

BAB 1 : Dasar Penulisan Statemen SQL

1.1 Sasaran

- Mengetahui kemampuan dari SQL SELECT Statement
- Dapat membuat perintah SQL dengan SELECT Statement

1.2 Dasar Statement SELECT

Sintak (cara penulisan) dari statement SELECT :

```
SELECT [DISTINCT] {*, column [alias], ... }
FROM table;
```

SELECT digunakan untuk memilih kolom yang ingin ditampilkan.

FROM digunakan untuk memilih table asal.

1.3 Memilih semua Kolom

```
SELECT *
FROM 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

8 rows selected.

1.4 Memilih Kolom yang Spesifik

```
SELECT department_id, location_id
FROM departments;
```

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
50	1500
60	1400
80	2500
90	1700
110	1700
190	1700

8 rows selected.

1.5 Aturan Penulisan Statement SQL

- SQL Statement tidak case sensitive artinya tidak dibedakan antara penulisan huruf kecil dan huruf besar.
- SQL Statement dapat terdiri dari lebih dari satu baris.
- Keyword tidak bisa disingkat atau dipisah di baris yang berbeda.

- Klausa biasanya ditempatkan pada baris yang berbeda.
- Inden digunakan untuk memudahkan pembacaan.

1.6 Ekspresi Aritmatik

Suatu ekspresi yang melibatkan tipe data bilangan (NUMBER) dan tanggal (DATE) menggunakan ekspresi aritmatika.

```
SELECT last_name, salary, salary + 300
FROM employees;
```

LAST_NAME	SALARY	SALARY+300
King	24000	24300
Kochhar	17000	17300
De Haan	17000	17300
Hunold	9000	9300
Ernst	6000	6300
...		
Hartstein	13000	13300
Fay	6000	6300
Higgins	12000	12300
Gietz	8300	8600

20 rows selected.

1.7 Operator Precedence

Perkalian dan pembagian memiliki prioritas (precedence) lebih tinggi daripada penambahan dan pengurangan.

```
SELECT last_name, salary, 12*salary+100
FROM employees;
```

LAST_NAME	SALARY	12*SALARY+100
King	24000	288100
Kochhar	17000	204100
De Haan	17000	204100
Hunold	9000	108100
Ernst	6000	72100
...		
Hartstein	13000	156100
Fay	6000	72100
Higgins	12000	144100
Gietz	8300	99700

20 rows selected.

1.8 Penggunaan Tanda Kurung

Penggunaan tanda kurung memiliki prioritas paling tinggi dibanding presedensi operator yang lain.

```
SELECT last_name, salary, 12*(salary+100)
FROM employees;
```

LAST_NAME	SALARY	12*(SALARY+100)
King	24000	289200
Kochhar	17000	205200
De Haan	17000	205200
Hunold	9000	109200
Ernst	6000	73200
...		
Hartstein	13000	157200
Fay	6000	73200
Higgins	12000	145200
Gietz	8300	100800

20 rows selected.

1.9 Mendefinisikan Nilai NULL

Nilai NULL (kosong) pada suatu kolom bisa berarti ada data yang tidak diisi, atau tidak diketahui nilainya. Nilai NULL tidak sama dengan NOL (zero). Nilai NULL juga tidak sama dengan spasi kosong.

```
SELECT last_name, job_id, salary, commission_pct
FROM employees;
```

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
King	AD_PRES	24000	
Kochhar	AD_VP	17000	
...			
Zlotkey	SA_MAN	10500	.2
Abel	SA_REP	11000	.3
Taylor	SA_REP	8600	.2
...			
Gietz	AC_ACCOUNT	8300	

20 rows selected.

Supaya nilai NULL tidak ditampilkan, digunakan fungsi NVL untuk memeriksa keberadaan nilai NULL.

Formatnya :

NVL(kolom yang di-evaluasi, diisi nilai ini jika kolom bernilai NULL)

1.10 Menggunakan Kolom Alias

Judul (secara default) pada tiap kolom yang ditampilkan (heading) selalu sama dengan nama kolomnya dan ditulis dengan huruf besar. Penggantian judul kolom, sehingga tidak sama dengan judul default disebut kolom alias.

```
SELECT last_name AS name, commission_pct comm
FROM employees;
```

NAME	COMM
King	
Kochhar	
De Haan	

...
20 rows selected.

```
SELECT last_name "Name", salary*12 "Annual Salary"
FROM employees;
```

Name	Annual Salary
King	288000
Kochhar	204000
De Haan	204000

...
20 rows selected.

1.11 Operator Penyambungan

Nilai dari dua kolom atau lebih dapat digabungkan dengan menggunakan operator penyambungan (||).

```
SELECT last_name || job_id AS "Employees"
FROM employees;
```

Employees
KingAD_PRE
KochharAD_VP
De HaanAD_VP
HunoldIT_PROG
ErnstIT_PROG
LorentzIT_PROG
MourgesST_MAN
RajsST_CLERK

...
20 rows selected.

1.12 Literal Character Strings

Literal dapat berupa karakter, ekspresi atau bilangan yang terdapat pada klausa SELECT. Literal berupa tanggal (DATE) dan karakter harus diapit dengan tanda petik tunggal.

```
SELECT last_name || ' is a ' || job_id
       AS "Employee Details"
FROM   employees;
```

Employee Details
King is a AD_PRES
Kochhar is a AD_VP
De Haan is a AD_VP
Hunold is a IT_PROG
Ernst is a IT_PROG
Lorentz is a IT_PROG
Mourgos is a ST_MAN
Rajs is a ST_CLERK

...
20 rows selected.

1.13 Baris yang Duplikat

Perintah query akan memberikan hasil termasuk suatu nilai yang bisa muncul lebih dari satu kali.

```
SELECT department_id
FROM   employees;
```

DEPARTMENT_ID
90
90
90
60
60
60
50
50
50

...
20 rows selected.

1.14 Menghapus Baris yang Duplikat

Untuk mengeliminasi baris ganda atau suatu nilai yang bisa muncul lebih dari satu kali digunakan keyword DISTINCT.

```
SELECT DISTINCT department_id
FROM   employees;
```

DEPARTMENT_ID
10
20
50
60
80
90
110

8 rows selected.

1.15 Menampilkan Struktur Tabel

Untuk menampilkan struktur table digunakan perintah DESCRIBE.

Formatnya :

DESC[RIBE] *namatabel*

DESCRIBE employees

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

1.16 Laporan Praktikum

1. Buatlah laporan praktikum dari langkah-langkah perintah diatas dan capture hasilnya
2. Latihan dibawah ini juga dimasukkan ke laporan praktikum

1.17 Latihan

1. Apakah statement SELECT berikut ini dapat dijalankan dengan benar ?
Benar/Salah

```
SQL> SELECT last_name,job_id,salary as sal
2 FROM EMPLOYEES;
```

2. Temukan 4 (empat) kesalahan pada statement SELECT berikut :

```
SQL> SELECT empno, ename
2 salary x 12 GAJI SETAHUN
3 FROM EMP;
```

3. Tampilkan struktur dari table departments, kemudian tampilkan semua datanya !
4. Tampilkan struktur dari table EMPLOYEES. Buat query untuk menampilkan nomer pegawai, nama, pekerjaan, dan tanggal mulai bekerja untuk tiap pegawai.

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

EMPNO	ENAME	JOB	HIREDATE
7369	SMITH	CLERK	17-DEC-80
7499	ALLEN	SALESMAN	20-FEB-81
7521	WARD	SALESMAN	22-FEB-81
7566	JONES	MANAGER	02-APR-81
7654	MARTIN	SALESMAN	28-SEP-81

5. Buat query untuk menampilkan pekerjaan secara unik dari table EMPLOYEES.

```

JOB
-----
ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN

```

6. Tampilkan nama pegawai digabung dengan pekerjaan dengan dipisah tanda koma, kemudian beri judul "Pegawai dan Pekerjaan"

Pegawai dan Pekerjaan

```

SMITH, CLERK
ALLEN, SALESMAN
WARD, SALESMAN
JONES, MANAGER
MARTIN, SALESMAN
BLAKE, MANAGER
CLARK, MANAGER
SCOTT, ANALYST

```

7. Buat query untuk menampilkan semua kolom dari table EMP. Semua kolom digabung jadi satu dengan tanda koma sebagai pemisah, kemudian beri judul "OUTPUT"

```

OUTPUT
-----
7369, SMITH, CLERK, 7902, 17-DEC-80, 800, , 20
7499, ALLEN, SALESMAN, 7698, 20-FEB-81, 1600, 300, 30
7521, WARD, SALESMAN, 7698, 22-FEB-81, 1250, 500, 30
7566, JONES, MANAGER, 7839, 02-APR-81, 2975, , 20
7654, MARTIN, SALESMAN, 7698, 28-SEP-81, 1250, 1400, 30

```