

# **Oracle Academic Initiative**

## **Oracle9i Introduction to SQL**



**Oleh:**

**Tessy Badriyah, SKom.MT**

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

## BAB 3 : Pembatasan dan Pengurutan Data

### 3.1. Sasaran

- Membatasi baris yang didapatkan dari suatu query
- Mengurutkan baris yang didapat dari suatu query

### 3.2. Membatasi Baris Menggunakan Klausa WHERE

Baris-baris data yang dihasilkan dari suatu query dapat dibatasi dengan memberikan klausa WHERE. Bentuk umumnya sebagai berikut :

```
SELECT [DISTINCT] {*, column [alias], ... }
FROM table;
[WHERE condition(s)];
```

#### EMPLOYEES

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
104	Ernst	IT_PROG	60
107	Lorentz	IT_PROG	60
124	Mourgos	ST_MAN	50

20 rows selected.

**“mendapatkan pegawai yang bekerja di departemen 90”**

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

Perintah SQLnya :

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE department_id = 90 ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

### 3.3. String Karakter dan Tanggal

Penggunaan string karakter dan tanggal pada klausa WHERE menggunakan tanda petik tunggal. Format default dari tipe data tanggal adalah DD-MON-YY.

```
SELECT last_name, job_id, department_id
FROM employees
WHERE last_name = 'Whalen';
```

### 3.4. Kondisi Perbandingan

Operator perbandingan dapat digunakan pada klausa WHERE, dan mempunyai sintak sebagai berikut :  
 WHERE *expr operator value*

```
SELECT last_name, salary
FROM employees
WHERE salary <= 3000;
```

LAST_NAME	SALARY
Matos	2600
Vargas	2500

### 3.5. Kondisi Perbandingan yang lain

Operator	Meaning
BETWEEN ...AND...	Between two values (inclusive),
IN (set)	Match any of a list of values
LIKE	Match a character pattern
IS NULL	Is a null value

### 3.6. Menggunakan kondisi BETWEEN

Operator BETWEEN digunakan untuk menampilkan baris berdasarkan suatu jangkauan (range) nilai.

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 2500 AND 3500;
```

Lower limit      Upper limit

LAST_NAME	SALARY
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500

### 3.7. Menggunakan Kondisi IN

Operator IN digunakan untuk menguji nilai-nilai yang ada dalam list (tanda kurung).

```
SELECT employee_id, last_name, salary, manager_id
FROM employees
WHERE manager_id IN (100, 101, 201);
```

EMPLOYEE_ID	LAST_NAME	SALARY	MANAGER_ID
202	Fay	6000	201
200	Whalen	4400	101
205	Higgins	12000	101
101	Kochhar	17000	100
102	De Haan	17000	100
124	Mourgos	5800	100
149	Zlotkey	10500	100
201	Hartstein	13000	100

8 rows selected.

### 3.8. Menggunakan Kondisi LIKE

Operator LIKE digunakan untuk membentuk pencarian string yang sesuai dengan nilai yang dicantumkan pada kondisi. Kondisi pencarian dapat berisi karakter atau bilangan, dan 2 (dua) simbol khusus berikut :

- % menotasikan zero atau banyak karakter
- \_ menotasikan satu karakter

```
SELECT first_name
FROM employees
WHERE first_name LIKE 'S%';
```

```
SELECT last_name
FROM employees
WHERE last_name LIKE '_o%';
```

LAST_NAME
Kochhar
Lorentz
Mourgos

### 3.9. Menggunakan Kondisi NULL

Untuk menguji adanya nilai NULL, digunakan operator IS NULL.

```
SELECT last_name, manager_id
FROM employees
WHERE manager_id IS NULL;
```

LAST_NAME	MANAGER_ID
King	

### 3.9. Kondisi Logika

Operator logika mengkombinasikan hasil dari dua komponen kondisi. 3 (tiga) macam operator logika yang tersedia, yaitu AND, OR, dan NOT.

### 3.10. Menggunakan Operator AND

Operator logika AND akan bernilai benar jika semua kondisi benar.

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary >= 10000
AND job_id LIKE '%MAN%';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
149	Zlotkey	SA_MAN	10500
201	Hartstein	MK_MAN	13000

### 3.11. Menggunakan Operator OR

Operator logika OR bernilai benar jika salah satu dari kondisi bernilai benar.

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary >= 10000
OR job_id LIKE '%MAN%';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
102	De Haan	AD_VP	17000
124	Mourgos	ST_MAN	5800
149	Zlotkey	SA_MAN	10500
174	Abel	SA_REP	11000
201	Hartstein	MK_MAN	13000
205	Higgins	AC_MGR	12000

8 rows selected.

### 3.12. Menggunakan Operator NOT

Berikut ini contoh penggunaan dari operator logika NOT :

```
SELECT last_name, job_id
FROM employees
WHERE job_id
NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP');
```

LAST_NAME	JOB_ID
King	AD_PRES
Kochhar	AD_VP
De Haan	AD_VP
Mourgos	ST_MAN
Zlotkey	SA_MAN
Whalen	AD_ASST
Hartstein	MK_MAN
Fay	MK_REP
Higgins	AC_MGR
Gietz	AC_ACCOUNT

10 rows selected.

### 3.13. Aturan Presedensi

Berikut ini table aturan presedensi :

Urutan operasi	Operator
1	Semua operator perbandingan
2	NOT
3	AND
4	OR

Perhatikan contoh berikut ini, dan perhatikan hasil yang diperoleh berdasarkan presedensi operator yang digunakan :

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id = 'SA_REP'
OR job_id = 'AD_PRES'
AND salary > 15000;
```

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000
Abel	SA_REP	11000
Taylor	SA_REP	8600
Grant	SA_REP	7000

Gunakan tanda kurung untuk menunjukkan prioritas :

```
SELECT last_name, job_id, salary
FROM employees
WHERE (job_id = 'SA_REP'
OR job_id = 'AD_PRES')
AND salary > 15000;
```

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000

### 3.14. Klausula ORDER BY

Klausula ORDER BY digunakan untuk mengurutkan data hasil query. Terdapat 2 (dua) jenis : ASC (Ascending – urut naik), dan DESC (Descending – urut turun).

Secara default jika tidak ditampilkan maka dianggap urut naik.

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
King	AD_PRES	90	17-JUN-87
Whalen	AD_ASST	10	17-SEP-87
Kochhar	AD_VP	90	21-SEP-89
Hunold	IT_PROG	60	03-JAN-90
Ernst	IT_PROG	60	21-MAY-91

...  
20 rows selected.



### 3.15. Pengurutan menurun

Untuk melakukan pengurutan menurun digunakan DESC.

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date DESC;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
Zlotkey	SA_MAN	80	29-JAN-00
Mourgos	ST_MAN	50	16-NOV-99
Grant	SA_REP		24-MAY-99
Lorentz	IT_PROG	60	07-FEB-99
Vargas	ST_CLERK	50	09-JUL-98
Taylor	SA_REP	80	24-MAR-98
Matos	ST_CLERK	50	15-MAR-98
Fay	MK_REP	20	17-AUG-97
Davies	ST_CLERK	50	29-JAN-97

...  
20 rows selected.

### 3.16. Pengurutan dengan Kolom Alias

Kolom alias juga bisa digunakan dalam klausa ORDER BY.

```
SELECT employee_id, last_name, salary*12 annsal
FROM employees
ORDER BY annsal;
```

EMPLOYEE_ID	LAST_NAME	ANNSAL
144	Vargas	30000
143	Matos	31200
142	Davies	37200
141	Rajs	42000
107	Lorentz	50400
209	Whalen	52800
124	Mourgos	69600
104	Ernst	72000
202	Fay	72000
178	Grant	84000

...  
20 rows selected.

### 3.17. Pengurutan dengan Banyak Kolom

Pengurutan data bisa melibatkan dua kolom atau lebih.

```
SELECT last_name, department_id, salary
FROM employees
ORDER BY department_id, salary DESC;
```

LAST_NAME	DEPARTMENT_ID	SALARY
Whalen	10	4400
Hartstein	20	13000
Fay	20	6000
Mourgos	50	5800
Rajs	50	3500
Davies	50	3100
Matos	50	2500
Vargas	50	2500

...  
20 rows selected.

## 3.18. Latihan

1. Buat query untuk menampilkan nama dan gaji dari pegawai yang memiliki gaji lebih dari \$2850.

ENAME	SAL
JONES	2975
SCOTT	3000
KING	5000
FORD	3000

2. Buat query untuk menampilkan nama pegawai dan nomer departemen dari pegawai tersebut untuk pegawai yang mempunyai nomer pegawai = 7566.

ENAME	DEPTNO
JONES	20

3. Tampilkan nama dan gaji untuk semua pegawai yang memiliki gaji yang tidak berada dalam range antara \$1500 sampai \$2850.

ENAME	SAL
SMITH	800
WARD	1250
JONES	2975
MARTIN	1250
SCOTT	3000
KING	5000
ADAMS	1100
JAMES	950
FORD	3000
MILLER	1300

4. Tampilkan nama pegawai, pekerjaan dan tanggal mulai pegawai bekerja antara 20 February 1981 sampai 1 Mei 1981. Urutkan dengan urutan menaik pada kolom tgl mulai kerja.

ENAME	JOB	HIREDATE
ALLEN	SALESMAN	20-FEB-81
WARD	SALESMAN	22-FEB-81
JONES	MANAGER	02-APR-81
BLAKE	MANAGER	01-MAY-81

5. Tampilkan nama pegawai dan nomer department dari semua pegawai dalam department 10 dan department 30 dalam urutan menaik berdasarkan nama.

ENAME	DEPTNO
ALLEN	30
BLAKE	30
CLARK	10
JAMES	30
KING	10
MARTIN	30
MILLER	10
TURNER	30
WARD	30

6. Tampilkan nama dan gaji pegawai yang memiliki gaji lebih dari \$1500 dan bekerja pada nomer department 10 atau 30. Beri judul kolom 'Pegawai' dan 'Gaji Sebulan'.

PEGAWAI	GAJI SEBULAN
ALLEN	1600
BLAKE	2850
CLARK	2450
KING	5000



7. Tampilkan nama dan tanggal mulai kerja dari seorang pegawai yang mulai bekerja di tahun 1982.

ENAME	HIREDATE
MILLER	23-JAN-82

8. Tampilkan nama, dan pekerjaan pegawai yang tidak mempunyai manager (nilai kolom mgr = NULL)

ENAME	JOB
KING	PRESIDENT

9. Tampilkan nama, gaji dan komisi dari semua pegawai yang memiliki komisi (kolom komisi tidak sama dengan NULL). Urutkan berdasarkan gaji dan komisi, dengan urutan menurun.

ENAME	SAL	COMM
ALLEN	1600	300
TURNER	1500	0
MARTIN	1250	1400
WARD	1250	500

10. Tampilkan nama dari semua pegawai yang memiliki nama yang mengandung karakter 'A' pada huruf ketiga.

ENAME
BLAKE
CLARK
ADAMS

11. Tampilkan nama pegawai yang memiliki dua huruf 'L' pada namanya, dan bekerja di department 30, atau managernya = '7782'

ENAME
ALLEN
MILLER

12. Tampilkan nama, pekerjaan, dan gaji dari semua pegawai yang memiliki pekerjaan 'CLERK' atau 'ANALYST' dan gajinya tidak sama dengan \$1000, \$3000 atau \$5000.

ENAME	JOB	SAL
SMITH	CLERK	800
ADAMS	CLERK	1100
JAMES	CLERK	950
MILLER	CLERK	1300

13. Tampilkan nama, gaji, dan komisi dari semua pegawai yang memiliki jumlah komisi lebih dari jumlah gaji setelah dinaikkan 10%.

ENAME	SAL	COMM
MARTIN	1250	1400