

# **Oracle Academic Initiative**

## **Oracle9i Introduction to SQL**



**Oleh:**

**Tessy Badriyah, SKom.MT**

**Politeknik Elektronika Negeri Surabaya  
Institut Teknologi Sepuluh Nopember  
Surabaya**

## BAB 12 : Pembuatan View

### 12.1. Sasaran

- Memahami definisi View
- Dapat membuat View
- Dapat Memanggil data melalui View
- Merubah definisi View
- Insert, Update, dan Delete data melalui View
- Menghapus (drop) view

### 12.2. Definisi View

View adalah salah satu object database, yang secara logika merepresentasikan sub himpunan dari data yang berasal dari satu atau lebih table.

Kegunaan dari view adalah :

- Membatasi akses database
- Membuat query kompleks secara mudah
- Mengijinkan independensi data
- Untuk menampilkan view (pandangan) data yang berbeda dari data yang sama.

Ada 2 (dua) tipe view, yaitu Simple View dan Complex View.

Berikut ini perbandingan antara Simple View dan Complex View :

Fitur	Simple View	Complex View
Jumlah table	Satu	Satu atau lebih
Berisi Fungsi	Tidak	Ya
Berisi Group Data	Tidak	Ya
DML melalui view	Ya	Tidak selalu

### 12.3. Pembuatan View

View dapat dibuat dengan perintah CREATE VIEW. Subquery dapat dicantumkan dalam CREATE VIEW, tapi subquery yang digunakan tidak boleh berisi klausa ORDER BY.

Sintak penulisan VIEW :

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view
  [(alias[, alias]...)]
  AS subquery
[WITH CHECK OPTION [CONSTRAINT constraint]]
[WITH READ ONLY [CONSTRAINT constraint]];
```

### 12.4. Memanggil Data dari View

Buat view EMPVU80 yang berisi detail pegawai yang bekerja di department 80.

```
CREATE VIEW empvu80
  AS SELECT employee_id, last_name, salary
     FROM employees
     WHERE department_id = 80;
View created.
```

Untuk menampilkan struktur dari view digunakan perintah DESCRIBE :

```
DESCRIBE empvu80
```

Perintah pembuatan View dapat dengan menggunakan kolom alias :

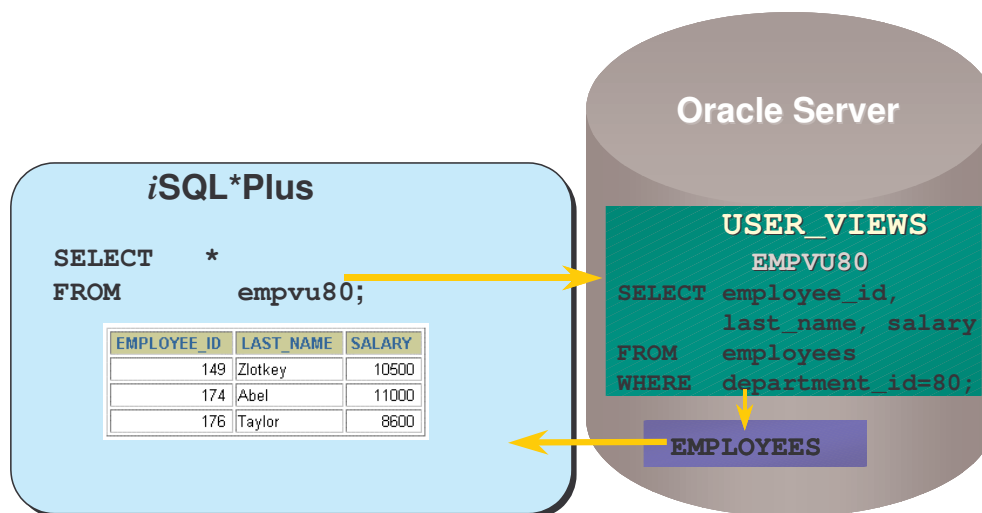
```
CREATE VIEW salvu50
AS SELECT employee_id ID_NUMBER, last_name NAME,
        salary*12 ANN_SALARY
FROM employees
WHERE department_id = 50;
View created.
```

Untuk memanggil data dari view, digunakan perintah yang sama seperti memanggil data dari table.

```
SELECT *
FROM salvu50;
```

ID_NUMBER	NAME	ANN_SALARY
124	Mourgos	69600
141	Rajs	42000
142	Davies	37200
143	Matos	31200
144	Vargas	30000

Proses pembuatan view :



## 12.5. Memodifikasi data View

Untuk memodifikasi View digunakan klausa CREATE OR REPLACE VIEW.

```
CREATE OR REPLACE VIEW empvu80
(id_number, name, sal, department_id)
AS SELECT employee_id, first_name || ' ' || last_name,
        salary, department_id
FROM employees
WHERE department_id = 80;
View created.
```

## 12.6. Pembuatan Complex View

Berikut ini akan dicontohkan pembuatan Complex View yang berisi fungsi group untuk menampilkan nilai yang berasal dari dua table.

```
CREATE VIEW dept_sum_vu
  (name, minsal, maxsal, avgsal)
AS SELECT    d.department_name, MIN(e.salary),
             MAX(e.salary), AVG(e.salary)
FROM        employees e, departments d
WHERE       e.department_id = d.department_id
GROUP BY    d.department_name;
View created.
```

## 12.7. Aturan untuk membentuk Operasi DML pada View

Berikut ini aturan untuk membentuk operasi DML pada View :

- Operasi DML dapat dibentuk pada Simple View
- Baris data pada View tidak dapat dihapus, jika berisi :
  - Fungsi Group
  - Klausula GROUP BY
  - Keyword DISTINCT
- Data pada View tidak bisa dimodifikasi jika berisi :
  - 3 Kondisi yang sudah disebutkan diatas
  - Kolom yang didefinisikan oleh suatu ekspresi
  - Kolom ROWNUM
- Pada View tidak bisa ditambahkan data, jika :
  - View berisi 5 kondisi yang sudah disebutkan diatas

Terdapat kolom NOT NULL pada *base table* (table asal darimana view dibuat) yang tidak dipilih oleh View.

## 12.8. Menggunakan Klausula WITH CHECK OPTION

Jika klausula WITH CHECK OPTION digunakan, maka tidak diperbolehkan terjadi perubahan data pada kolom yang punya relasi ke table yang lain. Misal pada view EMPVU20 kolom deptno punya relasi ke kolom deptno pada table department, maka perubahan data yang dilakukan pada kolom ini tidak diperbolehkan.

```
CREATE OR REPLACE VIEW empvu20
AS SELECT    *
FROM        employees
WHERE       department_id = 20
WITH CHECK OPTION CONSTRAINT empvu20_ck ;
View created.
```

## 12.9. Mengabaikan Operasi DML

Klausula READ ONLY digunakan jika kita ingin mengabaikan atau tidak mengijinkan semua operasi DML yang dilakukan pada data.

```
CREATE OR REPLACE VIEW empvu10
  (employee_number, employee_name, job_title)
AS SELECT    employee_id, last_name, job_id
FROM        employees
WHERE       department_id = 10
WITH READ ONLY;
View created.
```

### 12.10. Menghapus View

View dapat dihapus dengan menggunakan perintah DROP VIEW *nama\_view*;

Sintak penulisan untuk menghapus view :

```
DROP VIEW view;
```

Contoh penghapusan View :

```
DROP VIEW empvu80;
View dropped.
```

### 12.11. Inline View

Inline view adalah subquery dengan nama alias yang digunakan dalam SQL Statement yang digunakan untuk membuat view. Sebuah inline view bukan merupakan object dari suatu schema.

### 12.12. Top-N Analysis

Top N-queries adalah query untuk mendapatkan n buah nilai terbesar atau terkecil dari suatu kolom Semisal : produk apa saja yang penjualannya 10 terbesar ?

### 12.13. Membentuk Top-N Analysis

Sintak penulisan Top-N Analysis :

```
SELECT [column_list], ROWNUM
FROM (SELECT [column_list]
      FROM table
      ORDER BY Top-N_column)
WHERE ROWNUM <= N;
```

### 12.14. Contoh Top-N Analysis

Untuk menampilkan nama dan salary dari pegawai yang memiliki penghasilan tiga besar ?

```
SELECT ROWNUM as RANK, last_name, salary
FROM (SELECT last_name, salary FROM employees
      ORDER BY salary DESC)
WHERE ROWNUM <= 3;
```

RANK	LAST_NAME	SALARY
1	King	24000
2	Kochhar	17000
3	De Haan	17000

## 12.15. Latihan

1. Buat view EMP\_VU yang berisi nomer pegawai, nama pegawai, nomer department yang berasal dari table pegawai. Ubah judul kolom nama pegawai menjadi PEGAWAI.
2. Tampilkan view EMP\_VU

EMPNO	PEGAWAI	DEPTNO
7839	KING	10
7698	BLAKE	30
7782	CLARK	10
7566	JONES	20
7654	MARTIN	30
7499	ALLEN	30
7844	TURNER	30
7900	JAMES	30
7521	WARD	30
7902	FORD	20
7369	SMITH	20
-----		
EMPNO	PEGAWAI	DEPTNO
7788	SCOTT	20
7876	ADAMS	20
7934	MILLER	10

14 rows selected.

3. Tampilkan nama view dan teks-nya dari data dictionary USER\_VIEWS

```
SQL> SELECT view_name,text from user_views;
```

VIEW_NAME	TEXT
EMP_VU	SELECT empno,ename PEGAWAI, deptno FROM EMP

4. Buat view dengan nama DEPT20 yang berisi nomer, nama dan gaji dari pegawai yang bekerja di department 20. Beri judul kolom EMPLOYEE\_ID, EMPLOYEE, dan DEPARTMENT\_ID. Jangan perbolehkan pegawai untuk mendaftar kembali (atau mengisi datanya lagi) ke department yang lain melalui view.
5. Tampilkan struktur dari view DEPT20.

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(4)
EMPLOYEE		VARCHAR2(10)
DEPARTMENT_ID	NOT NULL	NUMBER(2)

6. Buat view SALARY\_VU yang berisi nama pegawai, nama department, gaji dan grade dari gaji untuk semua pegawai. Beri judul PEGAWAI, DEPARTMENT, GAJI, GRADE. Tampilkan data pada SALARY\_VU.

ENAME	DNAME	SAL	GRADE
JAMES	SALES	950	1
SMITH	RESEARCH	800	1
ADAMS	RESEARCH	1100	1
MARTIN	SALES	1250	2
WARD	SALES	1250	2
MILLER	ACCOUNTING	1300	2
ALLEN	SALES	1600	3
TURNER	SALES	1500	3
BLAKE	SALES	2850	4
CLARK	ACCOUNTING	2450	4
JONES	RESEARCH	2975	4
-----			
ENAME	DNAME	SAL	GRADE
FORD	RESEARCH	3000	4
SCOTT	RESEARCH	3000	4
KING	ACCOUNTING	5000	5

14 rows selected.