tarixi:
2. Hər bir kateqoriyanın üzərindəki məhsulların sayı:

```
SELECT c.Name AS Customer_Name,
       COUNT(o.ID) AS Total_Orders,
       MAX(o.Order_Date) AS Last_Order_Date
FROM Customers c
LEFT JOIN Orders o ON c.ID = o.Customer_ID
GROUP BY c.Name;
```

Sual 9:

Əməkdaşların əmək haqqlarının 20 % -nin üzərinə 500 əlavə etdikdə departamentlər üzrə orta əmək haqqıya bərabər olan(yuvarlaqlaşdırma tam hissəyə görə olmalıdır) məlumatları təyin edən sorğunu yazın.

```
SELECT e.department_id,
       e.first_name,
       e.last_name,
       ROUND(e.salary * 0.20 + 500, 0)
       AS adjusted_salary,
       ROUND(AVG(e.salary) OVER
       (PARTITION BY e.department_id), 0)
```

```

        AS avg_salary
FROM employees e
WHERE ROUND(e.salary * 0.20 + 500, 0) = ROUND(AVG(e.salary)
OVER (PARTITION BY e.department_id), 0);

```

Sual 10:

10.03.2002 və 25.03.2023 -cü il tarix intervallarında həftənin bazar günlərinin düşdüyü tarixi təyin edən sorğunu yazın.

```

WITH sundays AS (
    SELECT NEXT_DAY(TO_DATE('10.03.2002', 'DD.MM.YYYY') - 1,
    'SUNDAY') AS sunday_date
    FROM dual
    UNION ALL
    SELECT sunday_date + 7
    FROM sundays
    WHERE sunday_date + 7 <=
    TO_DATE('25.03.2023', 'DD.MM.YYYY')
)
SELECT sunday_date
FROM sundays
ORDER BY sunday_date;

```

Sual 11:

Tranzaksiya cədvəlindəki məlumatlara əsasən hər sətirin yanında gün ərzində müştərinin etdiyi əməliyyatların sayı və gün ərzində etdiyi tranzaksiyalar üzrə məbləğlərin cəmini göstərən sorğunu yazın.

```

SELECT t.Transaction_id,
       t.Amount,
       t.Dates,
       t.Customer_id,
       COUNT(t.Transaction_id)
       OVER (PARTITION BY t.Customer_id, t.Dates)
       AS daily_transaction_count,
       SUM(t.Amount) OVER (PARTITION BY t.Customer_id, t.Date

```

```
AS daily_total_amount
FROM Transactions t;
```

Sual 12:

Departament üzrə ümumi əmək haqqı Employees cədvəlində salary sütunu üzrə özündən 3 əvvəlki özü və özündən 3 sonrakının cəminə bərabər olan məlumatları ekrana çıxardan sorğunu yazın. Ekrana departament nömrəsi və ümumi cəm haqda məlumatlar çıxsın.

```
SELECT department_id, SUM(salary) AS total_salary
FROM (
    SELECT
        e.*,
        LAG(salary, 3)
        OVER (PARTITION BY department_id ORDER BY employee_id
              AS lag_3,
        LEAD(salary, 3)
        OVER (PARTITION BY department_id ORDER BY employee_id
              AS lead_3
    FROM employees e
)
WHERE salary = NVL(lag_3, 0) + NVL(lead_3, 0)
GROUP BY department_id;
```

Sual 13:

Dünya Azərbaycanlılarının Həmrəylik Günü cümləsindən hər bir sözü ayrıca sütun şəklində ekrana çıxardan sorğu yazın(dörd sütun olacaq sorğuda).

```
WITH text_data AS (
    SELECT 'Dünya Azərbaycanlılarının Həmrəylik Günü'
    AS text
    FROM dual
)
SELECT
    REGEXP_SUBSTR(text, '^[^\s]+', 1, 1) AS "Söz1",
```

```

    REGEXP_SUBSTR(text, '[^\s]+', 1, 2) AS "Söz2",
    REGEXP_SUBSTR(text, '[^\s]+', 1, 3) AS "Söz3",
    REGEXP_SUBSTR(text, '[^\s]+', 1, 4) AS "Söz4"
FROM text_data;

```

Sual 14:

12Oracle18 sözündən əvvəldə olan 12 və sonda olan 18 ədədlərini ekrana çıxardan sorğunu yazın.

```

SELECT
    REGEXP_SUBSTR('12Oracle18', '^\d+') AS first_number,
    REGEXP_SUBSTR('12Oracle18', '\d+$') AS last_number
FROM dual;

```

Sual 15:

D1-A-JOIN then the number 5 then another number like 7 cümləsinin içərisindən ancaq rəqəmləri ekrana çıxardan sorğunu yazın.

```

WITH text_data AS (
    SELECT 'D1-A-JOIN then the number 5 then another number 1'
    AS text
    FROM dual
)
SELECT
    REGEXP_SUBSTR(text, '\d+', 1, 1) AS "Rəqəm1",
    REGEXP_SUBSTR(text, '\d+', 1, 2) AS "Rəqəm2",
    REGEXP_SUBSTR(text, '\d+', 1, 3) AS "Rəqəm3"
FROM text_data;

```