

Manipulating Data

Tujuan Pembelajaran

- Mendeskripsikan setiap statement Data Manipulation Language (DML)
- Menambahkan data ke dalam tabel
- Mengubah data di dalam tabel
- Menghapus data dari suatu tabel
- Transaksi kontrol

Data Manipulation Language

- Statement DML dieksekusi ketika
 - Menambahkan data baru ke dalam tabel
 - Memodifikasi data yang ada pada tabel
 - Menghapus data yang ada dari suatu tabel
- Kontrol transaksi terdiri dari sekumpulan statement DML untuk mengontrol transaksi manipulasi yang dilakukan terhadap suatu sesi pada database

Menambahkan data baru ke dalam tabel

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

70	Public Relations	100	1700
----	------------------	-----	------

New row

Insert new row into the DEPARTMENTS table

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

70	Public Relations	100	1700
----	------------------	-----	------

Syntax Statement INSERT

- Menambahkan data atau baris baru ke dalam tabel dengan menggunakan statement INSERT

```
INSERT INTO  table [(column [, column...])]  
VALUES      (value [, value...]);
```

- Dari syntax ini, hanya satu baris data yang ditambahkan

Menambah atau menyisipkan data baru

- Penambahan data baru mengandung nilai untuk setiap kolom
- Daftar nilainya sesuai urutan default kolom yang ada pada tabel

```
INSERT INTO departments(department_id,  
                        department_name, manager_id, location_id)  
VALUES (70, 'Public Relations', 100, 1700);  
1 row created.
```

- Nilai yang bertipe karakter dan tanggal diberi tanda petik tunggal

Menambah atau menyisipkan data dengan nilai NULL

- Metode implicit: hilangkan kolom yang akan diberi nilai null

```
INSERT INTO departments (department_id,  
                        department_name )  
VALUES (30, 'Purchasing');  
1 row created.
```

- Metode eksplisit: mengisikan NULL pada klausa values

```
INSERT INTO departments  
VALUES (100, 'Finance', NULL, NULL);  
1 row created.
```

Menambah atau menyisipkan nilai khusus

- Fungsi SYSDATE mencatat waktu dan tanggal saat ini

```
INSERT INTO employees (employee_id,  
                        first_name, last_name,  
                        email, phone_number,  
                        hire_date, job_id, salary,  
                        commission_pct, manager_id,  
                        department_id)  
VALUES  
(113,  
  'Louis', 'Popp',  
  'LPOPP', '515.124.4567',  
  SYSDATE, 'AC_ACCOUNT', 6900,  
  NULL, 205, 100);
```

1 row created.

Menambah atau menyisipkan tanggal yang spesifik

- Add a new employee.

```
INSERT INTO employees
VALUES      (114,
            'Den', 'Raphealy',
            'DRAPHEAL', '515.127.4561',
            TO_DATE('FEB 3, 1999', 'MON DD, YYYY'),
            'AC_ACCOUNT', 11000, NULL, 100, 30);
```

1 row created.

- Verify your addition.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_P
114	Den	Raphealy	DRAPHEAL	515.127.4561	03-FEB-99	AC_ACCOUNT	11000	

Menyalin data dari tabel lain

- Tuliskan statement INSERT dengan subquery

```
INSERT INTO sales_reps(id, name, salary, commission_pct)
SELECT employee_id, last_name, salary, commission_pct
FROM employees
WHERE job_id LIKE '%REP%';
```

4 rows created.


- Jangan gunakan klausa values
- Cocokkan jumlah kolom pada klausa insert dengan yang ada pada subquery

Mengubah data

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSION_F
100	Steven	King	SKING	17-JUN-87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	9000	60	
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	6000	60	
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	4200	60	
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800	50	

Update rows in the EMPLOYEES table:



EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSIO
100	Steven	King	SKING	17-JUN-87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	9000	30	
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	6000	30	
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	4200	30	
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800	50	

Syntax statement UPDATE

- Mengubah data yang ada dengan statement UPDATE

```
UPDATE      table
SET         column = value [, column = value, ...]
[WHERE     condition];
```

- Mengubah lebih dari satu baris (jika diperlukan)

Mengubah data pada tabel

- Baris yang spesifik diubah dengan menentukan kondisi

```
UPDATE employees
SET    department id = 70
WHERE  employee_id = 113;
1 row updated.
```

- Semua baris diubah jika klausa where dihilangkan

```
UPDATE    copy_emp
SET       department_id = 110;
22 rows updated.
```

Mengubah dua kolom dengan subquery

- Coba pahami maksud statement update berikut ini !

```
UPDATE employees
SET job_id = (SELECT job_id
              FROM employees
              WHERE employee_id = 205),
    salary = (SELECT salary
              FROM employees
              WHERE employee_id = 205)
WHERE employee_id = 114;
1 row updated.
```

Mengubah data dari tabel lain

- Menggunakan subquery untuk mengubah baris berdasarkan nilai dari tabel lain

```
UPDATE copy_emp
SET department_id = (SELECT department_id
                     FROM employees
                     WHERE employee_id = 100)
WHERE job_id = (SELECT job_id
                FROM employees
                WHERE employee_id = 200);

1 row updated.
```

Menghapus data dari tabel

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing		
100	Finance		
50	Shipping	124	1500
60	IT	103	1400

Delete a row from the DEPARTMENTS table:

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing		
50	Shipping	124	1500
60	IT	103	1400

Statement DELETE

- Menghapus data dari tabel dengan menggunakan statement DELETE

```
DELETE [FROM] table  
[WHERE condition];
```

Menghapus data dari tabel

- Baris tertentu dihapus ketika menggunakan klausa where

```
DELETE FROM departments
WHERE department_name = 'Finance';
1 row deleted.
```

- Semua baris dihapus ketika klausa where dihapus

```
DELETE FROM copy_emp;
22 rows deleted.
```

Menghapus baris berdasarkan tabel lain

- Menggunakan statement DELETE untuk menghapus baris berdasarkan nilai dari tabel lain

```
DELETE FROM employees
WHERE department id =
      (SELECT department_id
       FROM departments
       WHERE department_name
         LIKE '%Public%');
1 row deleted.
```

Statement TRUNCATE

- Menghapus semua data tanpa melakukan proses back up
- Data yang sudah dihapus tidak bisa dikembalikan
 - **Syntax:**

```
TRUNCATE TABLE table_name;
```

- **Example:**

```
TRUNCATE TABLE copy_emp;
```

Menggunakan Subquery dalam statement INSERT

```
INSERT INTO
    (SELECT employee_id, last_name,
            email, hire_date, job_id, salary,
            department_id
     FROM   employees
     WHERE  department_id = 50)
VALUES (99999, 'Taylor', 'DTAYLOR',
        TO_DATE('07-JUN-99', 'DD-MON-RR'),
        'ST_CLERK', 5000, 50);
```

1 row created.

Menggunakan Subquery dalam statement INSERT

Verify the results:

```
SELECT employee_id, last_name, email, hire_date,  
       job_id, salary, department_id  
FROM   employees  
WHERE  department_id = 50;
```

EMPLOYEE_ID	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
124	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800	50
141	Rajs	TRAJS	17-OCT-95	ST_CLERK	3500	50
142	Davies	CDAVIES	29-JAN-97	ST_CLERK	3100	50
143	Matos	RMATOS	15-MAR-98	ST_CLERK	2600	50
144	Vargas	PVARGAS	09-JUL-98	ST_CLERK	2500	50
99999	Taylor	DTAYLOR	07-JUN-99	ST_CLERK	5000	50

6 rows selected.

Transaksi Database

- Transaksi database adalah operasi manipulasi yang mulai dilakukan dari waktu tertentu, kemudian diakhiri dengan menggunakan dengan suatu perintah penutup transaksi
 - Perintah COMMIT
 - Perintah ROLLBACK

Perintah COMMIT

- Operasi terhadap data yang sudah terjadi akan dinyatakan selesai dan semua perubahan yang sudah dilakukan akan disimpan secara permanen pada database

- **Make the changes:**

```
DELETE FROM employees
WHERE  employee_id = 99999;
1 row deleted.

INSERT INTO departments
VALUES (290, 'Corporate Tax', NULL, 1700);
1 row created.
```

- **Commit the changes:**

```
COMMIT;
Commit complete.
```

Perintah ROLLBACK

- Operasi rollback menyatakan bahwa transaksi yang sudah terjadi dibatalkan, data-data yang dimanipulasi tidak akan disimpan dalam database

```
DELETE FROM copy_emp;
```

```
22 rows deleted.
```

```
ROLLBACK ;
```

```
Rollback complete.
```

Apa maksud dari kontrol transaksi berikut?

```
DELETE FROM test;  
25,000 rows deleted.
```

```
ROLLBACK;  
Rollback complete.
```

```
DELETE FROM test WHERE id = 100;  
1 row deleted.
```

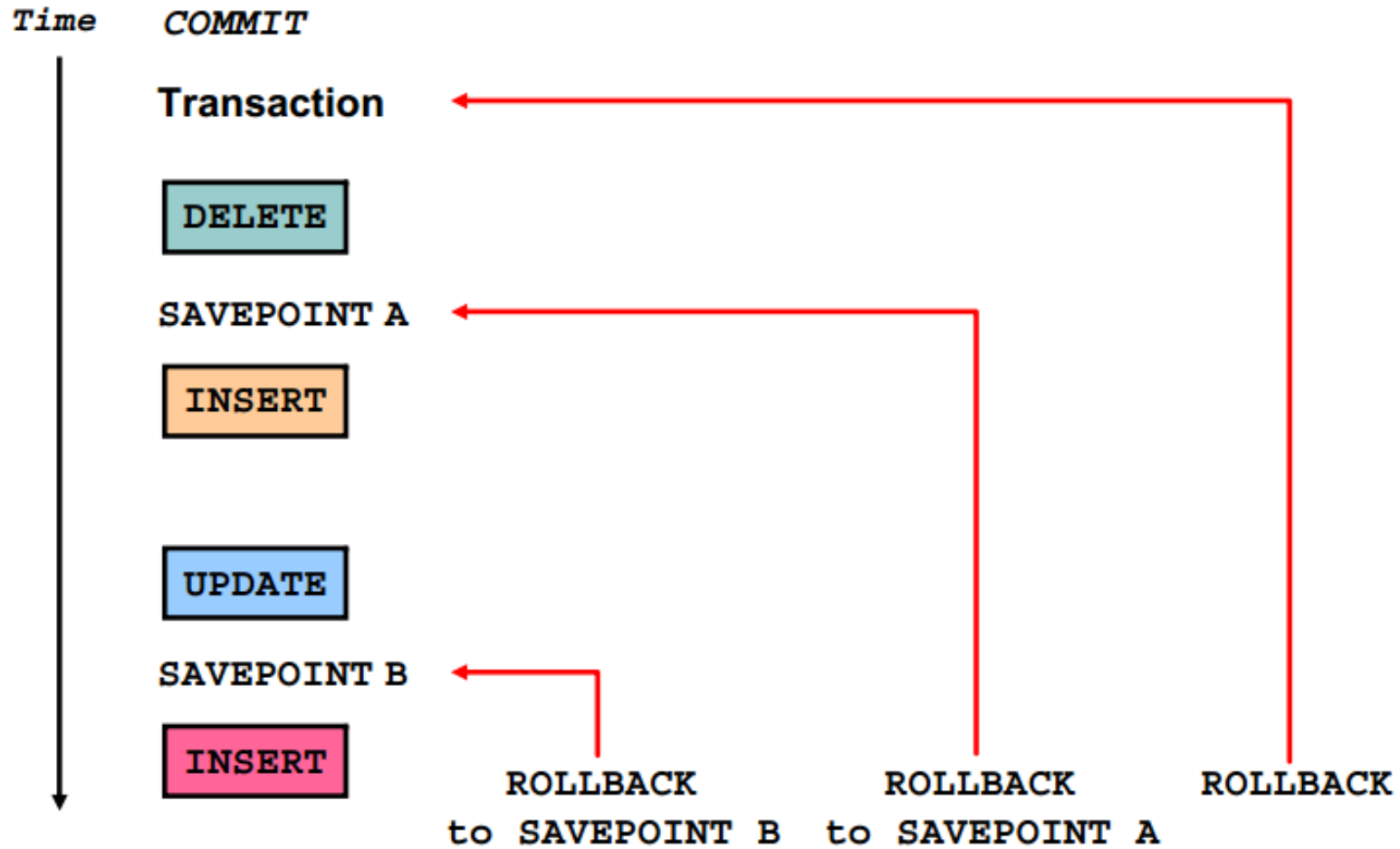
```
SELECT * FROM test WHERE id = 100;  
No rows selected.
```

```
COMMIT;  
Commit complete.
```

SAVEPOINT

- Savepoint merupakan penanda dari transaksi yang sedang berlangsung
- Fungsi: membagi sebuah transaksi menjadi bagian-bagian yang lebih kecil
- Untuk kembali ke savepoint tertentu digunakan perintah `ROLLBACK TO SAVEPOINT`

Pengontrolan Transaksi



Pemrosesan Transaksi secara implisit

- Perintah commit akan secara otomatis dijalankan ketika
 - Perintah DDL dan DCL dijalankan
 - Keluar secara normal dari iSQL*Plus
- Perintah rollback akan secara otomatis dijalankan ketika iSQL*Plus dimatikan secara tidak normal atau terjadi kegagalan sistem