



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA YOGYAKARTA**

**MODUL 5
BAHASA QUERY**

Fungsi Konversi, Umum dan Seleksi

A. TUJUAN

- Mahasiswa dapat mengetahui dan menggunakan berbagai jenis fungsi yang tersedia di SQL.
- Mahasiswa dapat menggunakan fungsi konversi seperti `to_char`, `to_date` dan `to_number`.
- Mahasiswa dapat menggunakan fungsi umum yaitu `nvl`, `nvl2`, `nullif`, `coalesce`.
- Mahasiswa dapat memahami fungsi bersarang (nested)
- Mahasiswa dapat menggunakan fungsi untuk seleksi yaitu `decode` dan ekspresi `case`.

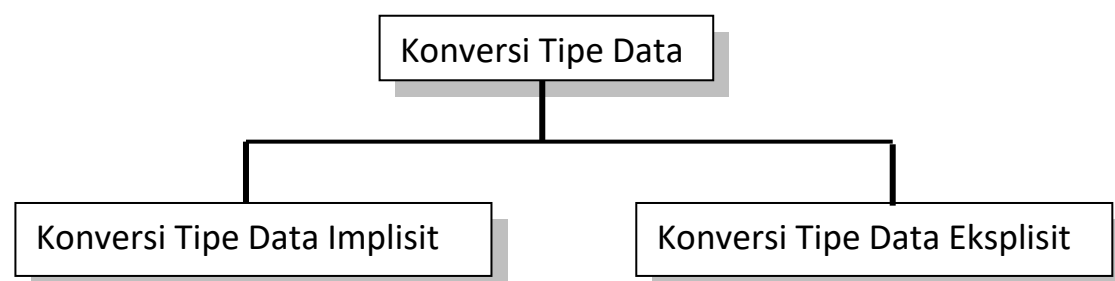
B. LANDASAN TEORI & LANGKAH PRAKTIKUM

Modul ini merupakan kelanjutan dari modul 4 sebelumnya yang membahas tentang fungsi karakter, bilangan maupun tanggal. Modul ini akan menjelaskan fungsi yang lain yaitu fungsi untuk kepentingan umum dan fungsi untuk melakukan konversi tipe data.

B.1. Fungsi Konversi

Fungsi konversi digunakan untuk melakukan konversi dari satu tipe data ke tipe data yang lain. Ada 2 jenis konversi tipe data yaitu konversi tipe data implisiti dan konversi tipe data eksplisit. Konversi tipe data implisit dilakukan secara otomatis oleh Oracle server sedangkan konversi tipe data eksplisit dilakukan oleh user. Konversi tipe data eksplisit dilakukan dengan menggunakan fungsi konversi.

Catatan : walaupun konversi tipe data implisit telah disediakan, sangat disarankan untuk menggunakan konversi tipe data eksplisit untuk menjaga realibility SQL statement.



Gambar 5.1. Pembagian Konversi Tipe Data

B.2. Konversi Tipe Data Implisit

Untuk *assignment*, Oracle server akan otomatis melakukan konversi seperti pada tabel 5.1.

Tabel 5.1. Untuk assignment, Konversi Tipe Data Implisit

Dari	Ke
varchar2 atau char	number
varchar2 atau char	Date
Number	Varchar2
Date	Varchar2

Sebagai contoh, ekspresi `hire_date > '01-Jan-90'` menghasilkan hasil dalam implisit konversi '01-Jan-90' sebagai sebuah string ke tipe data date.

Sedangkan untuk evaluasi ekspresi, konversi tipe data implisit seperti pada tabel 2

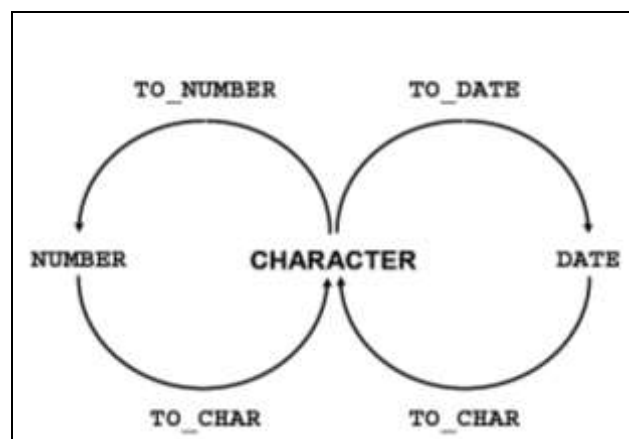
Tabel 5.2. Untuk evaluasi ekspresi, Konversi Tipe Data Implisit

Dari	Ke
varchar2 atau char	number
varchar2 atau char	Date

Secara umum, server Oracle menggunakan aturan untuk evaluasi ekspresi ketika konversi tipe data diperlukan dan tidak tercakup dalam aturan konversi untuk assignment. Sebagai contoh: `salary = '2000'` akan menghasilkan '2000' sebagai tipe data string yang merupakan konversi dari 2000 tipe data number.

B.3. Konversi Tipe Data Eksplisit

Konversi tipe data secara eksplisit dilakukan oleh user. Oracle menyediakan 3 fungsi konversi untuk melakukan konversi dari satu tipe data ke tipe data yang lain. Untuk membantu mempermudah pemahaman dapat dilihat pada gambar 5.2.

**Gambar 5.2. Fungsi Konversi, Copyright @ 2004, Oracle.**

Tabel 5.3 merupakan penjelasan ringkas tentang fungsi-fungsi konversi yang dapat digunakan.

Tabel 5.3. Tabel Fungsi Konversi

FUNGSI	TUJUAN
TO_CHAR(number date, [fmt], [nlsparams])	Mengkonversi nilai number atau date ke tipe data varchar2 dengan format string <i>fmt</i> (format model)
TO_NUMBER(char,[fmt],[nlsparams])	Mengkonversi nilai string yang mengandung angka ke tipe data numerik dengan format <i>fmt</i>
TO_DATE(char,[fmt],[nlsparams])	Mengkonversi nilai string yang mewakili tanggal ke tipe data date dengan menggunakan format <i>fmt</i> . Jika format <i>fmt</i> tidak diberikan maka format date yang digunakan adalah DD-MON-YY

B.3.1. Penggunaan Fungsi TO_CHAR dengan DATE

Format penggunaan fungsi to_char dengan date adalah sebagai berikut :

`to_char (date,'format model')`

Format model :

- Diantara tanda petik tunggal (' ');
- Case sensitive
- Dapat berisi elemen tanggal yang valid
- Memiliki elemen *fm* untuk menghilangkan spasi kosong atau menyembunyikan angka nol.

Elemen dari model format tanggal seperti pada tabel 4. Tabel 5 model format tanggal untuk waktu.

Catatan : Untuk Elemen dari model format tanggal dan waktu secara lengkap dapat dilihat di Oracle Reference.

Tabel 5.4. Elemen dari Model Format Tanggal

ELEMEN	HASIL
YYYY	Full Year dalam Number
YEAR	Year dalam bahasa Inggris
MM	Dua Digit nilai untuk bulan
MONTH	Nama lengkap dari bulan
MON	Tiga huruf singkatan dari bulan
DY	Tiga huruf singkatan dari hari dalam satu minggu
DAY	Nama lengkap dari hari dalam satu minggu
DD	Nilai numerik hari dalam satu bulan

Tabel 5.5. Elemen dari Model Format Tanggal (Format Jam)

ELEMEN	HASIL
HH24:MI:SS AM	Format elemen jam sebagai bagian dari tanggal. Contoh : 15:45:32 AM
DD "of" MONTH	Menambahkan karakter string dengan menggunakan pembatas petik ganda ("") Contoh : 12 of OCTOBER
ddsph	Jumlah angka dibaca dalam bahasa Inggris Contoh : fourteenth

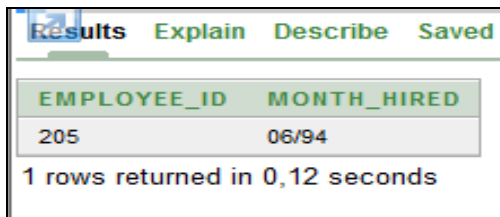
Berikut contoh-contoh penggunaan model format tanggal.

Contoh 1

- Menampilkan hire_date dengan format MM/YY dari employee yang last_namenya 'Higgins'

```
SELECT employee_id, TO_CHAR(hire_date, 'MM/YY') Month_Hired
FROM employees
WHERE last_name='Higgins';
```

Hasil :



EMPLOYEE_ID	MONTH_HIRED
205	06/94

1 rows returned in 0,12 seconds

Contoh 2:

- Menampilkan last_name dan hire_date dengan format model tanggal DD MONTH YYYY untuk seluruh employee. Catatan : Format Month tergantung dari setting server oracle, dalam format Indonesia atau Inggris.

```
SELECT last_name, TO_CHAR(hire_date, 'fmDD MONTH YYYY')
AS HIREDATE
FROM employees;
```

Hasil :

Results Explain Describe Saved SQL History	
LAST_NAME	HIREDATE
King	17 JUNI 1987
Kochhar	21 SEPTEMBER 1989
De Haan	13 JANUARI 1993
Hunold	3 JANUARI 1990
Ernst	21 MEI 1991
Austin	25 JUNI 1997
Pataballa	5 FEBRUARI 1998
Lorentz	7 FEBRUARI 1999
Greenberg	17 AGUSTUS 1994
Faviet	16 AGUSTUS 1994
More than 10 rows available. Increase rows selector to view more rows.	
10 rows returned in 0,02 seconds CSV Export	

Contoh 3:

- Modifikasi contoh 2 sehingga ditampilkan dengan format "Seventeenth of June 1987 12:00:00 AM"

```
SELECT last_name,TO_CHAR(hire_date,'fmDdspth "of" MONTH YYYY
fmHH:MI:SS AM') AS HIREDATE
FROM employees;
```

asil :

Results Explain Describe Saved SQL History	
LAST_NAME	HIREDATE
King	Seventeenth of JUNI 1987 12:00:00 PAGI
Kochhar	Twenty-First of SEPTEMBER 1989 12:00:00 PAGI
De Haan	Thirteenth of JANUARI 1993 12:00:00 PAGI
Hunold	Third of JANUARI 1990 12:00:00 PAGI
Ernst	Twenty-First of MEI 1991 12:00:00 PAGI
Austin	Twenty-Fifth of JUNI 1997 12:00:00 PAGI
Pataballa	Fifth of FEBRUARI 1998 12:00:00 PAGI
Lorentz	Seventh of FEBRUARI 1999 12:00:00 PAGI
Greenberg	Seventeenth of AGUSTUS 1994 12:00:00 PAGI
Faviet	Sixteenth of AGUSTUS 1994 12:00:00 PAGI
More than 10 rows available. Increase rows selector to view more rows.	
10 rows returned in 0,00 seconds CSV Export	

B.3.2. Penggunaan Fungsi TO_CHAR dengan NUMBER

Format umum :

`to_char (number,'format_model')`

Beberapa elemen untuk memformat number yang dapat digunakan dengan fungsi `to_char` dapat dilihat pada tabel 6.

Tabel 6. Elemen Untuk Format Number

Elemen	KEGUNAAN	CONTOH	HASIL CTH
9	Mewakili Angka (Jumlah angka 9 mewakili panjang numerik)	99999	12345
0	Memaksa angka 0 untuk ditampilkan diawal	099999	001234
\$	Menampilkan \$	\$99999	\$1234
L	Penggunaan Simbol Mata Uang Lokal	L99999	Rp1234
.	Menampilkan desimal point	99999.99	1234.00
,	Menampilkan koma sebagai pemisah ribuan	9,999	1,234

Contoh 4

- Menampilkan `last_name` dan `salary` dengan menggunakan elemen format number untuk employee yang `last_name`nya diawali huruf K;

```
SELECT last_name, TO_CHAR(salary, '$99,999.00') as SALARY
FROM employees where last_name like 'K%';
```

Hasil:

LAST_NAME	SALARY
Kaufing	\$7,900.00
Khoo	\$3,100.00
King	\$10,000.00
King	\$24,000.00
Kochhar	\$17,000.00
Kumar	\$6,100.00

B.3.3. Penggunaan Fungsi TO_NUMBER dan TO_DATE

Untuk mengkonversi string ke dalam format number menggunakan fungsi `TO_NUMBER`, yang memiliki format :

```
to_number(char, 'format_model')
```

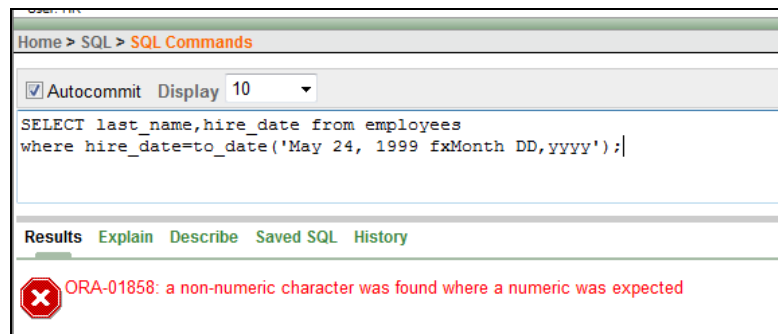
Untuk mengkonversi string ke dalam format tanggal menggunakan fungsi `TO_DATE`, yang memiliki format :

```
to_date(char, 'format_model')
```

Fungsi ini menggunakan fx *modifier*, oleh karenanya karakter dan format model date yang dipilih harus tepat.

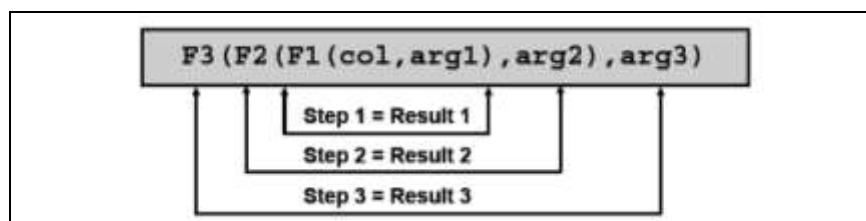
Contoh 5

Menampilkan employee yang hire_date nya dimulai dengan May 24,1999. Karena menggunakan fx modifier terdapat kesalahan pada contoh 5. Perbaiki kueri contoh 5 tersebut agar menghasilkan hasil yang benar.



B.4. Fungsi Bersarang (Nesting)

Fungsi nesting memungkinkan hasil dari suatu fungsi untuk dikirim sebagai argumen untuk fungsi yang lain. Fungsi nesting akan dievaluasi dari yang paling dalam terlebih dahulu baru ke yang paling luar, seperti ilustrasi pada gambar 3.



Gambar 3. Ilustrasi Fungsi Nesting (Oracle@2004)

Contoh 6

- Menampilkan last_name dari employee yang berada di department_id =60.

```
SELECT last_name,
UPPER(CONCAT(SUBSTR(LAST_NAME,1,8), '_US'))
FROM employees
WHERE department_id=60;
```

Hasil:

Results Explain Describe Saved SQL History	
LAST_NAME	UPPER(CONCAT(SUBSTR(LAST_NAME,1,8),'_US'))
Hunold	HUNOLD_US
Ernst	ERNST_US
Austin	AUSTIN_US
Pataballa	PATABALL_US
Lorentz	LORENTZ_US

5 rows returned in 0,01 seconds [CSV Export](#)

Penjelasan:

Kueri diatas akan mengevaluasi 3 kali, yaitu:

1. Fungsi yang paling dalam yang akan dievaluasi pertama kali yaitu
RESULT1 = SUBSTR(LAST_NAME,1,8)
2. Hasilnya akan DI CONCAT dengan string 'US'
RESULT2 = CONCAT(RESULT1,'US')
3. Dan yang terakhir menjadikan huruf besar dengan fungsi UPPER
RESULT3 = UPPER(RESULT2)

B.5. FUNGSI UMUM (GENERAL)

Fungsi umum dapat bekerja pada sembarang tipe data, begitu juga dengan null. Beberapa fungsi umum dapat dilihat pada tabel 7.

Tabel 7. Fungsi Umum

Fungsi	Tujuan
NVL(expr1,expr2)	Mengkonversi nilai null ke nilai aktual. Dapat untuk tipe data apa saja , akan tetapi nilai aktual yang dikembalikan harus sama tipenya dengan expr1.
NVL2(expr1,expr2,expr3)	Jika expr1 not null, NVL2 akan mengembalikan expr2. Jika expr1 null, NVL2 akan mengembalikan expr3. Expr1 dapat berbagai tipe data
NULLIF(expr1,expr2)	Membandingkan dua ekspresi, mengembalikan null jika sama dan anakan mengembalikan expr1 jika tidak sama.
COALESCE(expr1,expr2,... exprn)	Mengembalikan ekspresi pertama yang tidak null dari daftar.

Berikut contoh-contoh penggunaan fungsi general

Contoh 7

- Penggunaan Fungsi NVL

```
SELECT last_name,salary,
NVL(commission_pct,0),
(salary*12)+(salary*12*nvl(commission_pct,0)) as ANNSAL
FROM employees;
```

Hasil

LAST_NAME	SALARY	NVL(COMMISSION_PCT,0)	ANNSAL
King	24000	0	288000
Kochhar	17000	0	204000
De Haan	17000	0	204000
Hunold	9000	0	108000
Ernst	6000	0	72000
Austin	4800	0	57600
Pataballa	4800	0	57600
Lorentz	4200	0	50400
Greenberg	12000	0	144000
Faviet	9000	0	108000
More than 10 rows available. Increase rows selector to view more rows.			
10 rows returned in 0,01 seconds CSV Export			

Penjelasan:

Perhatikan bahwa ANNSAL dihitung dari gaji pokok selama 12 bulan ditambah dengan commission_pct yang diperolehnya. Jika tidak menggunakan fungsi NVL maka nilai ANNSAL untuk employee yang tidak memiliki commission_pct adalah NULL.

Contoh 8

- Penggunaan Fungsi NVL2

```
SELECT last_name,salary,commission_pct,
NVL2(commission_pct,'SAL+COMM','SAL') income
FROM employees
where department_id in (50,80)
order by last name;
```

Hasil:

LAST_NAME	SALARY	COMMISSION_PCT	INCOME
Abel	11000	,3	SAL+COMM
Ande	6400	,1	SAL+COMM
Atkinson	2800	-	SAL
Banda	6200	,1	SAL+COMM
Bates	7300	,15	SAL+COMM
Bell	4000	-	SAL
Bernstein	9500	,25	SAL+COMM
Bissot	3300	-	SAL
Bloom	10000	,2	SAL+COMM
Bull	4100	-	SAL
More than 10 rows available. Increase rows selector to view more rows.			

Penjelasan:

Jika commission_pct tidak null maka akan dikembalikan hasil 'SAL+COMM' sedangkan commission_pct null maka akan dikembalikan 'SAL'

Contoh 9

- Penggunaan Fungsi NULLIF

```
SELECT first_name, LENGTH(first_name)
"expr1", last_name, LENGTH(last_name) "expr2",
NULLIF(LENGTH(first_name), LENGTH(last_name)) result
FROM employees order by first name
```

Hasil:

FIRST_NAME	Expr1	LAST_NAME	Expr2	RESULT
Adam	4	Fripp	5	4
Alana	5	Walsh	5	-
Alberto	7	Errazuriz	9	7
Alexander	9	Khoo	4	9
Alexander	9	Hunold	6	9
Alexis	6	Bull	4	6
Allan	5	McEwen	6	5
Alyssa	6	Hutton	6	-
Amit	4	Banda	5	4
Anthony	7	Cabrio	6	7

More than 10 rows available. Increase rows selector to view more rows.

Dilakukan pengecekan antara expr1 dan expr2, jika expr1 sama dengan expr2 maka akan mengembalikan result null (Lihat Aiana), sedangkan jika expr1 tidak sama dengan expr2 akan mengembalikan nilai expr1 (Lihat Adam).

NULLIF secara logika sama dengan ekspresi CASE, yaitu

```
CASE WHEN expr1=expr2 THEN NULL ELSE expr1 END
```

Contoh 10

- Penggunaan Fungsi COALESCE

```
SELECT last_name,
COALESCE(manager_id, commission_pct, -1) comm
FROM employees
ORDER BY commission_pct
```

Hasil:

Cambrault	100
Doran	146
King	146
Sully	146
McEwen	146
Russell	100
King	-1
Kochhar	100
De Haan	100
Hunold	102

Perhatikan, jika Manager_id not null maka manager_id akan ditampilkan, kemudian dicek jika commission_pct not null maka commission_pct akan ditampilkan, jika manager_id dan commission_pct nya null maka akan ditampilkan -1 (Lihat King).

B.6. Ekspresi Kondisi

Logika IF-THEN-ELSE dapat digunakan dalam perintah SQL dengan menggunakan 2 metode yaitu dengan ekspresi CASE dan fungsi DECODE.

B.6.1 Ekspresi CASE

Sintak :

```
CASE expr WHEN comparison_expr1 THEN return_expr1
      [WHEN comparison_expr2 THEN return_expr2
      WHEN comparison_exprn THEN return_exprn
      ELSE else_expr]
END
```

- *Expr* akan diuji, jika sama dengan *comparison_expr1* maka akan dikembalikan *return_expr1*, jika tidak akan membandingkan dengan *comparison_expr2* dst, dan akan mengembalikan *else_expr* jika ELSE diberikan jika tidak akan mengembalikan nilai NULL. Nilai NULL otomatis akan diberikan jika kondisi tidak terpenuhi tidak perlu memberikan literal NULL pada *return_expr* maupun *else_expr*.
- Seluruh *expr* dan *comparison_expr* harus memiliki tipe data yang sama yaitu salah satu dari CHAR, VARCHAR2, NCHAR, NVARCHAR2.

Contoh 11

- Akan diadakan perubahan salary untuk IT_PROG, ST_CLEARK dan SA_REP.

```
SELECT last_name, job_id, salary,
CASE job_id WHEN 'IT_PROG' THEN 1.10*salary
            WHEN 'ST_CLERK' THEN 1.15*salary
            WHEN 'SA_REP' THEN 1.20*salary
ELSE salary END "REVISED_SALARY"
FROM employees;
```

Hasil :

LAST_NAME	JOB_ID	SALARY	REVISED_SALARY
King	AD_PRES	24000	24000
Kochhar	AD_VP	17000	17000
De Haan	AD_VP	17000	17000
Hunold	IT_PROG	9000	9900
Ernst	IT_PROG	6000	6600
Austin	IT_PROG	4800	5280
Pataballa	IT_PROG	4800	5280
Lorentz	IT_PROG	4200	4620
Greenberg	FI_MGR	12000	12000
Faviet	FI_ACCOUNT	9000	9000
More than 10 rows available. Increase rows selector to view more rows.			

B.6.2 Fungsi DECODE

Sintak :

```
DECODE(col|expression, search1,result1
      [,search2, result2,.....,]
      [,default])
```

- Expression akan diuji untuk setiap nilai *search* dan akan mengembalikan *result* sesuai dengan nilai *search*.
- Jika nilai default tidak diberikan maka akan mengembalikan nilai null jika tidak akan kondisi yang sesuai.

Contoh 12

- Seperti contoh 11 dengan menggunakan fungsi DECODE

```
SELECT last_name, job_id, salary,
       DECODE(job_id, 'IT_PROG', 1.10*salary,
                'ST_CLERK', 1.15*salary,
                'SA_REP', 1.20*salary,
                salary) "REVISED_SALARY"
FROM employees;
```

H

Hasil :

LAST_NAME	JOB_ID	SALARY	REVISED_SALARY
King	AD_PRES	24000	24000
Kochhar	AD_VP	17000	17000
De Haan	AD_VP	17000	17000
Hunold	IT_PROG	9000	9900
Ernst	IT_PROG	6000	6600
Austin	IT_PROG	4800	5280
Pataballa	IT_PROG	4800	5280
Lorentz	IT_PROG	4200	4620
Greenberg	FI_MGR	12000	12000
Faviet	FI_ACCOUNT	9000	9000
More than 10 rows available. Increase rows selector to view more rows.			

C. TUGAS

Catatan : Untuk setiap nomor simpanlah sebagai sebuah file dengan format lab5_xx (dimana xx adalah nomor tugas).

1. Buatlah kueri yang menghasilkan format di bawah ini untuk semua employee
<employee last name> mendapat <salary> perbulan tetapi menginginkan <3 kali salary>. Beri nama kolom DREAM SALARIES

DREAM SALARIES
King mendapat \$24,000.00 perbulan tetapi menginginkan \$72,000.00.
Kochhar mendapat \$17,000.00 perbulan tetapi menginginkan \$51,000.00.
De Haan mendapat \$17,000.00 perbulan tetapi menginginkan \$51,000.00.
Hunold mendapat \$9,000.00 perbulan tetapi menginginkan \$27,000.00.
Ernst mendapat \$6,000.00 perbulan tetapi menginginkan \$18,000.00.
Austin mendapat \$4,800.00 perbulan tetapi menginginkan \$14,400.00.
Pataballa mendapat \$4,800.00 perbulan tetapi menginginkan \$14,400.00.
Lorentz mendapat \$4,200.00 perbulan tetapi menginginkan \$12,600.00.
Greenberg mendapat \$12,000.00 perbulan tetapi menginginkan \$36,000.00.
Faviet mendapat \$9,000.00 perbulan tetapi menginginkan \$27,000.00.
More than 10 rows available. Increase rows selector to view more rows.

2. Buatlah kueri untuk menampilkan last_name,hire_date dan tanggal untuk melakukan review salary. Dimana tanggal review diperoleh dari hari Monday/Senin pertama setelah enam bulan hire_date. Beri nama kolom REVIEW. Format tanggal yang diinginkan seperti "Monday, the Thirty-First of July, 2000".

LAST_NAME	HIRE_DATE	REVIEW
King	17-06-1987	Senin, the Twenty-First of Desember, 1987
Kochhar	21-09-1989	Senin, the Twenty-Sixth of Maret, 1990
De Haan	13-01-1993	Senin, the Nineteenth of Juli, 1993
Hunold	03-01-1990	Senin, the Ninth of Juli, 1990
Ernst	21-05-1991	Senin, the Twenty-Fifth of Nopember, 1991
Austin	25-06-1997	Senin, the Twenty-Ninth of Desember, 1997
Pataballa	05-02-1998	Senin, the Tenth of Agustus, 1998
Lorentz	07-02-1999	Senin, the Ninth of Agustus, 1999
Greenberg	17-08-1994	Senin, the Twentieth of Februari, 1995
Faviet	16-08-1994	Senin, the Twentieth of Februari, 1995
More than 10 rows available. Increase rows selector to view more rows.		

3. Tampilkan last_name,hire_date dan nama hari pertama dalam minggu employee mulai bekerja. Beri nama kolom ini DAY. Urutkan berdasarkan nama hari mulai dari SUNDAY (Minggu)

Johnson	04-01-2000	SELASA
Mavris	07-06-1994	SELASA
Baer	07-06-1994	SELASA
Higgins	07-06-1994	SELASA
Faviet	16-08-1994	SELASA
King	17-06-1987	RABU
De Haan	13-01-1993	RABU
Hunold	03-01-1990	RABU
Austin	25-06-1997	RABU

4. Buatlah kueri untuk menampilkan commission_pct dari semua employee. Jika employee tidak mempunyai commission_pct tampilkan 'No Commission'
5. Buatlah kueri dengan menggunakan fungsi DECODE untuk menampilkan last_name, job_id dan grade dari semua employee berdasarkan nilai dari JOB_ID berdasarkan tabel berikut :

Job	GRADE
AD_PRES	A
ST_MAN	B
IT_PROG	C
SA_REP	D
ST_CLERK	E
Selain itu	0

LAST_NAME	JOB_ID	GRADE
King	AD_PRES	A
Kochhar	AD_VP	0
De Haan	AD_VP	0
Hunold	IT_PROG	C
Ernst	IT_PROG	C
Austin	IT_PROG	C
Pataballa	IT_PROG	C
Lorentz	IT_PROG	C
Greenberg	FI_MGR	0
Faviet	FI_ACCOUNT	0

6. Ubahlah tugas no 5 dengan menggunakan ekspresi CASE

D. DAFTAR PUSTAKA

1. Oracle Database 10g : SQL Fundamental I, Oracle Inc. 2004

😊 Bertekun sampai akhir 😊