

BÖLÜM- 5: BİRDEN FAZLA TABLODAN VERİ GETİRMEK

- Sorgularda birden fazla tablo kullanımı
- Tabloların birleştirilmesi
- Birleştirme türleri
- Eşiti olan birleştirme (Join)
- Eşiti olmayan birleştirme
- Left Outer Join
- Right Outer Join
- Küme (SET) operatörleriyle birleştirme
- Alt sorgular Nasıl oluşturulur?
- Çoklu satır alt sorguların yazılması
- FROM sözcüğü içinde alt sorgu tanımlama

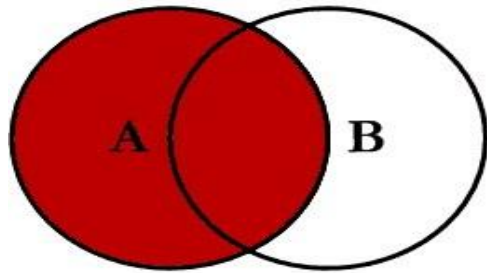
TABLoları BİRLEŞTİRME - JOIN

- Birden fazla tablo birleştirilirken ALIAS kullanılmasına dikkat edilmelidir. Eğer ALIAS kullanılmaz ise yanlış sonuçlar üretilebilir.
- Sütun belirsizlikleri yine ALIAS kullanımı ile giderilmelidir.
- Tablolar arasındaki ilişki iyi belirlenmelidir.
- Gereksiz sütunların gelmesi engellenmeli, gerekmediği sürece SELECT ifadesinde * kullanılmamalıdır.

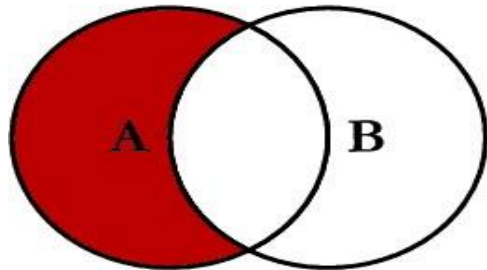


JOIN TÜRLERİ

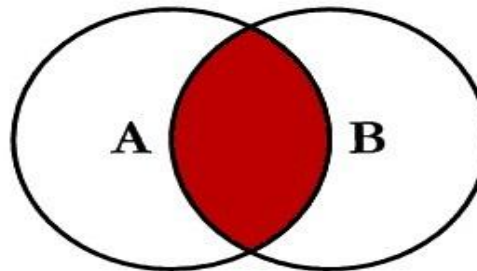
SQL JOINS



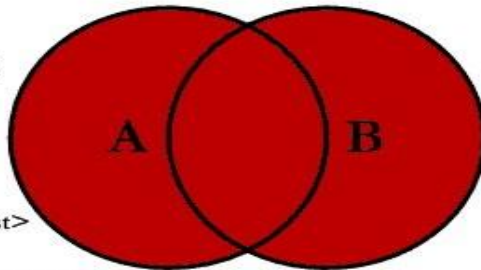
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



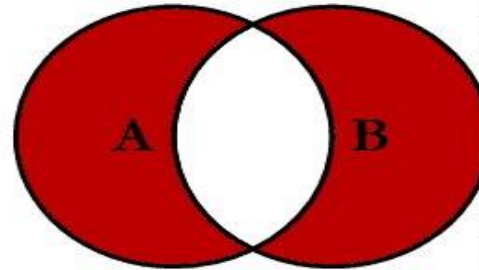
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



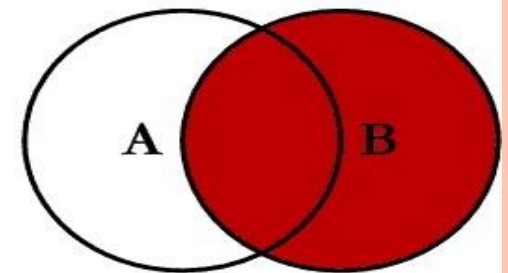
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



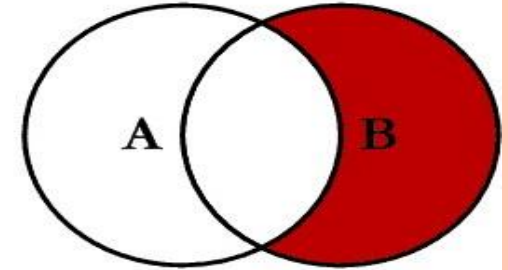
```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```

EŞİTİ OLAN BİRLEŞTİRME- INNER JOIN

- Müdürlüklerin bulunduğu şehirleri listeleyelim
- `SELECT department_id, department_name, city`
`FROM hr.departments d, hr.locations l where d.location_id=l.location_id;`

DEPARTMENT_ID	DEPARTMENT_NAME	CITY
60	IT	Southlake
50	Shipping	South San Francisco
10	Administration	Seattle
30	Purchasing	Seattle
90	Executive	Seattle
100	Finance	Seattle
110	Accounting	Seattle
120	Treasury	Seattle

2: 1 Row 1 of 27 total rows MSDEVECI@TESTDB

EŞİTİ OLAN BİRLEŞTİRME- INNER JOIN

- Müdürlüklerin bulunduğu şehirleri listeleyelim
- `SELECT department_id, department_name, city`
`FROM hr.departments d inner join hr.locations l on d.location_id=l.location_id;`
- `SELECT department_id, department_name, city`
`FROM hr.departments d join hr.locations l using(location_id);`

DEPARTMENT_ID	DEPARTMENT_NAME	CITY
60	IT	Southlake
50	Shipping	South San Francisco
10	Administration	Seattle
30	Purchasing	Seattle
90	Executive	Seattle
100	Finance	Seattle
110	Accounting	Seattle
120	Treasury	Seattle

2: 1 Row 1 of 27 total rows MSDEVECI@TESTDB

EŞİTİ OLMAYAN BİRLEŞTİRME

- Personellerin id,adı,soyadı,şehirlerini ve bölüm adlarını listeleyelim
- ```
SELECT employee_id, first_name, last_name, city, department_name
FROM hr.employees e
JOIN hr.departments d
ON d.department_id = e.department_id
JOIN hr.locations l
ON d.location_id = l.location_id;
```

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | CITY                | DEPARTMENT_NAME  |
|-------------|------------|-----------|---------------------|------------------|
| 198         | Donald     | OConnell  | South San Francisco | Shipping         |
| 199         | Douglas    | Grant     | South San Francisco | Shipping         |
| 200         | Jennifer   | Whalen    | Seattle             | Administration   |
| 201         | Michael    | Hartstein | Toronto             | Marketing        |
| 202         | Pat        | Fay       | Toronto             | Marketing        |
| 203         | Susan      | Mavris    | London              | Human Resources  |
| 204         | Hermann    | Baer      | Munich              | Public Relations |
| 205         | Shelley    | Higgins   | Seattle             | Accounting       |

8: 1 Row 1 of 106 total rows MSDEVECI@TESTDB Modified

# EŞİTİ OLMAYAN BİRLEŞTİRME

- Personellerin maximum ve minimum maaşına göre pozisyon tanımlarını listeleyelim
- ```
SELECT e.first_name,e.last_name, e.salary, j.job_title
FROM hr.employees e JOIN hr.jobs j
ON e.salary
BETWEEN j.min_salary AND j.max_salary;
```

FIRST_NAME	LAST_NAME	SALARY	JOB_TITLE
TJ	Olson	2100	Stock Clerk
Hazel	Philtanker	2200	Stock Clerk
Steven	Markle	2200	Stock Clerk
James	Landry	2400	Stock Clerk
Ki	Gee	2400	Stock Clerk
Randall	Perkins	2500	Stock Clerk
Karen	Colmenares	2500	Stock Clerk
Joshua	Patel	2500	Stock Clerk

18: 17 | Row 1 of 500 fetched so far (more rows exist) | MSDEVECI@TESTDB

LEFT OUTER JOIN

- Personellerin departman adlarını listeleyelim
- `SELECT first_name, e.last_name, e.department_id, d.department_name
FROM hr.employees e LEFT OUTER JOIN hr.departments d
ON (e.department_id = d.department_id) ;`

FIRST_NAME	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Jennifer	Whalen	10	Administration
Pat	Fay	20	Marketing
Michael	Hartstein	20	Marketing
Karen	Colmenares	30	Purchasing
Guy	Himuro	30	Purchasing
Sigal	Tobias	30	Purchasing
Shelli	Baida	30	Purchasing
Alexander	Khoo	30	Purchasing

23: 1 | Row 1 of 107 total rows | MSDEVECI@TESTDB | Modified

RIGHT OUTER JOIN

- Personellerin departmanlarını ve personeli bulunmayan departmanları listeleyelim
- `SELECT first_name, e.last_name, e.department_id, d.department_name
FROM hr.employees e RIGHT OUTER JOIN hr.departments d
ON (e.department_id = d.department_id) ;`

FIRST_NAME	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Jennifer	Whalen	10	Administration
Pat	Fay	20	Marketing
Michael	Hartstein	20	Marketing
Sigal	Tobias	30	Purchasing
Karen	Colmenares	30	Purchasing
Shelli	Baida	30	Purchasing
Den	Raphaely	30	Purchasing
Alexander	Khoo	30	Purchasing

28: 1 | Row 1 of 122 total rows | MSDEVCI@TESTDB | Modified

KARTEZYEN ÇARPIM

- Bir birleştirme şartı ihmal edildiğinde;
 - Bir birleştirme şartı geçersiz olduğunda;
 - Birinci tablodaki bütün satırlar ikinci tablodaki bütün satırlarla birleştirildiğinde, kartezyen çarpım oluşur.
- `SELECT * FROM hr.jobs,hr.departments ;`
 - `SELECT COUNT(*) FROM hr.jobs,hr.departments ;`

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
AD_PRES	President	20080	40000	10	Administration	200	1700
AD_PRES	President	20080	40000	20	Marketing	201	1800
AD_PRES	President	20080	40000	30	Purchasing	114	1700
AD_PRES	President	20080	40000	40	Human Resources	203	2400
AD_PRES	President	20080	40000	50	Shipping	121	1500
AD_PRES	President	20080	40000	60	IT	103	1400
AD_PRES	President	20080	40000	70	Public Relations	204	2700
AD_PRES	President	20080	40000	80	Sales	145	2500

36: 1 Row 1 of 500 fetched so far (more rows exist) MSDEVECI@TESTDB Modified

COUNT(*)
551

97: 1 Row 1 of 1 total rows MSDEVECI@TESTDB

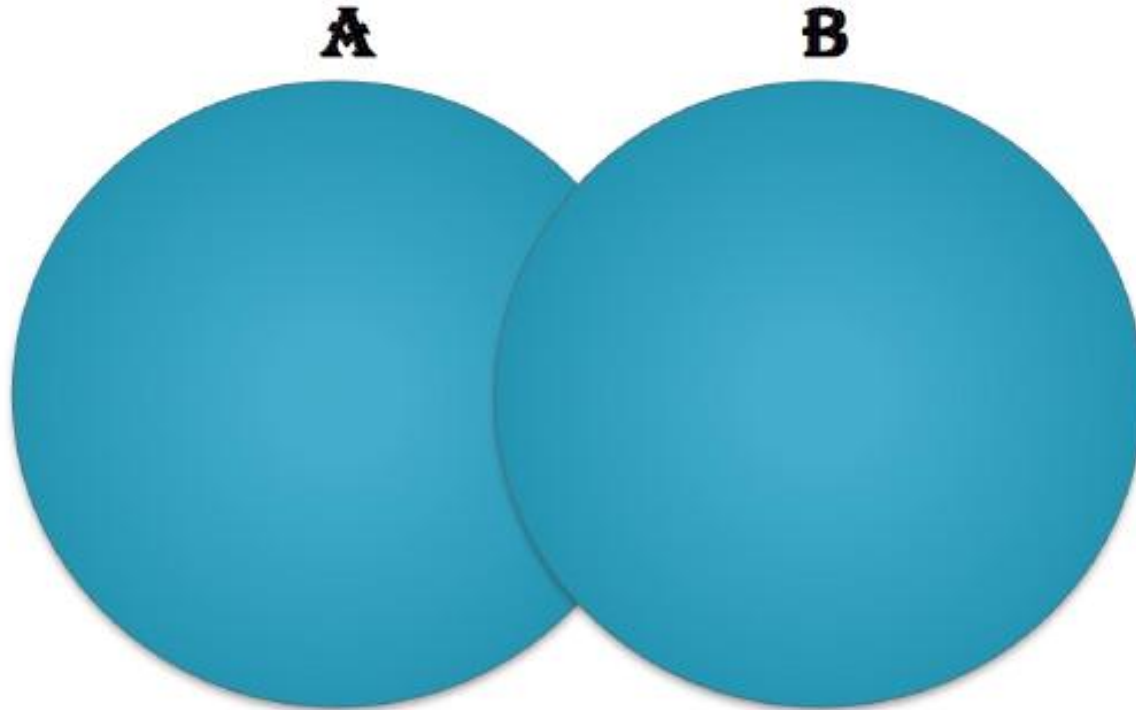
SET OPERATÖRLERİ

- SET operatörleri birden fazla tabloları ve sorguları birleştirmek için kullanılır.
- 4 adet SET operatörü vardır :
 - UNION
 - UNION ALL
 - INTERSECT
 - MINUS



UNION OPERATÖRÜ

- İki veri kümesindeki verileri birleştirir fakat kesişim kümesi olan aynı değere sahip kayıtları almaz.



UNION ÖRNEK

- Personeller ve pozisyon geçmişindeki aynı olmayan kayıtları listeleyelim
- ```
SELECT employee_id, job_id
FROM hr.employees

UNION

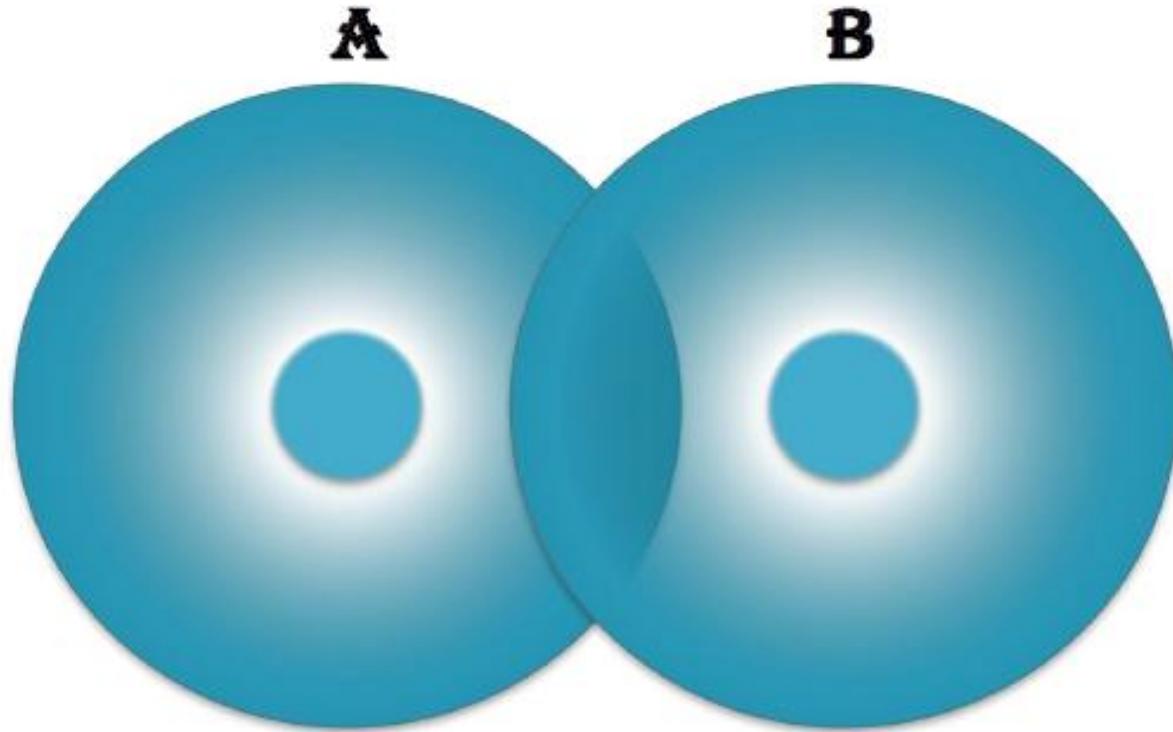
SELECT employee_id, job_id
FROM hr.job_history;
```

| EMPLOYEE_ID | JOB_ID     |
|-------------|------------|
| 100         | AD_PRES    |
| 101         | AC_ACCOUNT |
| 101         | AC_MGR     |
| 101         | AD_VP      |
| 102         | AD_VP      |
| 102         | IT_PROG    |
| 103         | IT_PROG    |
| 104         | IT_PROG    |

38: 1 Row 1 of 115 total rows MSDEVECI@TESTDB

# UNION ALL OPERATÖRÜ

- İki veri kümesindeki verileri birleştirir.
- Aynı değere sahip kayıtlarıda getirir



# UNION ALL ÖRNEK

- Personeller ve pozisyon geçmişindeki aynı olan tüm kayıtları listeleyelim

- ```
SELECT employee_id, job_id, department_id  
FROM hr.employees
```

```
UNION ALL
```

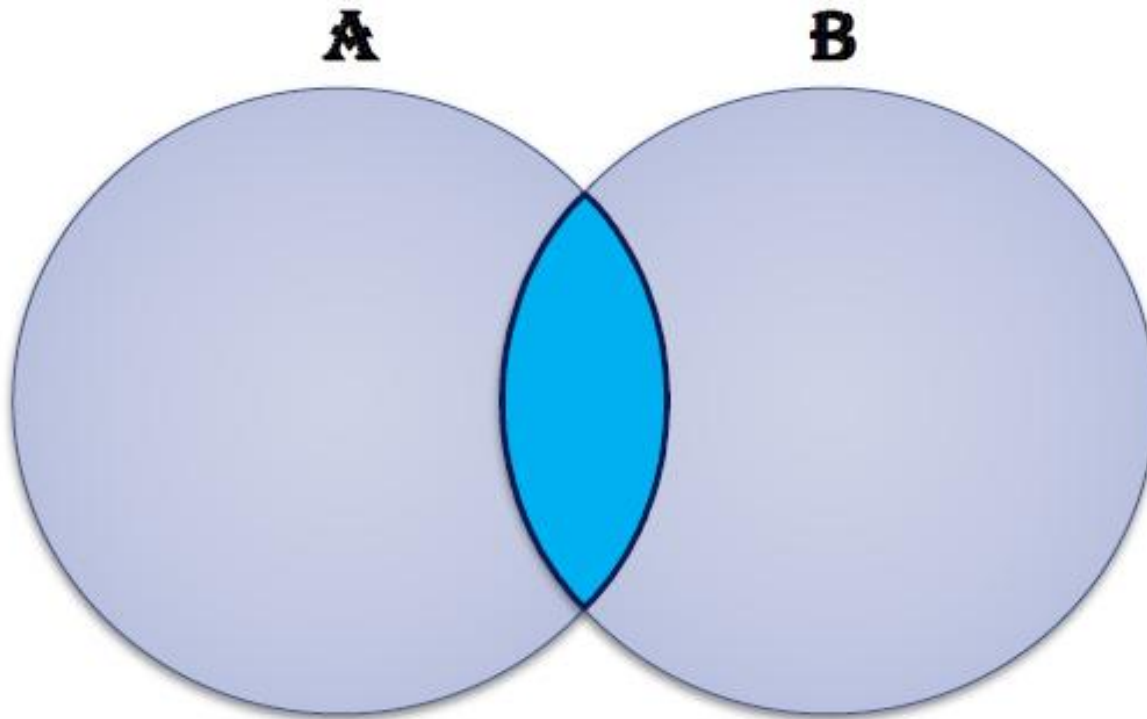
- ```
SELECT employee_id, job_id, department_id
FROM hr.job_history
```

| EMPLOYEE_ID | JOB_ID     | DEPARTMENT_ID |
|-------------|------------|---------------|
| 100         | AD_PRES    | 90            |
| 101         | AD_VP      | 90            |
| 101         | AC_ACCOUNT | 110           |
| 101         | AC_MGR     | 110           |
| 102         | AD_VP      | 90            |
| 102         | IT_PROG    | 60            |
| 103         | IT_PROG    | 60            |
| 104         | IT_PROG    | 60            |

45: 1 | Row 1 of 117 total rows | MSDEVECI@TESTDB

# INTERSECT OPERATÖRÜ

- İki veri kümesindeki verilerin kesişim kümesini bulur.





# INTERSECT ÖRNEK

- Maaşı 1000 ve 5000 arasındaki personeller ile maaşı 3000 ve 7500 arasındaki kesişen personelleri bulalım
- *Select first\_name,last\_name,salary from hr.employees where salary between 1000 and 5000 intersect  
select first\_name,last\_name,salary from hr.employees where salary between 3000 and 7500;*

|   | FIRST_NAME | LAST_NAME | SALARY |
|---|------------|-----------|--------|
| ▶ | Alana      | Walsh     | 3100   |
|   | Alexander  | Khoo      | 3100   |
|   | Alexis     | Bull      | 4100   |
|   | Anthony    | Cabrio    | 3000   |
|   | Britney    | Everett   | 3900   |
|   | Curtis     | Davies    | 3100   |

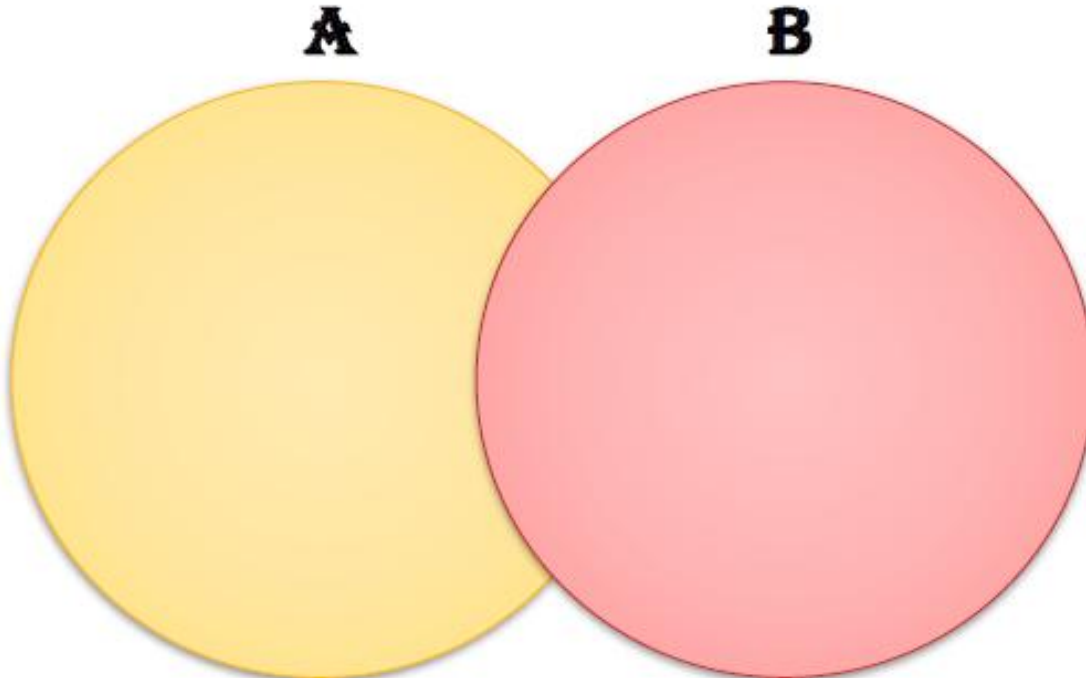
53: 1

Row 1 of 25 total rows

MSDEVECI@TESTDB

# MINUS OPERATÖRÜ

- İki veri kümesindeki verilerin birbirinden farkını gösterir.
- A' da olup B'de olmayanlar yada B'de olup A'da olmayanlar.



# MINUS ÖRNEK

- Employees tablosundaki kayıtların job\_history tablosundaki kayıtlardan farkını listeleyelim
- `SELECT employee_id, job_id`  
`FROM hr.employees`  
`minus`  
`SELECT employee_id, job_id`  
`FROM hr.job_history;`

| EMPLOYEE_ID | JOB_ID  |
|-------------|---------|
| 100         | AD_PRES |
| 101         | AD_VP   |
| 102         | AD_VP   |
| 103         | IT_PROG |
| 104         | IT_PROG |
| 105         | IT_PROG |
| 106         | IT_PROG |
| 107         | IT_PROG |

51: 1 | Row 1 of 105 total rows | MSDEVECI@TESTDB

# ALT SORGULAR

- Uygulamada, bir sorgudan elde edilen sonuç, bir diğer sorguyu ilgilendirebilir. Bu gibi durumlarda alt sorgular ya da bir başka deyişle iç sorgular kullanılır.
- Alt (iç) sorgu, ana (dış) sorgudan önce çalıştırılır.
- Alt sorgunun sonucu ana sorgu tarafından kullanılır.
- Alt sorgular istenilen sonuçların kolay ve anlaşılır bir şekilde bulunmasını sağlar.

```
SELECT liste
FROM tablo
WHERE ifade_işleç
 (
 SELECT liste FROM tablo
)
```

# ALT SORGU ÖRNEK

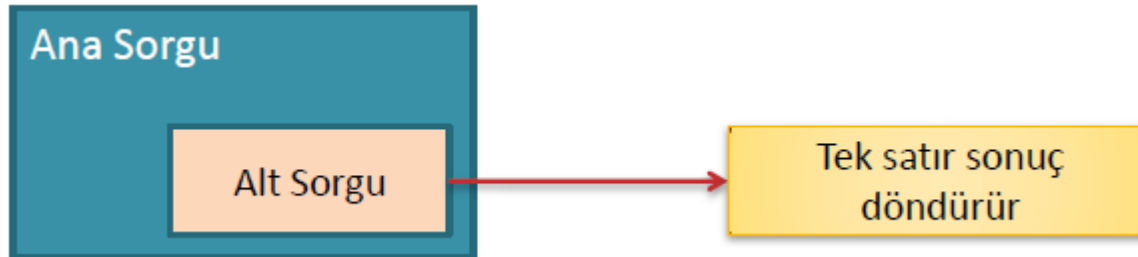
- Soyismi Abel olan personelin maaşından yüksek maaş alan personelleri listeleyelim
- `SELECT first_name, last_name, salary`  
`FROM hr.employees`  
`WHERE salary >`  
`(SELECT salary`  
`FROM hr.employees`  
`WHERE last_name = 'Abel');`

| FIRST_NAME | LAST_NAME | SALARY |
|------------|-----------|--------|
| Michael    | Hartstein | 13000  |
| Shelley    | Higgins   | 12008  |
| Steven     | King      | 24000  |
| Neena      | Kochhar   | 17000  |
| Lex        | De Haan   | 17000  |
| Nancy      | Greenberg | 12008  |
| John       | Russell   | 14000  |
| Karen      | Partners  | 13500  |

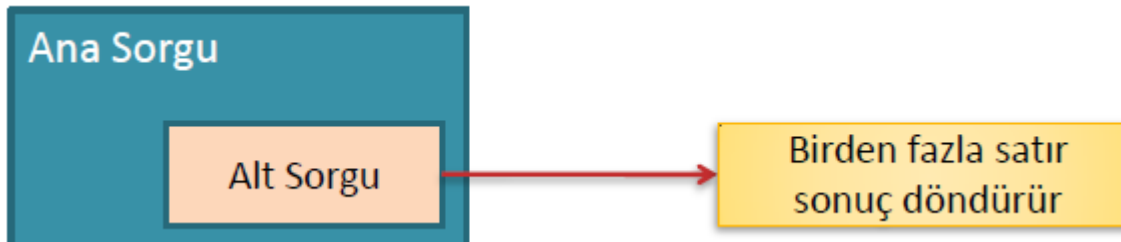
59: 1 Row 1 of 10 total rows MSDEVECI@TESTDB

# ALT SORGU ÇEŞİTLERİ

- Tek Satır Döndüren Alt Sorgular



- Birden Fazla Satır Döndüren Alt Sorgular



# TEK SATIR ALT SORGULAR -1

- Tek satır döndürürler.
- =, <, >, <=, >=, <> operatörleri ile birlikte kullanılırlar.
- Soyadı Austin olan Personelle aynı pozisyonda olup ondan yüksek maaş alan personelleri listeleyelim

```
○ SELECT first_name,last_name, job_id, salary
FROM hr.employees WHERE job_id =
 (SELECT job_id FROM hr.employees
 WHERE last_name = 'Austin')
AND salary >
 (SELECT salary FROM hr.employees
 WHERE last_name = 'Austin');
```

| FIRST_NAME | LAST_NAME | JOB_ID  | SALARY |
|------------|-----------|---------|--------|
| Alexander  | Hunold    | IT_PROG | 9000   |
| Bruce      | Ernst     | IT_PROG | 6000   |

67: 1 | Row 1 of 2 total rows | MSDEVECI@TESTDB

# TEK SATIR ALT SORGULAR -2

- Bu sorgular içerisinde *grup fonksiyonları kullanılabilir.*
- Personeller içerisinde minimum maaşı alan kişiyi listeleyelim

```
○ SELECT first_name, last_name, job_id, salary
FROM hr.employees
WHERE salary =
 (SELECT MIN(salary)
FROM hr.employees);
```

| FIRST_NAME | LAST_NAME | JOB_ID   | SALARY |
|------------|-----------|----------|--------|
| TJ         | Olson     | ST_CLERK | 2100   |

80: 1 Row 1 of 1 total rows MSDEVECI@TESTDB



# TEK SATIR ALT SORGULAR -3

- Alt sorgularla birlikte Having kullanımı.
- 50 nolu departmandaki minimum maaşlı personelden yüksek maaş alan personelleri listeleyelim

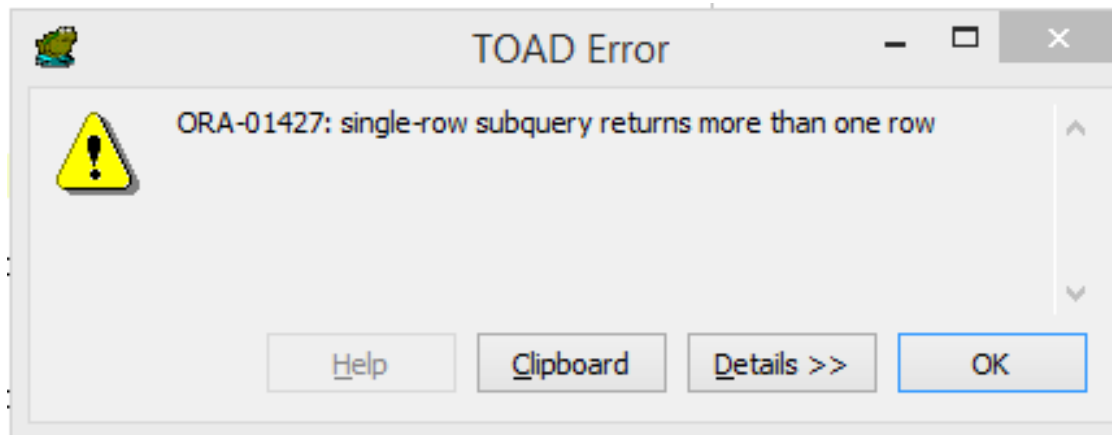
```
SELECT d.department_name,e.department_id, MIN(salary)
FROM hr.employees e,hr.departments d where e.department_id=d.department_id
GROUP BY e.department_id,d.department_name
HAVING MIN(salary) >
 (SELECT MIN(salary) FROM hr.employees
 WHERE department_id = 50);
```

| DEPARTMENT_NAME  | DEPARTMENT_ID | MIN(SALARY) |
|------------------|---------------|-------------|
| Accounting       | 110           | 8300        |
| Purchasing       | 30            | 2500        |
| Marketing        | 20            | 6000        |
| Public Relations | 70            | 10000       |
| Executive        | 90            | 17000       |
| Administration   | 10            | 4400        |

95: 1 Row 1 of 10 total rows MSDEVECI@TESTDB

# TEK SATIR ALT SORGULAR -3

- Departmantların minimum maaşlarına sahip personelleri listelediğimizde hata aldık neden ?
- `SELECT employee_id, last_name  
FROM hr.employees  
WHERE salary =  
    (SELECT MIN(salary)  
    FROM hr.employees  
    GROUP BY department_id);`



# ÇOK SATIR ALT SORGULAR

- Birden fazla satır döndürürler.
- Çok satır karşılaştırma operatörleri ile birlikte kullanılırlar.
- IN
- ANY
- ALL



# IN OPERATÖRÜ

- Bir liste içindeki değerlerden herhangi birine eşit olup olmadığını bulmak için kullanılır.
- `SELECT employee_id, first_name, last_name, salary`  
`FROM hr.employees`  
`WHERE salary in`  
`(SELECT MIN(salary)`  
`FROM hr.employees`  
`GROUP BY department_id);`

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY |
|-------------|------------|-----------|--------|
| 200         | Jennifer   | Whalen    | 4400   |
| 202         | Pat        | Fay       | 6000   |
| 203         | Susan      | Mavris    | 6500   |
| 204         | Hermann    | Baer      | 10000  |
| 206         | William    | Gietz     | 8300   |
| 101         | Neena      | Kochhar   | 17000  |

119: 1 Row 1 of 26 total rows MSDEVECI@TESTDB

# ANY OPERATÖRÜ

- Alt sorgu tarafından üretilen herbir değeri ana sorgu içinde belirlenen bir değerle karşılaştırmak söz konusu ise ANY işleci kullanılır.
- Bu işleç "=", "<", ">" işleçlerinden biri ile birlikte kullanılır. "=ANY" biçimindeki bir tanım IN tanımını ile eşdeğerdir.
- <ANY ifadesi "en büyükten daha küçük";
- >ANY ise "en küçükten daha büyük" anlamında değerlendirilir.



# ANY KULLANIMI

- Personeller içerisinde IT\_PROG pozisyonuna sahip personellerin içerisindeki en yüksek maaş alan personelden daha düşük maaş alan personelleri listeleyelim

```
SELECT employee_id, last_name, job_id, salary
FROM hr.employees
WHERE salary < ANY
 (SELECT salary
 FROM hr.employees
 WHERE job_id = 'IT_PROG')
```

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME  | JOB_ID   | SALARY |
|-------------|------------|------------|----------|--------|
| 132         | TJ         | Olson      | ST_CLERK | 2100   |
| 136         | Hazel      | Philtanker | ST_CLERK | 2200   |
| 128         | Steven     | Markle     | ST_CLERK | 2200   |
| 135         | Ki         | Gee        | ST_CLERK | 2400   |
| 127         | James      | Landry     | ST_CLERK | 2400   |
| 119         | Karen      | Colmenares | PU_CLERK | 2500   |

26 msec  
Row 1 of 76 total rows  
MSDEVECI@TESTDB  
Modified

# ANY KULLANIMI

- Personeller içerisinde IT\_PROG pozisyonuna sahip personellerin içerisindeki en düşük maaş alan personelden yüksek maaş alan personelleri listeleyelim

- `SELECT employee_id, last_name, job_id, salary`

`FROM hr.employees`

`WHERE salary > ANY`

`(SELECT salary`

`FROM hr.employees`

`WHERE job_id = 'IT_PROG')`

`AND job_id <> 'IT_PROG';`

| EMPLOYEE_ID | LAST_NAME | JOB_ID  | SALARY |
|-------------|-----------|---------|--------|
| 100         | King      | AD_PRES | 24000  |
| 101         | Kochhar   | AD_VP   | 17000  |
| 102         | De Haan   | AD_VP   | 17000  |
| 145         | Russell   | SA_MAN  | 14000  |
| 146         | Partners  | SA_MAN  | 13500  |
| 201         | Hartstein | MK_MAN  | 13000  |

27 msec | Row 1 of 57 total rows | MSDEVCI@TESTDB

# ALL OPERATÖRÜ

- Alt sorgu tarafından üretilen tüm değerlerin, ana sorgu için bir girdi olarak kullanılması söz konusu ise ALL işleci kullanılabilir.
- $>ALL$  ifadesi "*en büyükten daha büyük*";
- $<ALL$  ise "*en küçükten daha küçük*" olanları ifade eder.



# ALL KULLANIMI

- Personel tablosunda IT\_PROG pozisyonundaki en düşük maaştan daha düşük maaş alan personelleri listeleyelim
- ```
SELECT employee_id, first_name, last_name, job_id, salary
FROM hr.employees
WHERE salary < ALL
      (SELECT salary FROM hr.employees
       WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID	SALARY
185	Alexis	Bull	SH_CLERK	4100
192	Sarah	Bell	SH_CLERK	4000
193	Britney	Everett	SH_CLERK	3900
188	Kelly	Chung	SH_CLERK	3800
137	Renske	Ladwig	ST_CLERK	3600
189	Jennifer	Dilly	SH_CLERK	3600

135: 1 Row 1 of 44 total rows MSDEVECI@TESTDB Modified

ALL KULLANIMI

- Personel tablosunda IT_PROG pozisyonundaki en yüksek maaştan daha yüksek maaş alan personelleri listeleyelim
- ```
SELECT employee_id, first_name, last_name, job_id, salary
FROM hr.employees
WHERE salary > ALL
 (SELECT salary
 FROM hr.employees
 WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | JOB_ID | SALARY |
|-------------|------------|-----------|--------|--------|
| 157         | Patrick    | Sully     | SA_REP | 9500   |
| 151         | David      | Bernstein | SA_REP | 9500   |
| 163         | Danielle   | Greene    | SA_REP | 9500   |
| 170         | Tayler     | Fox       | SA_REP | 9600   |
| 204         | Hermann    | Baer      | PR_REP | 10000  |
| 169         | Harrison   | Bloom     | SA_REP | 10000  |

144: 1 Row 1 of 23 total rows MSDEVECI@TESTDB Modified