

BÖLÜM- 13: ALT SORGULAR İLE VERİ İŞLEME

- Birden fazla sütuna sahip alt sorgular yazmak
- Scalar alt sorgular yazmak
- Birbiri ile ilişkili alt sorgulardaki problemleri çözmek
- İlişkili alt sorgular kullanarak update yapmak
- EXISTS ve NOT EXISTS operatörlerini kullanmak
- WITH operatörünü kullanmak
- ...

ÇOK SÜTUNLU ALT SORU – ÇİFT YÖNLÜ

- John ile aynı müdüre ve bölüme sahip personelleri listeleyelim
- ```
SELECT employee_id,first_name,last_name,manager_id, department_id
FROM hr.employees
WHERE (manager_id, department_id) IN
 (SELECT manager_id, department_id
 FROM hr.employees
 WHERE first_name = 'John');
```

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | MANAGER_ID | DEPARTMENT_ID |
|-------------|------------|-----------|------------|---------------|
| 137         | Renske     | Ladwig    | 123        | 50            |
| 138         | Stephen    | Stiles    | 123        | 50            |
| 140         | Joshua     | Patel     | 123        | 50            |
| 192         | Sarah      | Bell      | 123        | 50            |
| 193         | Britney    | Everett   | 123        | 50            |

28 msec Row 1 of 15 total rows MSDEVECI@TESTDB Modified

# ÇOK SÜTUNLU ALT SORU – TEK YÖNLÜ

- `SELECT` *employee\_id, first\_name, last\_name, manager\_id, department\_id*  
`FROM` *hr.employees*  
`WHERE` *manager\_id IN*  
    (`SELECT` *manager\_id*  
    `FROM` *hr.employees*  
    `WHERE` *first\_name = 'John'*)  
`AND` *department\_id IN*  
    (`SELECT` *department\_id*  
    `FROM` *hr.employees*  
    `WHERE` *first\_name = 'John'*)  
`AND` *first\_name <> 'John';*

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | MANAGER_ID | DEPARTMENT_ID |
|-------------|------------|-----------|------------|---------------|
| 120         | Matthew    | Weiss     | 100        | 50            |
| 121         | Adam       | Fripp     | 100        | 50            |
| 122         | Payam      | Kaufling  | 100        | 50            |
| 123         | Shanta     | Vollman   | 100        | 50            |
| 124         | Kevin      | Mourgos   | 100        | 50            |

10: 1 Row 1 of 20 total rows MSDEVECI@TESTDB Modified

# SCALAR ALT SORGULAR

- Bu sorgu tiplerinde alt sorgular bir sütun ve bir satır geri dönerler.
- Bu dönülen değerler genelde sayısal değerlerdir.
- Kullanım alanlarına örnek vermek gerekirse :
- DECODE ve CASE gibi şart cümlelerinde
- GROUP BY ifadelerinde
- WHERE, SET ve UPDATE cümlelerinde



# SCALAR ALT SORGULAR

- Personel tablosunda 1800 nolu lokasyona ait personeller Kanada diğerlerini Amerika olacak şekilde listeleyelim
- ```
SELECT employee_id, first_name, last_name,  
(CASE  
  WHEN department_id =  
    (SELECT department_id  
     FROM hr.departments  
    WHERE location_id = 1800)  
  THEN 'Canada' ELSE 'USA' END) location  
FROM hr.employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	LOCATION
198	Donald	OConnell	USA
199	Douglas	Grant	USA
200	Jennifer	Whalen	USA
201	Michael	Hartstein	Canada
202	Pat	Fay	Canada

31: 21 Row 1 of 107 total rows MSDEVECI@TESTDB Modified

İLİŞKİLİ ALT SORGULAR

- İlişkili alt sorgular satır satır çalışır.
- Alt sorgular, dış sorgudaki her bir satır için ayrı ayrı çalışır.

GET

- Dış sorgudaki ilgili satır alınır

EXECUTE

- İç sorgu, dış sorgudaki ilgili satırın değerini kullanarak çalışır

USE

- İç sorgudan gelen değere bakılarak dış sorgudaki satırın alınıp alınmamasına karar verilir

İLİŞKİLİ ALT SORGULAR

- Kendi departmanlarının ortalama maaşından yüksek maaş alan personelleri listeleyelim
- `SELECT first_name, last_name, salary, department_id
FROM hr.employees e
WHERE salary >
 (SELECT AVG(salary)
 FROM hr.employees
 WHERE department_id =
 e.department_id);`

	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
▶	Michael	Hartstein	13000	20
	Shelley	Higgins	12008	110
	Steven	King	24000	90
	Alexander	Hunold	9000	60
	Bruce	Ernst	6000	60
	Nancy	Greenberg	12008	100

34:	1	Row 1 of 38 total rows	MSDEVECI@TESTDB	Modified
-----	---	------------------------	-----------------	----------

İLİŞKİLİ ALT SORGULAR

- En az 2 defa iş değiştirmiş elemanları listeleyelim
- ```
SELECT e.employee_id, first_name, last_name, e.job_id
FROM hr.employees e
WHERE 2<= (SELECT COUNT(*)
 FROM hr.job_history
 WHERE employee_id = e.employee_id);
```

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | JOB_ID  |
|-------------|------------|-----------|---------|
| 200         | Jennifer   | Whalen    | AD_ASST |
| 101         | Neena      | Kochhar   | AD_VP   |
| 176         | Jonathon   | Taylor    | SA_REP  |

⏮

⏪

⏩

⏭

+

-

⬆

✓

✗

↺

\*

⌂

⏴

42: 1

Row 1 of 3 total rows

MSDEVECI@TESTDB

Modified



# EXISTS

- Bu operatör, satırların alt sorgu içerisinde olup olmadığını kontrol eder. Bir çeşit test eder.
- Eğer alt sorguda satıra uygun kayıt bulunursa, sonuç TRUE olur ve alt sorgu içinde arama sona erer.
- Eğer alt sorguda satıra uygun kayıt bulunamazsa, sonuç FALSE olur ve alt sorgu içinde arama devam eder.



# EXISTS

- En az 1 personeli bulunan müdürleri listeleyelim

- `SELECT employee_id, last_name, job_id, department_id`

`FROM hr.employees outer`

`WHERE EXISTS ( SELECT 'T'`

`FROM hr.employees`

`WHERE manager_id =`

`outer.employee_id);`

| EMPLOYEE_ID | LAST_NAME | JOB_ID  | DEPARTMENT_ID |
|-------------|-----------|---------|---------------|
| 201         | Hartstein | MK_MAN  | 20            |
| 205         | Higgins   | AC_MGR  | 110           |
| 100         | King      | AD_PRES | 90            |
| 101         | Kochhar   | AD_VP   | 90            |
| 102         | De Haan   | AD_VP   | 90            |
| 103         | Hunold    | IT_PROG | 60            |

50: 1 Row 1 of 18 total rows MSDEVECI@TESTDB Modified

# NOT EXISTS

- Personeli bulunmayan departmanları listeleyelim

- *SELECT* department\_id, department\_name

*FROM* hr.departments d

*WHERE NOT EXISTS (SELECT 'T'*

*FROM* hr.employees

*WHERE* department\_id = d.department\_id);

| DEPARTMENT_ID | DEPARTMENT_NAME      |
|---------------|----------------------|
| 300           | Engineering          |
| 120           | Treasury             |
| 130           | Corporate Tax        |
| 140           | Control And Credit   |
| 150           | Shareholder Services |
| 160           | Benefits             |

58: 1 Row 1 of 18 total rows MSDEVECI@TESTDB

# İLİŞKİSEL UPDATE

- `ALTER TABLE hr.employees ADD(department_name VARCHAR2(25));`
- `UPDATE hr.employees e`  
`SET department_name =`  
`(SELECT department_name`  
`FROM hr.departments d`  
`WHERE e.department_id = d.department_id);`

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL    | PHONE_NUMBER | HIRE_DATE                    | JOB_ID   | SALARY | COMMISSION_PCT | MANAGER_ID | DEPARTMENT_ID | DEPARTMENT_NAME |
|-------------|------------|-----------|----------|--------------|------------------------------|----------|--------|----------------|------------|---------------|-----------------|
| 198         | Donald     | OConnell  | DOCONNEL | 650.507.9833 | 21-JUN-07 12:00:00.000000 AM | SH_CLERK | 2800   |                | 124        | 50            | Shipping        |
| 199         | Douglas    | Grant     | DGRANT   | 650.507.9844 | 13-JAN-08 12:00:00.000000 AM | SH_CLERK | 2600   |                | 124        | 50            | Shipping        |
| 200         | Jennifer   | Whalen    | JWHALEN  | 515.123.4444 | 17-SEP-03 12:00:00.000000 AM | AD_ASST  | 4400   |                | 101        | 10            | Administration  |
| 201         | Michael    | Hartstein | MHARTSTE | 515.123.5555 | 17-FEB-04 12:00:00.000000 AM | MK_MAN   | 13000  |                | 100        | 20            | Marketing       |
| 202         | Pat        | Fay       | PFAY     | 603.123.6666 | 17-AUG-05 12:00:00.000000 AM | MK_REP   | 6000   |                | 201        | 20            | Marketing       |
| 203         | Susan      | Mavris    | SMAVRIS  | 515.123.7777 | 07-JUN-02 12:00:00.000000 AM | HR_REP   | 6500   |                | 101        | 40            | Human Resources |

64: 1 Row 1 of 107 total rows MSDEVECI@TESTDB Modified

# WITH

- WITH bağlacı sayesinde, karmaşık bir sorguda birden fazla tekrarlayan aynı sorgu bloğunu bir SELECT cümlesinde kullanabiliriz.
- WITH sorgu bloğunun sonuçları kullanıcının geçici tablo alanlarında (temporary tablespace) depolanır.
- WITH bağlacı performans arttıran bir kullanıma sahiptir.

```
With query_name As
(
 SQL query
)
SELECT * FROM query_name;
```

# WITH

- Toplam maaşı Bölümlerinin ortalama maaşından yüksek olan personelleri listeleyelim

- *WITH dept\_costs AS (*

*SELECT d.department\_name, SUM(e.salary) AS dept\_total*

*FROM hr.employees e JOIN hr.departments d*

*ON e.department\_id = d.department\_id*

*GROUP BY d.department\_name),*

*avg\_cost AS (*

*SELECT SUM(dept\_total)/COUNT(\*) AS dept\_avg*

*FROM dept\_costs)*

*SELECT \**

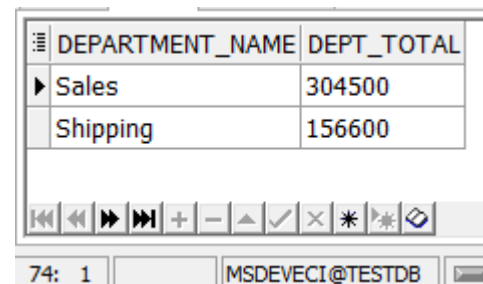
*FROM dept\_costs*

*WHERE dept\_total >*

*(SELECT dept\_avg*

*FROM avg\_cost)*

*ORDER BY department\_name;*



| DEPARTMENT_NAME | DEPT_TOTAL |
|-----------------|------------|
| Sales           | 304500     |
| Shipping        | 156600     |