

i2i Academy

Training Document

Topic	Oracle SQL Language Fundamentals – Analytic Functions
Document Name	SQLAN01-EX-02

Document Difficulty Level			
Beginner	Junior	Senior	Expert
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Copyright of  i2i Systems Turkey 2025

Document History

Date	Author	Ver	Comments
01.10.2017	Aslan Yüksel Mennan Tekbir	1.0	Initial Draft
11.07.2025	Mennan Tekbir	1.1	Revisions

Oracle SQL Language Fundamentals – Analytic Functions

Exercise SQLAN-EX-02:

Aşağıdaki soruları, analitik fonksiyonları kullanarak SQL dilinde yazınız.

HR şemasını kullanabilirsiniz.

Soru 1 : Bir departman'da çalışan tüm kişilerin isimlerini tek kolonda yan yana olacak şekilde yazınız.

```
SELECT department_id,  
       LISTAGG(first_name || ' ' || last_name, ', ' )  
         WITHIN GROUP (ORDER BY hire_date) AS employees_list  
FROM employees  
GROUP BY department_id;
```

Soru 2 : Job ID'ye göre gruplanacak şekilde Employee ID sıralaması baz alınarak, her çalışan için kendisinden 1 önce ve sonra çalışanların maaşlarının toplamını yazınız.

```
SELECT employee_id, first_name, last_name, job_id, salary,  
       NVL(LAG(salary, 1, 0) OVER (PARTITION BY job_id ORDER BY employee_id), 0) +  
       NVL(LEAD(salary, 1, 0) OVER (PARTITION BY job_id ORDER BY employee_id), 0) AS adjacent_sal_sum  
FROM employees;
```

Soru 3 : Job ID'ye göre gruplanacak şekilde Employee ID sıralaması baz alınarak, her çalışan için kendinden 1 sonra gelen çalışanın telefon numarasını yazınız.

```
SELECT employee_id, first_name, last_name, job_id, phone_number,  
       LEAD(phone_number, 1, 'N/A') OVER (PARTITION BY job_id ORDER BY employee_id) AS next_emp_phone  
FROM employees;
```

Soru 4 : Tüm çalışanların, şirketteki maaş sıralamasını yazınız. Aynı maaşa ait birden fazla kişi varsa işe giriş tarihini de sıralamaya ekleyiniz

```
SELECT employee_id, first_name, last_name, salary, hire_date,  
       DENSE_RANK() OVER (ORDER BY salary DESC, hire_date ASC) AS company_salary_rank  
FROM employees;
```

Soru 5 : Tüm çalışanları, Employee ID sıralamasına göre 10’arlı gruplara ayırınız.

```
SELECT employee_id, first_name, last_name,  
       NTILE(10) OVER (ORDER BY employee_id) AS employee_decile  
FROM employees;
```

Soru 6 : Her departman için, o departmanın ortalama maaşın altında olan çalışanları 0; üstünde olanları ise 1 ile belirtiniz.

```
SELECT employee_id, first_name, last_name, department_id, salary,  
       CASE  
         WHEN salary >= AVG(salary) OVER (PARTITION BY department_id) THEN 1  
         ELSE 0  
       END AS above_avg_flag  
FROM employees;
```

Soru 7 : Her yıl için, işe ilk alınan çalışanı bulunuz.

```
SELECT hire_year, employee_id, first_name, last_name, hire_date  
FROM (  
  SELECT EXTRACT(YEAR FROM hire_date) AS hire_year,  
         employee_id, first_name, last_name, hire_date,  
         ROW_NUMBER() OVER (  
           PARTITION BY EXTRACT(YEAR FROM hire_date)  
           ORDER BY hire_date, employee_id  
         ) AS hire_order  
  FROM employees  
)  
WHERE hire_order = 1;
```

Soru 8 : Bir departman’daki en yüksek maaş alan çalışan dışında kalanları listeleyiniz.

```
SELECT employee_id, first_name, last_name, department_id, salary  
FROM (  
  SELECT e.*,  
         DENSE_RANK() OVER (  
           PARTITION BY department_id  
           ORDER BY salary DESC  
         ) AS salary_rank  
  FROM employees e  
)  
WHERE salary_rank > 1  
   OR department_id IS NULL;
```

Soru 9 : Bir departman içinde en yüksek maaş alan 2 çalışanını listeleyiniz.

```
SELECT department_id, employee_id, first_name, last_name, salary
FROM (
    SELECT e.*,
           DENSE_RANK() OVER (
               PARTITION BY department_id
               ORDER BY salary DESC
           ) AS salary_rank
    FROM employees e
)
WHERE salary_rank <= 2;
```

Soru 10 : Bir departman'da bulunan tüm çalışanlar için, kendisinden önce ve sonra işe başlayan çalışanların isim ve soyisim bilgilerini getiriniz.

```
SELECT employee_id, first_name, last_name, department_id, hire_date,
       LAG(first_name || ' ' || last_name, 1, 'N/A')
       OVER (PARTITION BY department_id ORDER BY hire_date) AS prev_employee,
       LEAD(first_name || ' ' || last_name, 1, 'N/A')
       OVER (PARTITION BY department_id ORDER BY hire_date) AS next_employee
FROM employees;
```

Objectives : To learn oracle analytic functions.

Exercise Keywords: LEAD, LAG, ROW_NUMBER, RANK, DENSE_RANK, MAX, MIN, SUM, NTILE

Solution of SQLAN-EX-02:

Tüm soruları cevaplayınız. Telefon numaranızın son hanesinin + 1 nolu sorunun cevabını LinkedIn'e post olarak ekleyiniz. Örneğin telefon numaranız 5339635384 ise son hane 4 olur. (4 +1) 5. soruda kullandığınız fonksiyon(lar)ın açıklamasını içeren bir post yayınlayınız. Eğer LEAD fonksiyonunu kullandıysanız, LEAD ne demek ne amaçla kullanılır vs. yazdıktan sonra soru ve cevabı Hand-on-Lab şeklinde post ediniz.