

# BÖLÜM- 11: BÜYÜK VERİ KÜMELERİ

- Alt sorgular kullanarak verileri değiştirmek
- Insert ve update işleminde DEFAULT ifadesini kullanmak.
- Çoklu insert etme
- Tablo satırlarını birleştirme : Merge

# DEFAULT DEĞERİNİ KULLANMAK

- Tablo kolonu için belirlenmiş default değer atanır.
- Kullanıcı yanlış girişlerini önler.
- INSERT ve UPDATE içerisinde kullanılabilir.
- `INSERT INTO hr.departments (department_id, department_name, manager_id) VALUES (300, 'Engineering', DEFAULT);`
- `select * from hr.departments where department_id=300;`

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
300	Engineering		

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

# DEFAULT DEĞERİNİ KULLANMAK

- Test tablosu oluşturup soyadı kolonunu default yapalım

- `create table test_table`

```
(  
  adi varchar2(40),  
  soyadi varchar2(40) default 'deveci',  
  tarih date  
)
```

- Bir kayıt eklendiğinde soyadı default olarak deveci olarak ekleniyor

- `insert into test_table values('Mehmet Salih', default, sysdate);`

- `select * from test_table;`

ADI	SOYADI	TARİH
Mehmet Salih	deveci	23-Jul-14 10:50:14 AM

7: 1 | Row 1 of 1 total rows | MSDEVCI@TESTDB

# BAŞKA TABLODAN VERİ KOPYALAMAK

- Insert cümlesinde alt sorgular kullanılarak bir tablodan başka bir tabloya veri aktarılabilir.
- VALUES cümlesi kullanılmaz.
- Insert cümlesindeki sütun sayısı, alt sorgudaki sütun sayısı ile aynı olmalıdır.
- Test tablosunu employee tablosunun yapısıyla oluşturalım
- Where 1=0 kullanarak datayı değilde tablonun yapısını sadece alıyorum
- *create table employee\_test as select employee\_id, last\_name, salary, commission\_pct from hr.employees where 1=0;*

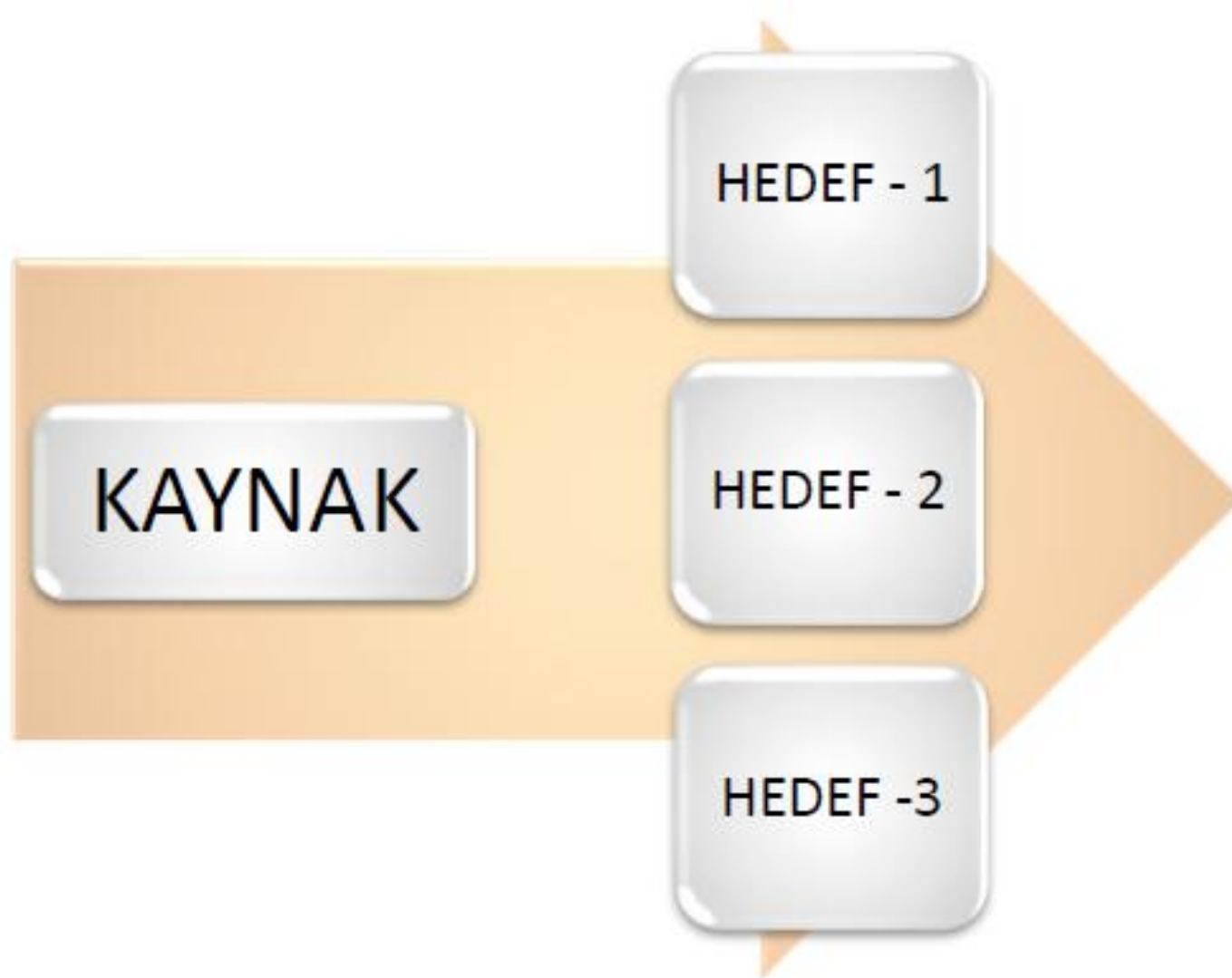
# BAŞKA TABLODAN VERİ KOPYALAMAK

- Test tablosuna Pozisyonunda REP geçen personelleri ekliyorum
- *INSERT INTO* employee\_test  
*SELECT* employee\_id, last\_name, salary, commission\_pct  
*FROM* hr.employees  
*WHERE* job\_id *LIKE* '%REP%';
- *select \* from* employee\_test;

EMPLOYEE_ID	LAST_NAME	SALARY	COMMISSION_PCT
202	Fay	6000	
203	Mavris	6500	
204	Baer	10000	
150	Tucker	10000	0.3
151	Bernstein	9500	0.25
152	Hall	9000	0.25

24 msec Row 1 of 33 total rows MSDEVECI@TESTDB Modified

# ÇOKLU İNSERT



# ÇOKLU INSERT

- Bir insert cümlesi ile aynı anda birden fazla tabloya kayıt insert edilir.
- Alt sorgular kullanılabilir.
- DataWarehouse sistemlerinde çok kullanılır.
- Bir DML cümlesi olduğu için yönetim kolaydır.
- IF—THEN kullanılarak şartlı insert sağlanabilir.

```
INSERT  ALL
  INTO target_a VALUES (... , ... , ...)
  INTO target_b VALUES (... , ... , ...)
  INTO target_c VALUES (... , ... , ...)
  SELECT ...
  FROM   sourcetable
  WHERE ...;
```

# ÇOKLU INSERT ÇEŞİTLERİ





# ÇOKLU INSERT SYNTAX

- Çoklu insert syntax

```
INSERT [conditional_insert_clause]  
[insert_into_clause values_clause] (subquery)
```

- Şartlı insert syntax

```
[ALL] [FIRST]  
[WHEN condition THEN]  
[insert_into_clause values_clause]  
[ELSE] [insert_into_clause values_clause]
```



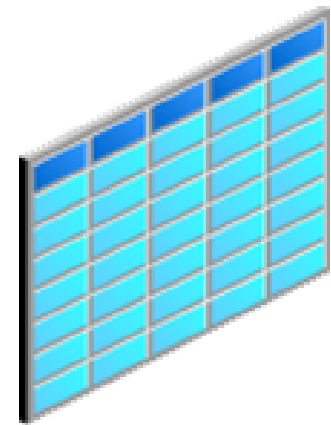
# ŞARTSIZ INSERT ALL

- 2 tane test tablosunu jobs tablosundan oluşturalım
  - `create table bolum_test as select * from hr.jobs where 1=0;`
  - `create table bolum_deneme as select * from hr.jobs where 1=0;`
- Her iki test tablosunada maaşı 4000 den büyük personelleri ekleyelim
  - `INSERT ALL`  
`INTO bolum_test values(job_id,job_title,min_salary,max_salary)`  
`INTO bolum_deneme(job_id,job_title,min_salary,max_salary)`  
`SELECT job_id,job_title,min_salary,max_salary from hr.jobs`  
`WHERE min_salary>4000;`
- `select * from bolum_test;`

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
AD_PRES	President	20080	40000
AD_VP	Administration Vice President	15000	30000
FI_MGR	Finance Manager	8200	16000
FI_ACCOUNT	Accountant	4200	9000

42: 2 | Row 0 of 0 total rows | MSDEVCI@TESTDB | Modified

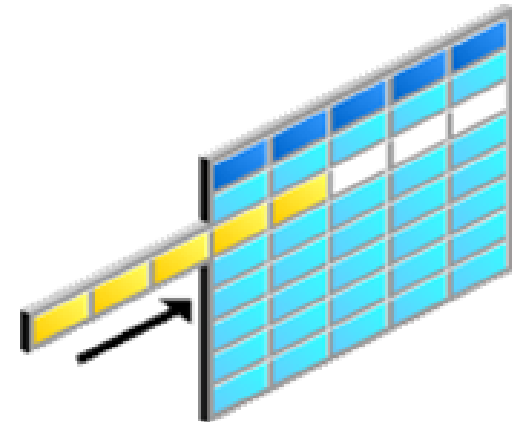
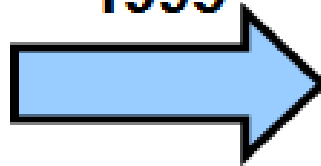
# ŞARTLI INSERT ALL



A 3D grid representing the Employees table, with 5 columns and 10 rows. The top row is dark blue, and the rest are light blue.

Employees

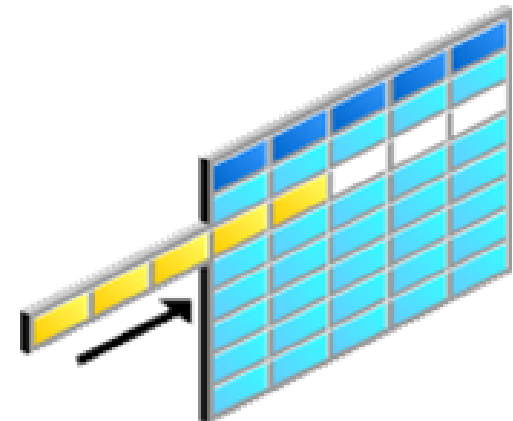
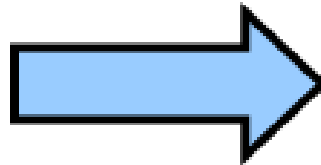
Hired before  
1995



A 3D grid representing the EMP\_HISTORY table, with 5 columns and 10 rows. The top row is dark blue, and the rest are light blue. A yellow bar highlights the first column, and an arrow points to it.

EMP\_HISTORY

With sales  
commission



A 3D grid representing the EMP\_SALES table, with 5 columns and 10 rows. The top row is dark blue, and the rest are light blue. A yellow bar highlights the first column, and an arrow points to it.

EMP\_SALES

# ŞARTLI INSERT ALL

- 2 tane test tablosunu employees tablosundan oluşturalım
- `create table employee_test as select employee_id, first_name, hire_date, salary, commission_pct from hr.employees where 1=0;`
- `create table employee_deneme as select employee_id, first_name, hire_date, salary, commission_pct from hr.employees where 1=0;`
- Employees tablosunun verilerinden 01-JAN-95 tarihinden önce işe girenleri employee\_test, olmayanları ise employee\_deneme tablosuna ekliyoruz
- `INSERT ALL`  
`WHEN hire_date < '01-MAY-03' THEN`  
`INTO employee_test VALUES(employee_id, first_name, hire_date, salary, commission_pct)`  
`WHEN commission_pct IS NOT NULL THEN`  
`INTO employee_deneme VALUES(employee_id, first_name, hire_date, salary, commission_pct)`  
`SELECT employee_id, first_name, hire_date, salary, commission_pct`  
`FROM hr.employees;`

# ŞARTLI INSERT ALL

- Tabloları kontrol ettiğimiz zaman işe giriş tarihleri 01-MAY-03 küçük olan employee\_test, büyük olanlar employee\_deneme tablosuna eklendiğini görebiliriz
- `select * from employee_test;`

EMPLOYEE_ID	FIRST_NAME	HIRE_DATE	SALARY	COMMISSION_PCT
203	Susan	07-JUN-02 12.00.00.000000 AM	6500	
204	Hermann	07-JUN-02 12.00.00.000000 AM	10000	
205	Shelley	07-JUN-02 12.00.00.000000 AM	12008	
206	William	07-JUN-02 12.00.00.000000 AM	8300	
102	Lex	13-JAN-01 12.00.00.000000 AM	17000	
108	Nancy	17-AUG-02 12.00.00.000000 AM	12008	

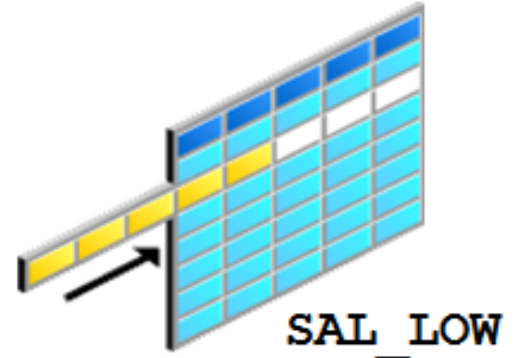
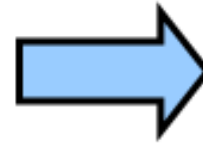
- `select * from employee_deneme;`

EMPLOYEE_ID	FIRST_NAME	HIRE_DATE	SALARY	COMMISSION_PCT
145	John	01-OCT-04 12.00.00.000000 AM	14000	0.4
146	Karen	05-JAN-05 12.00.00.000000 AM	13500	0.3
147	Alberto	10-MAR-05 12.00.00.000000 AM	12000	0.3
148	Gerald	15-OCT-07 12.00.00.000000 AM	11000	0.3
149	Eleni	29-JAN-08 12.00.00.000000 AM	10500	0.2
150	Peter	30-JAN-05 12.00.00.000000 AM	10000	0.3

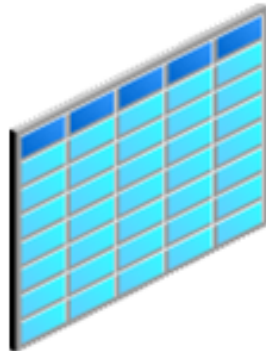
# ŞARTLI INSERT FIRST

**Scenario:** If an employee salary is 2,000, the record is inserted into the SAL\_LOW table only.

Salary < 5,000

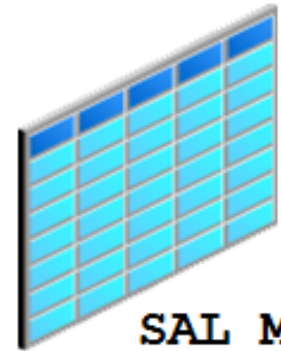


SAL\_LOW



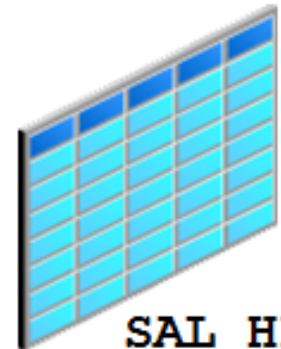
EMPLOYEES

5000 <= Salary  
<= 10,000



SAL\_MID

Otherwise



SAL\_HIGH

# ŞARTLI INSERT FIRST

- 3 tane test tablosu oluşturalım

- `create table salary_high as select employee_id, first_name, salary from hr.employees where 1=0;`
- `create table salary_middle as select employee_id, first_name, salary from hr.employees where 1=0;`
- `create table salary_low as select employee_id, first_name, salary from hr.employees where 1=0;`

- Personel tablosundaki kayıtları, maaşlarına göre 3 sınıfa ayırıp 3 farklı tabloya aynı anda insert edelim.

- `INSERT FIRST`

`WHEN salary < 5000 THEN`

`INTO salary_low VALUES (employee_id, first_name, salary)`

`WHEN salary between 5000 and 10000 THEN`

`INTO salary_middle VALUES (employee_id, first_name, salary)`

`ELSE`

`INTO salary_high VALUES (employee_id, first_name, salary)`

`SELECT employee_id, first_name, salary`

`FROM hr.employees;`



# ŞARTLI INSERT FIRST

- Şartlı insert sonrası tabloları görüntüleyelim
- `select * from salary_high;`

EMPLOYEE_ID	FIRST_NAME	SALARY
201	Michael	13000
205	Shelley	12008
100	Steven	24000
101	Neena	17000

98 msec

MSDEVCI@TESTDB

- `select * from salary_middle;`

EMPLOYEE_ID	FIRST_NAME	SALARY
202	Pat	6000
203	Susan	6500
204	Hermann	10000
206	William	8300

863 msec

MSDEVCI@TESTDB

- `select * from salary_low;`

EMPLOYEE_ID	FIRST_NAME	SALARY
198	Donald	2800
199	Douglas	2600
200	Jennifer	4400
105	David	4800

192 msec

MSDEVCI@TESTDB



# DIKEY INSERT

Emp_ID	Week_ID	MON	TUES	WED	THUR	FRI
176	6	2000	3000	4000	5000	6000



Employee_ID	WEEK	SALES
176	6	2000
176	6	3000
176	6	4000
176	6	5000
176	6	6000

# DIKEY INSERT

- Sales\_source tablosundaki verileri yorumlayıp sales\_info tablosuna dikey şekilde ve tek seferde insert edelim.
- `create table sales_source (empno number(5),  
weekid number(2),  
sales_m number(8,2),  
sales_tu number(8,2),  
sales_w number(8,2),  
sales_th number(8,2),  
sales_f number(8,2));`
- `create table sales_info (empid number(6),  
week number(2),  
SALES NUMBER(8,2));`
- `insert into sales_source values(180,2,1000,2000,3000,4000,5000);`



# DIKEY INSERT

- Dikey insert yaparak sales\_source tablosunun verilerini sales\_info ya insert edelim

- INSERT ALL*

```
INTO sales_info VALUES (empno,weekid,sales_m)
```

```
INTO sales_info VALUES (empno,weekid,sales_tu)
```

```
INTO sales_info VALUES (empno,weekid,sales_w)
```

```
INTO sales_info VALUES (empno,weekid,sales_th)
```

```
INTO sales_info VALUES (empno,weekid,sales_f)
```

```
SELECT empno, weekid, sales_m, sales_tu,
```

```
sales_w, sales_TH,sales_f
```

```
FROM sales_source;
```

- select \* from sales\_info;*

EMPID	WEEK	SALES
180	2	1000
180	2	2000
180	2	3000
180	2	4000
180	2	5000

29 msecs | Row 1 of 5 total rows | MSDEVECI@TESTDB

# MERGE

- Bir tablo üzerinde yapılan insert, update, delete gibi DML işlemlerini yetenekli ve şartlı bir şekilde gerçekleştirir.
- Eğer ilgili satır tabloda var ise update , yok ise kayıt insert olur.
- Ayır ayrı update'lerden kurtarır.
- Performans artışı sağlar.
- DataWarehouse uygulamalarında kullanışlıdır.

```
MERGE INTO table_name table_alias
  USING (table|view|sub_query) alias
  ON (join condition)
  WHEN MATCHED THEN
    UPDATE SET
      col1 = col1_val,
      col2 = col2_val
  WHEN NOT MATCHED THEN
    INSERT (column_list)
    VALUES (column_values);
```

# MERGE

- Employee tablosundan Test tablosu oluşturalım

- *create table merge\_test as select \* from hr.employees where 1=0;*

- *MERGE INTO merge\_test c*

*USING (SELECT \* FROM HR.EMPLOYEES ) e*

*ON (c.employee\_id = e.employee\_id)*

*WHEN MATCHED THEN*

*UPDATE SET*

*c.first\_name = e.first\_name,*

*c.last\_name = e.last\_name*

*WHEN NOT MATCHED THEN*

*INSERT VALUES(e.employee\_id, e.first\_name, e.last\_name,*

*e.email, e.phone\_number, e.hire\_date, e.job\_id,*

*e.salary, e.commission\_pct, e.manager\_id,*

*e.department\_id);*



# MERGE

- Test tablosunu sorguladığımızda employee tablosundaki kayıtların eklendiğini görebiliriz
- *Select \* from merge\_test;*

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
189	Jennifer	Dilly	JDILLY	650.505.2876	13-Aug-05	SH_CLERK	3600		122	50
190	Timothy	Gates	TGATES	650.505.3876	11-Jul-06	SH_CLERK	2900		122	50
191	Randall	Perkins	RPERKINS	650.505.4876	19-Dec-07	SH_CLERK	2500		122	50
192	Sarah	Bell	SBELL	650.501.1876	04-Feb-04	SH_CLERK	4000		123	50
193	Britney	Everett	BEVERETT	650.501.2876	03-Mar-05	SH_CLERK	3900		123	50

114: 1 Row 1 of 107 total rows MSDEVECI@TESTDB Modified