

## Pertemuan 13

### INSERT DAN CREATE TABLE

<b>Tujuan Intruksional</b>	:
Pokok Bahasan ini mempelajari tentang INSERT dan CREATE tabel dalam APEX	
<b>Kompetensi Yang Diharapkan</b>	:
Mahasiswa diharapkan memahami tentang cara INSERT dan CREATE table dalam APEX	
<b>Waktu Pertemuan</b>	: 100 Menit

#### 13.1 INSERT

INSERT adalah sebuah fungsi yang digunakan untuk memasukkan data ke dalam suatu tabel. Dengan fungsi ini kita akan menambahkan satu row/baris data baru pada tabel yang diinginkan.

SQL untuk melakukan INSERT yaitu :

```
INSERT INTO nama_tabel (kolom1, kolom2, kolom3, kolom4)
VALUES (isi_kolom1, isi_kolom2, isi_kolom3, isi_kolom4);
```

Contoh : Kita ingin menambahkan data baru pada tabel employees\_pg.

Sebelum melakukan INSERT ke dalam tabel, kita harus tahu dulu kolom apa saja yang terdapat pada tabel tersebut. Jadi kita perlu melihatnya dengan perintah **SELECT \* FROM**.

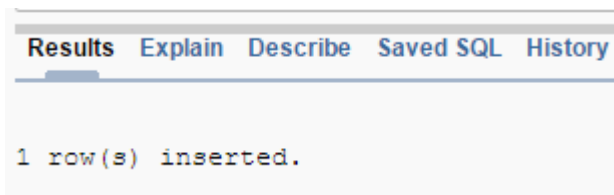
SELECT * from copy_employees;										
Results Explain Describe Saved SQL History										
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17-Jun-1987	AD_PRES	24000	-	-	90

Dari situ kita bisa melihat kolom yang terdapat pada tabel employees\_pg.

Sehingga perintah SQL untuk meng-INSERT data baru yaitu :

```
INSERT INTO employees_pg (employee_id, first_name, last_name, email,
                           phone_number, hire_date, job_id, salary, commission_pct, manager_id,
                           department_id)
VALUES (300, 'Hanifah', 'Atiya', 'hanifah@unila.ac.id', '082372882000', '05-Dec-2017',
        'AD_MGR', 50000, '.3', 100, 80);
```

Hasil :



The screenshot shows a SQL interface with tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, displaying the message "1 row(s) inserted." in a monospaced font.

Results Explain Describe Saved SQL History

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
300	Fadhli	Iman	fadhli@unila.ac.id	0857123456	06-Jun-2016	AD_MGR	50000	.3	100	80

Selain dengan menggunakan SQL diatas, kita juga bisa melakukan INSERT pada TABEL tanpa perlu menuliskan nama kolom pada tabel, dengan syarat kita harus mengisi semua VALUES pada setiap kolom tabel.

SQL:

```
INSERT INTO employees_pg
VALUES (300, 'Hanifah', 'Atiya', 'hanifah@unila.ac.id', '082372882000', '05-Dec-2017',
        'AD_MGR', 50000, '.3', 100, 80);
```

Namun untuk menghindari kekeliruan lebih dianjurkan untuk menuliskan setiap nama kolom pada tabel yang akan di-INSERT.

## HAL-HAL yang perlu diperhatikan dalam melakukan INSERT pada TABLE.

- Untuk setiap kolom yang memiliki tipe data selain INTEGER (misal VARCHAR, DATE, dll) harus menggunakan tanda petik ( ' ) pada VALUES yang akan di-INSERT.
- Jika ingin mengosongkan salah satu kolom pada TABEL yang akan di-INSERT (memasukkan nilai NULL) maka dapat dilakukan dengan cara sebagai berikut:  
Contoh : kita akan menginputkan data ke tabel employees\_pg, tetapi tidak mengisi kolom **manager\_id** dan **department\_id**.

Maka ada 2 cara untuk SQL yang dipakai :

Cara 1 (mengosongkan nilai VALUES):

```
INSERT INTO employees_pg (employee_id, first_name, last_name, email,  
                           phone_number, hire_date, job_id, salary, commission_pct,  
                           manager_id, department_id)  
VALUES (300, 'Hanifah', 'Atiya', 'hanifah@unila.ac.id', '082372882000', '05-Dec-  
2017', 'AD_MGR', 50000, '.3', ' ', '');
```

Cara 2 (tidak memasukkan kolom yang ingin dikosongkan):

```
INSERT INTO employees_pg (employee_id, first_name, last_name, email,  
                           phone_number, hire_date, job_id, salary, commission_pct)  
VALUES (300, 'Hanifah', 'Atiya', 'hanifah@unila.ac.id', '082372882000', '05-Dec-  
2017', 'AD_MGR', 50000, '.3');
```

Sebelum memasukkan nilai null kita harus mengecek terlebih dahulu apakah kolom tersebut bisa diisi NULL (NULLABLE) atau tidak. Caranya adalah dengan menggunakan query SQL :

```
DESCRIBE employees_pg
```

Hasil :

Object Type TABLE Object COPY\_EMPLOYEES

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COPY_EMPLOYEES	EMPLOYEE_ID	NUMBER	-	6	0	-	✓	-	-
	EMPLOYEE_ID	NUMBER	-	6	0	-	✓	-	-
	FIRST_NAME	VARCHAR2	20	-	-	-	✓	-	-
	FIRST_NAME	VARCHAR2	20	-	-	-	✓	-	-
	LAST_NAME	VARCHAR2	25	-	-	-	-	-	-
	LAST_NAME	VARCHAR2	25	-	-	-	-	-	-
	EMAIL	VARCHAR2	25	-	-	-	-	-	-
	EMAIL	VARCHAR2	25	-	-	-	-	-	-
	PHONE_NUMBER	VARCHAR2	20	-	-	-	✓	-	-
	PHONE_NUMBER	VARCHAR2	20	-	-	-	✓	-	-
	HIRE_DATE	DATE	7	-	-	-	-	-	-
	HIRE_DATE	DATE	7	-	-	-	-	-	-
	JOB_ID	VARCHAR2	10	-	-	-	-	-	-
	JOB_ID	VARCHAR2	10	-	-	-	-	-	-
	SALARY	NUMBER	-	8	2	-	✓	-	-
	SALARY	NUMBER	-	8	2	-	✓	-	-
	COMMISSION_PCT	NUMBER	-	2	2	-	✓	-	-
	COMMISSION_PCT	NUMBER	-	2	2	-	✓	-	-
	MANAGER_ID	NUMBER	-	6	0	-	✓	-	-
	MANAGER_ID	NUMBER	-	6	0	-	✓	-	-
	DEPARTMENT_ID	NUMBER	-	4	0	-	✓	-	-

Dari situ kita bisa melihat kolom mana yang bisa diisi nilai NULL dan yang tidak bisa.

## 13.2 CREATE

Fungsi CREATE digunakan untuk membuat sebuah TABEL baru di dalam database.

Sebelum membuat TABEL baru ada beberapa hal yang perlu diperhatikan dalam memberi nama pada TABEL dan KOLOM pada tabel tersebut :

- Harus diawali dengan huruf.
- Harus terdiri dari 1-30 karakter.
- Karakter yang diperbolehkan adalah huruf A-Z, angka 0-9, \_ (underscore), \$, dan #.
- Nama tabel tidak boleh sama dengan tabel yang sudah ada.
- Nama kolom harus berbeda satu sama lain dalam satu tabel.

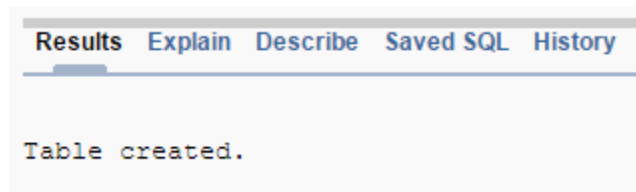
SQL yang digunakan untuk membuat sebuah TABEL yaitu :

```
CREATE TABLE nama_tabel (kolom1 TИPЕDATA (panjang), kolom2 TИPЕDATA  
(panjang), .....);
```

Contoh: kita akan membuat sebuah tabel baru dengan nama Mahasiswa, maka SQL yang digunakan yaitu ;

```
CREATE TABLE mahasiswa  
(nama VARCHAR (50) ,  
npm VARCHAR (15),  
jurusan VARCHAR (30),  
tanggal_lahir DATE);
```

Hasil:



## SOAL LATIHAN (POSTTEST)

1. Buatlah perintah SQL untuk meng-INSERT data ke tabel **employees\_pg**, dengan ketentuan sebagai berikut :
  - First name dan last name menggunakan nama kalian masing-masing.
  - Phone number diisi NPM kalian masing-masing.
  - Hire date diisi tanggal lahir.
  - Job ID diisi dengan :
    - Untuk kelas A : CA-STU
    - Untuk kelas B : CB-STU
    - Untuk kelas C : CC-STU
    - Untuk kelas D : CD-STU
  - Salary diisi 5 digit terakhir NPM masing-masing.
  - Commision PCT dan Manager ID dikosongkan.
  - Department ID diisi 2 digit terakhir NPM.
2. Buatlah perintah SQL untuk membuat tabel yang memiliki **minimal 5 kolom** dan memiliki **minimal 5 tipe data** yang berbeda.

Nama Table: nama kemudian 4 digit terakhir NPM

Contoh: HANIFAH1063