**VERİ TABANI - 1**

Bahar Dönemi

1. **TABLOLARIN BİRLEŞTİRİLMESİ**

Ortak alanları olan tablolar birleştirilerek sorgulanırlar. Buna **JOIN** işlemleri denilir.

Hangi tablodan hangi sütunun kullanılacağının belirlenmesi için kullanılacak ifade:

**tablo\_adi.alan\_adi**

JOIN, iki tabloyu birbiriyle ilişkilendirmek için kullanılan komuttur. İki veya daha fazla yapıdan tek bir tablosal yapı üretir. 3 temel Join işlemi vardır.

1. INNER JOIN
2. OUTER JOIN
3. LEFT OUTER JOIN
4. RIGHT OUTER JOIN
5. FULL OUTER JOIN
6. CROSS JOIN

|  |  |  |
| --- | --- | --- |
| A tablosu: |  | B Tablosu: |
|  |  |  |
| a |  | b |
| 102 |  | 101 |
| 104 |  | 102 |
| 106 |  | 104 |
| 107 |  | 106 |
|  |  | 108 |
|  |  |  |

* 1. **INNER JOIN**

tablo\_adi1 INNER JOIN tablo\_adi2

ON alan\_adi\_1 = alan\_adi\_2

INNER JOIN yapısında sol tarafa yazılan tablo listelenirken sol tarafa yazılır. Yapının sağ tarafına yazılan tablo ise listelenirken sağ tarafa yazılarak listelenir.

INNER JOIN sadece ON cümleciğindeki koşulu sağlayan satırları listeler.

|  |  |
| --- | --- |
| a | b |
| 102 | 102 |
| 104 | 104 |
| 106 | 106 |

SELECT A.a, B.b

FROM A INNER JOIN B

ON a = b;

* 1. **OUTER JOIN**

ON cümleciğindeki şartı sağlayanlarla birlikte şartı sağlamayan verileri de listeler.

* + 1. **LEFT OUTER JOIN**

tablo\_adi1 LEFT OUTER JOIN tablo\_adi2

ON alan\_adi\_1 = alan\_adi\_2

Sol tarafa yazılan tablo ile sağ tarafa yazılan tabloda eşleşme olup olmadığına bakılmaksızın sol taraftaki tablo tamamen yazılır. Şartı sağlayan alanlar soldaki sütunların yanına eklenir.

|  |  |
| --- | --- |
| a | b |
| 102 | 102 |
| 104 | 104 |
| 106 | 106 |
| 107 |  |

SELECT A.a, B.b

FROM A LEFT OUTER JOIN B

ON a = b;

* + 1. **RIGHT OUTER JOIN**

tablo\_adi1 RIGHT OUTER JOIN tablo\_adi2

ON alan\_adi\_1 = alan\_adi\_2

Sol tarafa yazılan tablo ile sağ tarafa yazılan tabloda eşleşme olup olmadığına bakılmaksızın sağ taraftaki tablo tamamen yazılır. Şartı sağlayan alanlar sağdaki sütunların yanına eklenir.

|  |  |
| --- | --- |
| a | b |
|  | 101 |
| 102 | 102 |
| 104 | 104 |
| 106 | 106 |
|  | 108 |

SELECT A.a, B.b

FROM A RIGHT OUTER JOIN B

ON a = b;

* + 1. **FULL OUTER JOIN**

tablo\_adi1 FULL OUTER JOIN tablo\_adi2

ON alan\_adi\_1 = alan\_adi\_2

ON cümleciğindeki şartı sağlayan kayıtlar listelenir. Daha sonra şartı sağlamayanlar listelenir.

|  |  |
| --- | --- |
| a | b |
| 102 | 102 |
| 104 | 104 |
| 106 | 106 |
| 107 |  |
|  | 101 |
|  | 108 |

SELECT A.a, B.b

FROM A FULL OUTER JOIN B

ON a = b;

* 1. **CROSS JOIN**

tablo\_adi\_1 CROSS JOIN tablo\_adi\_2

ON yapısı kullanılmadan iki tablodaki bütün satırlar listelenir. Eşleşme kontrolü yapmaz. Listelenen tablo boyutu iki tablonun satır sayılarının çarpımı kadardır.

SELECT A.a, B.b

FROM A CROSS JOIN B;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b |  | a devam | b devam |
| 102 | 101 |  | 106 | 101 |
| 102 | 102 |  | 106 | 102 |
| 102 | 104 |  | 106 | 104 |
| 102 | 106 |  | 106 | 106 |
| 102 | 108 |  | 106 | 108 |
| 104 | 101 |  | 107 | 101 |
| 104 | 102 |  | 107 | 102 |
| 104 | 104 |  | 107 | 104 |
| 104 | 106 |  | 107 | 106 |
| 104 | 108 |  | 107 | 108 |

1. **İÇ İÇE SQL İFADELERİ**

Birden fazla SQL cümlesi kullanılması gereken durumlarda iç içe SQL sorguları kullanılır.

* 1. **OPERATORLER**

|  |  |
| --- | --- |
| Operatör | Anlamı |
| = | Eşit |
| < > | Eşit değil |
| != | Eşit değil |
| > | Büyüktür |
| >= | Büyük eşit |
| < | Küçüktür |
| <= | Küçük eşit |
| ! | Değil |

* 1. **IN**

Bir veri kümesi içerisinde arama yapmamızı, veri kümesi içerisinden seçim yapmamızı sağlayan, iç içe yazılan sql sorgularından dönen değerler arasından seçim yapmamızı sağlayan bir komuttur.

SELECT alanad(lar)i FROM tablo\_adi

WHERE koşul IN [koşul | SQL sorgusu]

SELECT \* FROM ogr\_tbl

WHERE bolum IN (1,2,3);

SELECT \* FROM

WHERE okul IN (‘ön lisans’, ’lisans’, ‘yüksek lisans’);

SELECT \* FROM veriler

WHERE veri\_id

IN (SELECT personel\_id FROM personel\_tbl);

* 1. **UNION**

Birden fazla select sorgusunu birleştirir. Bağımsız sorguların birleştirilmesini sağlar.

SELECT alanadi1 FROM tablo\_adi1

UNION

SELECT alanadi1 FROM tablo\_adi2;

* 1. **ANY & SOME**

IN yapısına benzer. Herhangi bir değeri alt sorgu ile karşılaştırır. İki komut da aynı sorguyu döndürür.

SELECT alan\_ad(lar)i

FROM tablo\_adi

WHERE sutun\_adi operatör ANY ( SELECT alan\_adi2

FROM tablo\_adi2

WHERE koşul);

SELECT alan\_ad(lar)i

FROM tablo\_adi

WHERE sutun\_adi operatör SOME ( SELECT alan\_adi2

FROM tablo\_adi2

WHERE koşul);

* 1. **ALL**

Alt sorgu ifadesinin verilen tüm şartları sağlayıp sağlamadığını kontrol eder.

SELECT alan\_ad(lar)i

FROM tablo\_adi

WHERE alan\_adi operatör ALL (SELECT alan\_adi2

FROM tablo\_adi2

WHERE koşul);

* 1. **EXIST & NOT EXIST**

Sorgunun herhangi bir değer üretip üretmediğini gösterir.

SELECT alan\_ad(lar)i

FROM tablo\_adi

WHERE EXIST (SELECT alan\_adi2

FROM tablo\_adi2

WHERE koşul);

SELECT alan\_ad(lar)i

FROM tablo\_adi

WHERE NOT EXIST (SELECT alan\_adi2

FROM tablo\_adi2

WHERE koşul);

**Kaynakça:**

* Özdemir S., “C# .Net 2008 ve ASP.NET”, Nirvana Yayınları, Ankara, 2008.
* Obalı M., “Oracle 10g”, Pusula Yayıncılık, İstanbul, 2007.
* Kaya Y. ve Tekin R., “Veritabanı ve Uygulamaları”, Papatya Yayıncılık, İstanbul, 2007.

Öğr. Gör. Kevser ONUR