

Pertemuan 12

Function

Tujuan Intruksional:

Pokok bahasan ini mempelajari tentang fungsi-fungsi dalam APEX

Kompetensi Yang Diharapkan:

Mahasiswa diharapkan memahami tentang fungsi-fungsi dalam APEX

Waktu Pertemuan : 100 Menit

3.3 Pengenalan Fungsi

Fungsi adalah program yang melakukan tindakan pada nilai atau kolom dan menghasilkan sesuatu yang berbeda sebagai output.

Fungsi memiliki input dan output. Masukan ke fungsi disebut sebagai argumen.

Input (Argumen)	Function	Output
--------------------	----------	--------

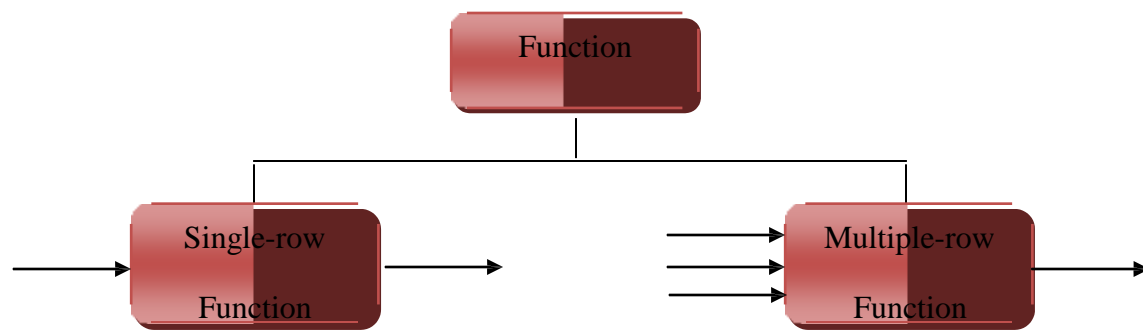
Contoh :

Pada *drink machine*, inputnya adalah uang dan hasilnya adalah minuman.



Oracle memiliki dua jenis fungsi yang berbeda, yaitu :

- Single-row Function
- Multiple-row Function



- Single-row Function beroperasi pada satu baris saja dan mengembalikan satu hasil per baris.
- Multiple-row Function dapat memanipulasi sekelompok baris untuk memberi satu hasil per kelompok baris. Fungsi ini juga dikenal sebagai Group Function.

Single-Row Function

Di dalam SQL, single-row function dapat digunakan untuk :

- Melakukan perhitungan seperti pembulatan angka ke desimal tertentu
- Memodifikasi item data individual seperti mengubah nilai karakter dari huruf besar ke huruf kecil
- Mengubah format tanggal dan angka untuk tampilan seperti mengubah format tanggal database numerik internal menjadi format standar
- Mengkonversi kolom data jenis seperti mengubah karakter string ke nomor atau tanggal

Multiple-Row Function

Multiple-Row Function (Group Function) menerima banyak baris sebagai masukan, dan mengembalikan satu nilai sebagai output.

Contoh Multiple-Row Function (Group Function) meliputi :

- MAX: menemukan nilai tertinggi dalam kelompok baris
- MIN : menemukan nilai terendah dalam kelompok baris
- AVG: menemukan nilai rata-rata dalam kelompok baris

4.1 Case and Character Manipulation

Dual Table

DUMMY
X

Dual Table memiliki satu baris yang disebut "X" dan satu kolom bernama "DUMMY." Dual Table digunakan untuk membuat pernyataan SELECT dan menjalankan fungsi yang tidak terkait langsung dengan tabel database tertentu.

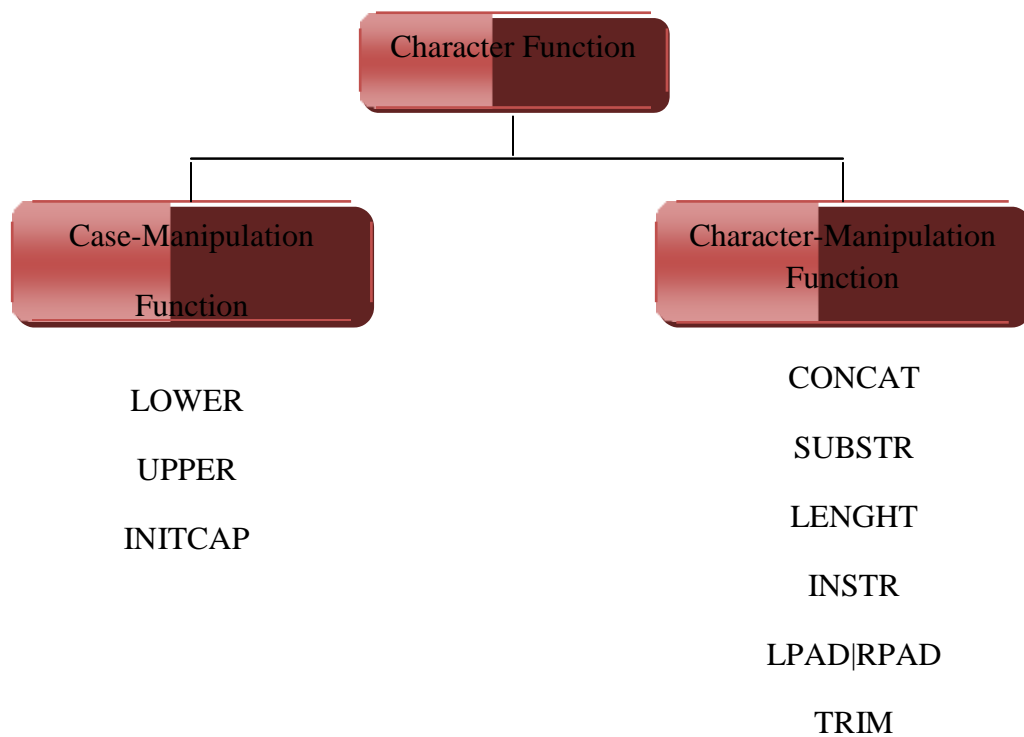
Contoh :

SELECT (319/29) + 12 FROM DUAL;	(319/29)+12 23
------------------------------------	-------------------

Single-Row Character Function

Single-row Character Function dibagi menjadi dua kategori:

- Fungsi yang mengkonversikan case dari karakter string
- Fungsi yang dapat berupa join, extract, show, find, pad, dan trim karakter string



Case Manipulation Function

- Case manipulation function digunakan untuk mengubah data dari state yang disimpan dalam tabel menjadi lower, upper, atau mixed case.
- Konversi ini dapat digunakan untuk memformat output dan juga dapat digunakan untuk mencari string tertentu.
- Case manipulation function dapat digunakan pada hampir semua bagian sql statement

Case manipulation function sering kali berguna saat mencari data tetapi tidak tahu apakah data yang dicari ada di huruf besar atau kecil. Dari sudut pandang database, 'V' dan 'v' BUKAN karakter yang sama dan, karena itu, perlu dicari dengan menggunakan case yang benar.

- LOWER (column | expression) mengubah karakter alfa menjadi huruf kecil.

Query :

```
SELECT last_name  
FROM employees  
WHERE LOWER(last_name) = 'abel';
```

Output :

LAST_NAME
Abel

- UPPER (column | expression) mengubah karakter alfa menjadi huruf besar.

Query :

```
SELECT last_name  
FROM employees  
WHERE UPPER(last_name) = 'ABEL';
```

Output :

LAST_NAME
Abel

- INITCAP(column|expression) mengubah karakter alfa menjadi huruf besar pada setiap awal kata.

Query :

```
SELECT last_name  
FROM employees  
WHERE INITCAP(last_name) = 'Abel';
```

Output :

LAST_NAME
Abel

Character Manipulation Functions

Character manipulation function digunakan untuk mengekstrak, mengubah, memformat, atau mengubah beberapa string karakter.

CONCAT : untuk menggabungkan dua nilai.

Mengambil dua argumen karakter string, dan menggabungkan string kedua dengan yang pertama. Dapat juga ditulis menggunakan concatenation operator (||).

Query :

```
SELECT CONCAT('Hello',  
'World')  
FROM DUAL;
```

Output :

CONCAT('HELLO','WORLD')
helloworld

Query :

```
SELECT CONCAT(first_name,  
last_name)  
FROM employees;
```

Output :

CONCAT(FIRST_NAME, LAST_NAME)
EllenAbel
CurtisDavies
LexDe Haan
BruceErnst
PatFay
WilliamGietz
KimberelyGrant
MichaelHartstein
ShelleyHiggins
AlexanderHunold
More than 10 rows available. Increase rows selector to view more rows.

SUBSTR : untuk mengekstrak string dengan panjang yang ditentukan.

Argumennya adalah (character String, starting position, length). Length argument bersifat opsional, dan jika dihilangkan, akan mengembalikan semua karakter ke akhir string.

Query :

```
SELECT SUBSTR('HelloWorld',1,5)  
FROM DUAL;
```

Output :

SUBSTR('HELLOWORLD',1,5)
Hello

Query :

```
SELECT SUBSTR('HelloWorld', 6)
FROM DUAL;
```

Output :

SUBSTR('HELLOWORLD',6)
World

Query :

```
SELECT SUBSTR(last_name,1,3)
FROM employees;
```

Output :

SUBSTR(LAST_NAME,1,3)
Abe
Dav
De
Ern
Fay
Gie
Gra
Har
Hig
Hun
More than 10 rows available. Increase rows selector to view more rows.

LENGTH : untuk menampilkan panjang string dengan nilai angka.

Fungsi ini mengambil karakter string sebagai argumen dan mengembalikan angka yang merupakan panjang dari karakter string tersebut.

Query :

```
SELECT LENGTH('HelloWorld')
FROM DUAL;
```

Output :

LENGTH('HELLOWORLD')
10

Query :

```
SELECT LENGTH(last_name)
FROM employees;
```

Output :

LENGTH(LAST_NAME)
4
6
7
5
3
5
5
9
7
6
More than 10 rows available. Increase rows selector to view more rows.

INSTR : untuk menemukan posisi numerik dari karakter tertentu.

INSTR mencari substring pertama dalam string karakter dan mengembalikan posisinya sebagai angka. Jika substring tidak ditemukan, angka nol akan dikembalikan.

Query :

```
SELECT INSTR('HelloWorld', 'W')
FROM DUAL;
```

Output :

INSTR('HELLOWORLD','W')
6

Query :

```
SELECT last_name, INSTR(last_name, 'a')
FROM employees;
```

Output :

LAST_NAME	INSTR(LAST_NAME,'A')
Abel	0
Davies	2
De Haan	5
Ernst	0
Fay	2
Gietz	0
Grant	3
Hartstein	2
Higgins	0
Hunold	0
More than 10 rows available. Increase rows selector to view more rows.	

LPAD : Pads pada sisi kiri karakter string, menghasilkan right-justified.

LPAD membutuhkan 3 argumen: string karakter, jumlah karakter dalam padded string, dan karakter untuk pad.

Query :

```
SELECT LPAD('HelloWorld',15, '-')
FROM DUAL;
```

Output :

LPAD('HELLOWORLD',15,'-')
-----HelloWorld

Query :

```
SELECT LPAD(last_name, 10, '*')  
FROM employees;
```

Output :

LPAD(LAST_NAME,10,'*')
*****Abel
****Davies
***De Haan
*****Ernst
*****Fay
*****Gietz
*****Grant
*Hartstein
***Higgins
****Hunold
More than 10 rows available. Increase rows selector to view more rows.

RPAD : Pads pada sisi kanan karakter string, menghasilkan left-justified.

Query :

```
SELECT RPAD('HelloWorld',15, '-')  
FROM DUAL;
```

Output :

RPAD('HELLOWORLD',15,'-')
HelloWorld-----

Query :

```
SELECT RPAD(last_name, 10, '*')  
FROM employees;
```

Output :

RPAD(LAST_NAME,10,'*')
Abel*****
Davies****
De Haan***
Ernst*****
Fay*****
Gietz*****
Grant*****
Hartstein*
Higgins***
Hunold****
More than 10 rows available. Increase rows selector to view more rows.

TRIM : untuk menghapus semua karakter spesifik dari awal, akhir, atau awal dan akhir string.

Query :

```
SELECT TRIM(LEADING 'a' FROM 'abcba')  
FROM DUAL;
```

Output :

TRIM(LEADING'A'FROM'ABCBA')
bcba

Query :

```
SELECT TRIM(TRAILING 'a' FROM 'abcba')  
FROM DUAL;
```

Output :

TRIM(TRAILING'A'FROM'ABCBA')
abcb

Query :

```
SELECT TRIM(BOTH 'a' FROM 'abcba')  
FROM DUAL;
```

Output :

TRIM(BOTH'A'FROM'ABCBA')
bcb

REPLACE : untuk menggantikan urutan karakter dalam string dengan kumpulan karakter lainnya.

```
REPLACE (string1, string_to_replace, [replacement_string] )
```

- string1 adalah string yang akan memiliki karakter pengganti di dalamnya
- string_to_replace adalah string yang akan dicari dan diambil dari string1
- [replacement_string] adalah string baru yang akan dimasukkan ke dalam string1

Query :

```
SELECT REPLACE('JACK and JUE','J','BL')  
FROM DUAL;
```

Output :

REPLACE('JACKANDJUE','J','BL')
BLACK and BLUE

Query :

```
SELECT REPLACE('JACK and JUE','J')  
FROM DUAL;
```

Output :

REPLACE('JACKANDJUE','J')
ACK and UE

Query :

```
SELECT REPLACE(last_name,'a','*')
FROM employees;
```

Output :

REPLACE(LAST_NAME,'A','*')
Abel
D*vies
De H**n
Ernst
F*y
Gietz
Gr*nt
H*rtstein
Higgins
Hunold
More than 10 rows available. Increase rows selector to view more rows.

Column Aliases

Kolom alias digunakan untuk memberi nama sebuah fungsi. Bila kolom alias digunakan, kolom alias hanya akan muncul di output dan bukan pada sintaks fungsi sebenarnya.

Contoh :

```
SELECT LOWER(last_name) || LOWER(SUBSTR(first_name,1,1))
AS "User Name"
FROM employees;
```

User Name
abele
daviesc
de haanl
ernstb
fayp
gietzw
grantk
hartsteinm
higginss
hunolda
More than 10 rows available. Increase rows selector to view more rows.

Ketika dicoba untuk kedua kalinya tanpa menggunakan kolom alias, nama kolom akan kembali seperti semula.

LOWER(LAST_NAME) LOWER(SUBSTR(FIRST_NAME,1,1))
abele
daviesc
de haanl
ernstb
fayp
gietzw
grantk
hartsteinm
higginss
hunolda
More than 10 rows available. Increase rows selector to view more rows.

Substitution Variables

Substitution variables digunakan untuk menjalankan query yang sama dengan banyak nilai berbeda untuk mendapatkan set hasil yang berbeda.

Oracle Application Express mendukung variabel substitusi. Untuk menggunakannya, nilai hardcoded dalam pernyataan harus diganti dengan “: named_variable”.

Misalnya ini adalah query aslinya :

```
SELECT first_name, last_name, salary, department_id
FROM employees
WHERE department_id= 10;
```

Lalu ingin menjalankan dengan nilai yang berbeda (20, 30, 40, ... dsb).

Maka dapat ditulis dengan :

```
SELECT first_name, last_name, salary, department_id
FROM employees
WHERE department_id=:enter_dept_id;
```

Catatan : gunakan simbol titik dua (“ : ”) di depan ‘enter_dept_id. Ini adalah simbol yang membuat Oracle Application Express mengenali teks sebagai variabel.

:ENTER_DEPT_ID

Submit

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
Jennifer	Whalen	4400	10

4.2 Number Function

Terdapat 3 jenis fungsi angka dalam SQL query, yaitu:

1. Round
2. Trunc
3. Mod

1. ROUND

ROUND dapat digunakan untuk membulatkan angka ke jumlah tertentu dari nilai decimal.

Query:

```
ROUND(column|expression, decimal places)
```

2. TRUNC

TRUNC digunakan untuk mengakhiri kolom, ekspresi, atau nilai ke angka tertentu dari nilai decimal.

Query:

```
TRUNC(column|expression, decimal places)
```

- TRUNC tidak membulatkan angka, tetapi mengakhiri angka pada titik tertentu.

Query:

```
SELECT MIN_salary, Max_salary, MIN_salary / Max_salary AS "Koma",  
ROUND(MIN_salary/Max_salary,2), TRUNC(MIN_salary/Max_salary,2)  
FROM JOBS  
WHERE max_salary=8500
```

Output:

MIN_SALARY	MAX_SALARY	Koma	ROUND(MIN_SALARY/MAX_SALARY,2)	TRUNC(MIN_SALARY/MAX_SALARY,2)
5500	8500	.647058823529411764705882352941176470588	.65	.64

3. MOD

MOD digunakan untuk menemukan sisa setelah satu nilai dibagi dengan nilai lain.

Contoh: MOD dari 5 dibagi dengan 2 adalah 1.

Query:

```
SELECT region_id, MOD(region_id,2) FROM countries;
```

REGION_ID	MOD(REGION_ID,2)
2	0
1	1
1	1
2	0

Output:

- Kolom 'MOD(REGION_ID,2)' menunjukkan jumlah bandara pada setiap region berjumlah ganjil (1) atau genap (0).

4.2 Date Function

1. SYSDATE

SYSDATE adalah fungsi tanggal yang digunakan untuk menghasilkan tanggal dan waktu saat ini dari database server.

Query:

```
SELECT SYSDATE
FROM dual;
```

Output:

SYSDATE
21-May-2017

- Date Data Type

Type data 'Date' pada Oracle selalu menyimpan informasi sebagai 4 digit angka: 2 digit pertama menunjukkan abad dan 2 digit terakhir untuk menunjukkan tahun.

Contoh:

Database Oracle menyimpan tahun sebagai 1996 atau 2004, bukan hanya 96 atau 04.

Query:

Examples:	Result
<pre>SELECT last_name, hire_date + 60 FROM employees;</pre>	Adds 60 days to hire_date.

Output:

LAST_NAME	HIRE_DATE+60
King	16-Aug-1987
Kochhar	20-Nov-1989
De Haan	14-Mar-1993
Whalen	16-Nov-1987
Higgins	06-Aug-1994
Gietz	06-Aug-1994
Zlotkey	29-Mar-2000
Abel	10-Jul-1996
Taylor	23-May-1998
Grant	23-Jul-1999

Query:

<pre>SELECT last_name, (SYSDATE - hire_date)/7 FROM employees;</pre>	Displays the number of weeks since the employee was hired.
--	--

Output:

LAST_NAME	(SYSDATE-HIRE_DATE)/7
King	1561.70294477513227513227513227513
Kochhar	1443.56008763227513227513227513227
De Haan	1270.70294477513227513227513227513
Whalen	1548.56008763227513227513227513227
Higgins	1197.84580191798941798941798941799
Gietz	1197.84580191798941798941798941799
Zlotkey	903.274373346560846560846560846561
Abel	1097.27437334656084656084656084656
Taylor	999.845801917989417989417989417989
Grant	938.988659060846560846560846560847

Query:

<pre>SELECT employee_id, (end_date - start_date)/365 AS "Tenure in last job" FROM job_history;</pre>	Finds the number of days employee held a job, then divides by 365 to display in years.
--	--

Output:

EMPLOYEE_ID	Tenure in last job
200	5.75342465753424657534246575342465753425
101	3.38082191780821917808219178082191780822
200	4.5041095890410958904109589041095890411
101	4.1013698630136986301369863013698630137
176	.997260273972602739726027397260273972603
176	.772602739726027397260273972602739726027
122	.997260273972602739726027397260273972603
114	1.77260273972602739726027397260273972603
102	5.52876712328767123287671232876712328767
201	3.83835616438356164383561643835616438356

- MONTHS_BETWEEN:

Mengambil 2 buah data DATE dan menghasilkan nilai berupa jumlah bulan diantara kedua tanggal tersebut.

Query:

```
SELECT last_name, hire_date
FROM employees
WHERE MONTHS_BETWEEN
      (SYSDATE, hire_date)>240;
```

Output:

LAST_NAME	HIRE_DATE
King	17-Jun-1987
Kochhar	21-Sep-1989
De Haan	13-Jan-1993
Whalen	17-Sep-1987
Higgins	07-Jun-1994
Gietz	07-Jun-1994
Abel	11-May-1996
Rajs	17-Oct-1995
Davies	29-Jan-1997
Hunold	03-Jan-1990

- ADD_MONTHS:

Mengambil 2 data berupa tanggal (DATE) dan angka, kemudian menghasilkan tanggal (DATE) dengan menambahkan angka tersebut kedalam komponen bulan dari data DATE.

Query:

```
SELECT ADD_MONTHS (SYSDATE, 12)
       AS "Next Year"
FROM dual;
```

Output:

Next Year
21-May-2018

- **NEXT_DAY**

Mengambil 2 argumen berupa tanggal (DATE) dan hari, kemudian menghasilkan tanggal dari hari tersebut pada minggu setelahnya.

Query:

```
SELECT NEXT_DAY (SYSDATE, 'Saturday')
       AS "Next Saturday"
FROM dual;
```

Output:

Next Saturday
27-May-2017

- **LAST_DAY:**

Mengambil tanggal (DATE) untuk mencari hari terakhir pada bulan dari tanggal (DATE) yang dilihat.

Query:

```
SELECT LAST_DAY (SYSDATE)
       AS "End of the Month"
FROM dual;
```

Output:

End of the Month
31-May-2017

- **ROUND**

Menghasilkan nilai DATE yang telah dibulatkan kedalam unit yang telah ditentukan pada argument kedua.

Query:

```
SELECT hire_date,
       ROUND (hire_date, 'Month')
FROM employees
WHERE department_id=50;
```

Output:

HIRE_DATE	ROUND(HIRE_DATE,'MONTH')
16-Nov-1999	01-Dec-1999
17-Oct-1995	01-Nov-1995
29-Jan-1997	01-Feb-1997
15-Mar-1998	01-Mar-1998
09-Jul-1998	01-Jul-1998

- **TRUNC**

Menghasilkan nilai DATE yang telah dipotong kedalam unit yang telah ditentukan pada statement kedua.

Query:

SELECT hire_date,	16/Nov/1999	01/Jan/1999
TRUNC(hire_date, 'Year')	17/Oct/1995	01/Jan/1995
FROM employees	29/Jan/1997	01/Jan/1997
WHERE department_id=50;

Output:

HIRE_DATE	ROUND(HIRE_DATE,'MONTH')
16-Nov-1999	01-Dec-1999
17-Oct-1995	01-Nov-1995
29-Jan-1997	01-Feb-1997
15-Mar-1998	01-Mar-1998
09-Jul-1998	01-Jul-1998

Berikut adalah contoh query SQL query dengan menggunakan beberapa fungsi DATE:

```
SELECT employee_id, hire_date,
ROUND(MONTHS_BETWEEN(SYSDATE, hire_date)) AS TENURE,
ADD_MONTHS (hire_date, 6) AS REVIEW,
NEXT_DAY(hire_date, 'FRIDAY'),
LAST_DAY(hire_date)
FROM employees
WHERE MONTHS_BETWEEN (SYSDATE, hire_date) > 36;
```

Output:

EMPLOYEE_ID	HIRE_DATE	TENURE	REVIEW	NEXT_DAY(HIRE_DATE,'FRIDAY')	LAST_DAY(HIRE_DATE)
100	17/Jun/1987	316	17/Dec/1987	19/Jun/1987	30/Jun/1987
101	21/Sep/1989	289	21/Mar/1990	22/Sep/1989	30/Sep/1989
102	13/Jan/1993	249	13/Jul/1993	15/Jan/1993	31/Jan/1993
200	17/Sep/1987	313	17/Mar/1988	18/Sep/1987	30/Sep/1987
205	07/Jun/1994	232	07/Dec/1994	10/Jun/1994	30/Jun/1994
206	07/Jun/1994	232	07/Dec/1994	10/Jun/1994	30/Jun/1994
149	29/Jan/2000	165	29/Jul/2000	04/Feb/2000	31/Jan/2000
174	11/May/1996	209	11/Nov/1996	17/May/1996	31/May/1996
176	24/Mar/1998	187	24/Sep/1998	27/Mar/1998	31/Mar/1998
178	24/May/1999	173	24/Nov/1999	28/May/1999	31/May/1999

TUGAS OBSERVASI-DEMONSTRASI/PRAKTIK

1. Tampilkan gabungan kata “Oracle” dan “Academy” dari tabel dual kedalam kolom “Subject” menggunakan huruf kapital, serta tampilkan pula angka yang menyatakan panjang string dari subject tersebut kedalam kolom “Length of Subject” dalam satu kali output.

Subject	Length of Subject
ORACLEACADEMY	13

2. Tampilkan semua last_name dari table employees yang memiliki karakter ‘a’ pada posisi kedua dalam urutan karakter string last_name.

LAST_NAME
Taylor
Rajs
Davies
Matos
Vargas
Hartstein
Fay

3. Tuliskan query yang dapat menghasilkan output berikut :

PAD STRING
****Oracle****Internet****Academy****

4. Tuliskan query untuk menampilkan karakter pertama dari first_name yang digabungkan dengan last_name, dan salary untuk karyawan yang bekerja di departemen 20 dalam kolom alias “Employees Data”. Gunakan tabel EMPLOYEES. Gunakan konsep Substitution Variable untuk menjalankan kasus yang berbeda (department_id = 30, 40, 50, dst).
5. Tuliskan query untuk menampilkan:
 - employee_id, salary, commission_pct, nilai round dari salary dibagi comission_pct dengan nilai decimal 4) jadikan kolom ‘hasil bagi’, trunc dari salary dibagi

commission_pct dengan nilai decimal 2, dan MOD 2 dari employee_id dari tabel employees, dimana salary lebih dari 5000.

EMPLOYEE_ID	SALARY	COMMISSION_PCT	hasil bagi		
100	24000	-	-	-	0
101	17000	-	-	-	1
102	17000	-	-	-	0
205	12000	-	-	-	1
206	8300	-	-	-	0
149	10500	.2	52500	52500	1
174	11000	.3	36666.6667	36666.66	0
176	8600	.2	43000	43000	0
178	7000	.15	46666.6667	46666.66	0
124	5800	-	-	-	0

4. Tampilkan last_name dan tanggal employee dipekerjakan (hire date) dikurangi sebulan sebagai kolom 'H-30' dari tabel employees.

LAST_NAME	H-30
King	18-May-1987
Kochhar	22-Aug-1989
De Haan	14-Dec-1992
Whalen	18-Aug-1987
Higgins	08-May-1994
Gietz	08-May-1994
Zlotkey	30-Dec-1999
Abel	11-Apr-1996
Taylor	22-Feb-1998
Grant	24-Apr-1999

5. Tampilkan tanggal employee dipekerjakaan ditambah 24 bulan dan tanggal di hari selasa seminggu setelah employee mulai bekerja dari tabel employees.

17-Jun-1989	23-Jun-1987
21-Sep-1991	26-Sep-1989
13-Jan-1995	19-Jan-1993
17-Sep-1989	22-Sep-1987
07-Jun-1996	14-Jun-1994
07-Jun-1996	14-Jun-1994
29-Jan-2002	01-Feb-2000
11-May-1998	14-May-1996
24-Mar-2000	31-Mar-1998
24-May-2001	25-May-1999