Vocabulary

| **Tanım** | **Cevap** |
| --- | --- |
| Her iki sorgudan gelen tüm satırları döndürür, ancak yinelenenleri kaldırır | UNION |
| Diğer tabloyla eşleşmesi için oluşturulmuş ancak iki tabloda da olmayan sütunlar | Dummy Columns veya Derived Columns |
| Her iki sorgudan gelen tüm satırları döndürür, yinelenenleri de içerir | UNION ALL |
| Birden fazla SELECT sorgusunun sonuçlarını tek bir sonuç olarak birleştirir | Set Operator |
| Her tabloya özgü (eşsiz) satırları döndüren operatör | MINUS |
| Her iki tabloda da bulunan ortak satırları döndüren operatör | INTERSECT |

Try It / Solve It

**1. SORU**

Farklı küme (SET) operatörlerinin isimlerini yaz.

**ÇÖZÜM:**

1. UNION – Yinelenenleri kaldırır, birleşik sonuç
2. UNION ALL – Yinelenenler dahil birleşik sonuç
3. INTERSECT – İki sorgunun ortak (kesişim) satırları
4. MINUS – İlk sorguda olup ikinci sorguda olmayanlar

**2. SORU**

İki sorgu yaz:

* İlki: employees tablosundan employee\_id, job\_id, hire\_date, department\_id
* İkincisi: job\_history tablosundan employee\_id, job\_id, start\_date, department\_id  
  Bunları küme operatörü ile birleştir ve yinelenenleri kaldır.

**ÇÖZÜM:**

SELECT employee\_id, job\_id, hire\_date, department\_id

FROM employees

UNION

SELECT employee\_id, job\_id, start\_date, department\_id

FROM job\_history;

📌 **Not:** UNION yinelenenleri kaldırır.

**3. SORU**

Yukarıdaki sorguyu, yinelenenleri kaldırmadan (UNION ALL) yaz ve sonucu incele. Kaç ekstra satır geldi ve hangileriydi? Sonuçları employee\_id'ye göre sırala.

**ÇÖZÜM:**

SELECT employee\_id, job\_id, hire\_date, department\_id

FROM employees

UNION ALL

SELECT employee\_id, job\_id, start\_date, department\_id

FROM job\_history

ORDER BY employee\_id;

Farkı görmek için UNION ile UNION ALL sonucu karşılaştır. Ekstra satırlar, bir çalışanın geçmiş iş bilgisi ile şu anki işi aynıysa tekrar olarak görünür.

**4. SORU**

Hiçbir zaman iş değiştirmemiş çalışanları listele. (Bu kişiler job\_history tablosunda yoktur.)

**ÇÖZÜM:**

SELECT employee\_id, last\_name

FROM employees

MINUS

SELECT employee\_id, last\_name

FROM employees

JOIN job\_history USING (employee\_id);

**5. SORU**

İş değiştirmiş çalışanları listele (en az bir defa).

**ÇÖZÜM:**

SELECT employee\_id, last\_name

FROM employees

WHERE employee\_id IN (

SELECT employee\_id FROM job\_history

);

**6. SORU**

UNION kullanarak hem şu anki hem geçmiş çalışanların employee\_id, job\_id ve salary bilgilerini gösteren bir sorgu yaz.  
Eğer maaş bilgisi yoksa 0 gösterilsin.

**ÇÖZÜM:**

SELECT employee\_id, job\_id, salary

FROM employees

UNION

SELECT employee\_id, job\_id, NVL(salary, 0)

FROM job\_history;

Not: job\_history tablosunda salary sütunu yoksa, ek sütun eklenmelidir:

SELECT employee\_id, job\_id, salary

FROM employees

UNION

SELECT employee\_id, job\_id, 0 AS salary

FROM job\_history;