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

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Difficulty Level** | | | |
| **Beginner** | **Junior** | **Senior** | **Expert** |
| □ | ■ | □ | □ |

# 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.

**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.

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

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

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

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

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

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

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

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

**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.

**Job ID’ye göre gruplanmış şekilde** ve **Employee ID sıralamasına göre**, **bir önceki ve bir sonraki çalışanın maaşlarının toplamını** hesaplayan SQL sorgusu yer almaktadır.

SELECT

employee\_id,

job\_id,

salary,

NVL(LAG(salary) OVER (PARTITION BY job\_id ORDER BY employee\_id), 0) +

NVL(LEAD(salary) OVER (PARTITION BY job\_id ORDER BY employee\_id), 0)

AS previous\_next\_salary\_sum

FROM

employees;

**Açıklama:**

* PARTITION BY job\_id: Her **iş tanımı (JOB\_ID)** için ayrı ayrı gruplanır.
* ORDER BY employee\_id: Her grup içinde **employee\_id'ye göre sıralama** yapılır.
* LAG(salary): Kendinden **bir önceki çalışanın maaşını** alır.
* LEAD(salary): Kendinden **bir sonraki çalışanın maaşını** alır.
* NVL(..., 0): Eğer önceki ya da sonraki çalışan yoksa (örneğin ilk veya son satır), **NULL yerine 0** alınır.
* previous\_next\_salary\_sum: Bu iki değerin toplamını gösteren sütundur.

| **employee\_id** | **job\_id** | **salary** |
| --- | --- | --- |
| 101 | IT\_PROG | 6000 |
| 102 | IT\_PROG | 7000 |
| 103 | IT\_PROG | 4000 |
| 104 | HR\_REP | 3000 |
| 105 | HR\_REP | 3500 |

| **employee\_id** | **job\_id** | **salary** | **prev\_salary** | **next\_salary** | **toplam** |
| --- | --- | --- | --- | --- | --- |
| 101 | IT\_PROG | 6000 | NULL | 7000 | 0+7000 = 7000 |
| 102 | IT\_PROG | 7000 | 6000 | 4000 | 6000+4000 = 10000 |
| 103 | IT\_PROG | 4000 | 7000 | NULL | 7000+0 = 7000 |

Sonuç: Her çalışanın "kendinden önce ve sonra gelen" aynı job\_id'li çalışanların maaş toplamı bulunur.