Objectives

1.SQL sorgularında **ROUND**, **TRUNC** ve **MOD** gibi tek satırlık (single-row) sayı fonksiyonlarını seçmek ve uygulamak.

Cevap: SELECT ROUND(45.926, 2) FROM DUAL; (Sayı, virgülden sonra 2 basamak olacak şekilde yuvarlanır. Çıktı: 45.93) (YUVARLAMA)

SELECT TRUNC(45.926, 2) FROM DUAL; (Sayı, virgülden sonra 2 basamak alınarak kesilir, yuvarlama yapılmaz. çıktı: 45.92 ) ( KESME)

SELECT MOD(19, 4) FROM DUAL; (19'u 4'e böldüğümüzde kalan 3’tür. Çıktı: 3) (KALAN)

2. **TRUNC** ve **ROUND** fonksiyonları bir sayısal değere uygulandığında elde edilen sonuçlar arasındaki farkı ayırt etmek.

Cevap:

| **Fonksiyon** | **Açıklama** | **Örnek** | **Sonuç** |
| --- | --- | --- | --- |
| ROUND | Sayıyı **en yakın değere yuvarlar**. | ROUND(123.456, 2) | 123.46 |
| TRUNC | Sayının **istenen basamaktan sonrasını keser**. | TRUNC(123.456, 2) | 123.45 |

3. Sayısal değerlere **TRUNC** ve **ROUND** uygulanmasının iş dünyası açısından ne gibi sonuçlar doğurabileceğini açıklamak.

Cevap:

* **ROUND** kullanmak, sayıları en yakın değere yuvarlar. Örneğin, finansal raporlarda **net kar** gibi hesaplamalarda kullanılırsa ufak farklar yaratabilir.
* **TRUNC** sayının belirli bir basamakta kesilmesini sağlar. Örneğin, vergi hesaplamalarında kesirli kısımlar dikkate alınmıyorsa **TRUNC** tercih edilir.
* Yanlış seçim, faturalarda ya da ödeme sistemlerinde hatalara neden olabilir.

Vocabulary

* 1. Sütun, ifade veya değeri belirtilen ondalık basamak sayısına **kesmek** için kullanılır:TRUNC
  2. Bu fonksiyonlar **sayısal veri alır** ve sayısal değerler döndürür: NUMBER FUNCTİON
  3. Bir bölme işleminin **kalanını döndürür**: MOD
  4. Sütun, ifade veya değeri belirli bir ondalık basamağa **yuvarlar**: ROUND

Try It / Solve It

1.Oracle veritabanındaki **employee\_id**’si 100 ile 102 arasında olan çalışanların **last\_name** ve **salary** bilgilerini görüntüleyin. Üçüncü bir sütunda, her maaşı 1.55’e bölün ve sonucu iki ondalık basamağa yuvarlayın.

Cevap: SELECT last\_name, salary, ROUND(salary / 1.55, 2) AS rounded\_salary

FROM employees

WHERE employee\_id BETWEEN 100 AND 102;

2. Departman numarası **80** olan çalışanların **last\_name** ve **salary** bilgilerini görüntüleyin. Her birine %5.333 oranında zam yapın ve sonucu iki ondalık basamağa **kesin** (trunc).

Cevap: SELECT last\_name, salary, TRUNC(salary \* 1.05333, 2) AS increased\_salary

FROM employees

WHERE department\_id = 80;

3. **MOD** fonksiyonunu kullanarak 38873 sayısının **tek** mi **çift** mi olduğunu belirleyin.

Cevap: SELECT MOD(38873, 2) AS result

FROM dual; ( Eğer sonuç **1** ise sayı **tek**, **0** ise **çift**tir.Cevap: 1 → **tek sayı)**

4. **DUAL** tablosunu kullanarak aşağıdaki sayılar üzerinde işlem yapın:

* 845.553 → bir ondalık basamağa yuvarlayın
* 30695.348 → iki ondalık basamağa yuvarlayın
* 30695.348 → -2 basamağa yuvarlayın (yüzlük basamaklara)
* 2.3454 → ondalık kısmın 454'ünü kesin (truncate)

Cevap:

-- a)

SELECT ROUND(845.553, 1) AS result1 FROM dual;

-- b)

SELECT ROUND(30695.348, 2) AS result2 FROM dual;

-- c)

SELECT ROUND(30695.348, -2) AS result3 FROM dual;

-- d)

SELECT TRUNC(2.3454, 1) AS result4 FROM dual;

5. Her çalışanın maaşını 3'e bölün. **Maaşı 3’ün katı olan** çalışanların sadece **last\_name** ve **salary** bilgilerini gösterin.

Cevap: SELECT last\_name, salary

FROM employees

WHERE MOD(salary, 3) = 0;

6. **34’ü 8’e bölün**, sadece bölümden kalan **kalanı (remainder)** gösterin. Sonucu **EXAMPLE** adında gösterin.

Cevap: SELECT MOD(34, 8) AS EXAMPLE

FROM dual; (sonuc: 2)

7. Maaşınızı nasıl istersiniz – **yuvarlanmış mı** yoksa **kesilmiş mi**? Diyelim ki haftalık maaşınız 565.784$ hesaplandı, ama size 565.78$ ödendi. 0.004 cent küçük görünebilir. Ama bu işlem 1.000 kişi, 100.000 kişi, hatta 1 milyon kişi için yapıldığında fark eder mi? Ne kadar fark eder?

Ceevap: Eğer maaşlar **trunc** edilirse, şirket her kişiden 0.004$ tasarruf eder.  
Bu;

* 1.000 kişi için: 0.004 \* 1000 = 4$
* 100.000 kişi için: 0.004 \* 100000 = 400$
* 1 milyon kişi için: 0.004 \* 1000000 = 4.000$ tasarruf sağlar.

Yani küçük bir miktar gibi görünse de **büyük çaplı çalışan sayılarında ciddi bir fark** yaratabilir. Bu nedenle **round vs trunc** tercihi, finansal uygulamalarda çok önemlidir.