

Single Row Function

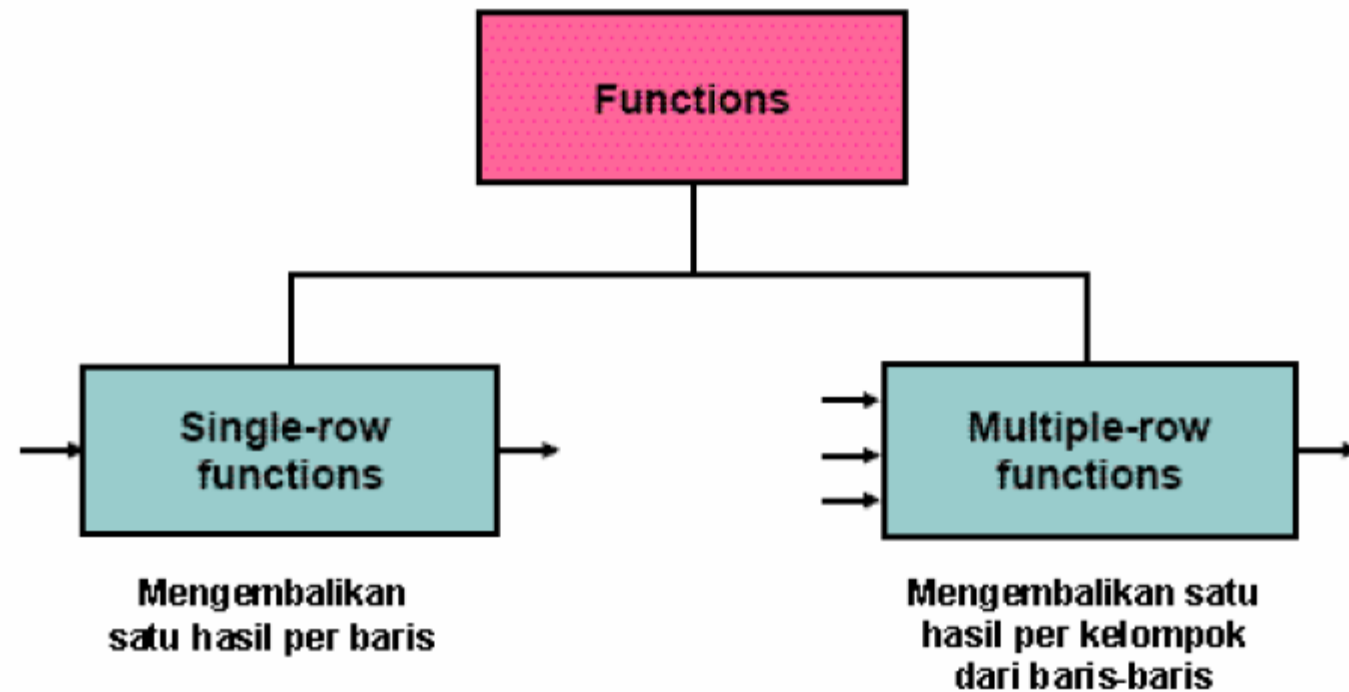


Menggunakan *Single-Row Function* untuk Meng-*customize* Output

ORACLE

Copyright © 2004, Oracle. All rights reserved.

Dua Tipe pada Fungsi-Fungsi SQL

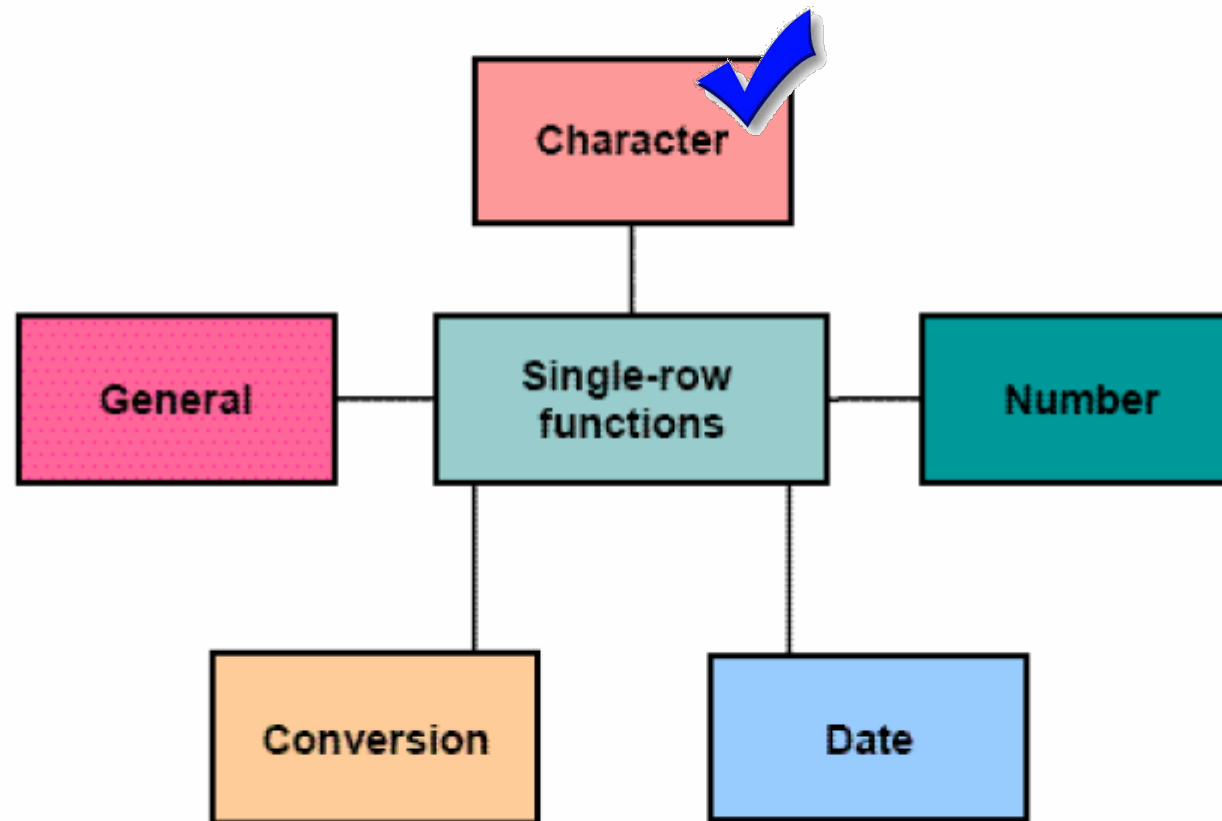


ORACLE

Penggunaan Fungsi

```
Select Col_Name,Fungsi(Col_Name), Col_Name  
From Table ;
```

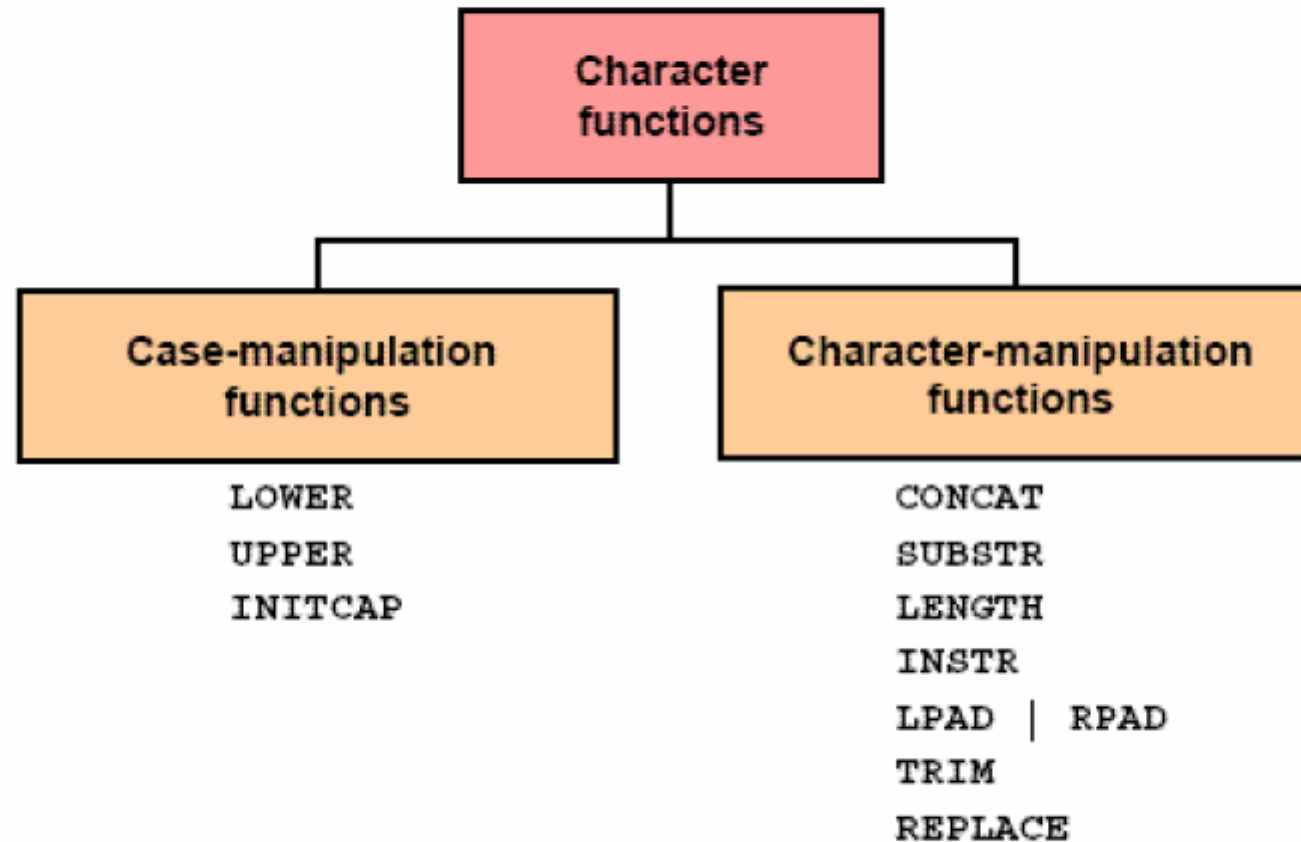
Single-Row Functions



ORACLE

Copyright © 2004, Oracle. All rights reserved.

Fungsi-Fungsi *Character*



ORACLE

Copyright © 2004, Oracle. All rights reserved.

Fungsi-Fungsi *Case-Manipulation*

Fungsi-fungsi ini merubah bentuk pada karakter-karakter *string* :

Function	Result
<code>LOWER('SQL Course')</code>	<code>sql course</code>
<code>UPPER('SQL Course')</code>	<code>SQL COURSE</code>
<code>INITCAP('SQL Course')</code>	<code>Sql Course</code>

Fungsi-Fungsi *Character-Manipulation*

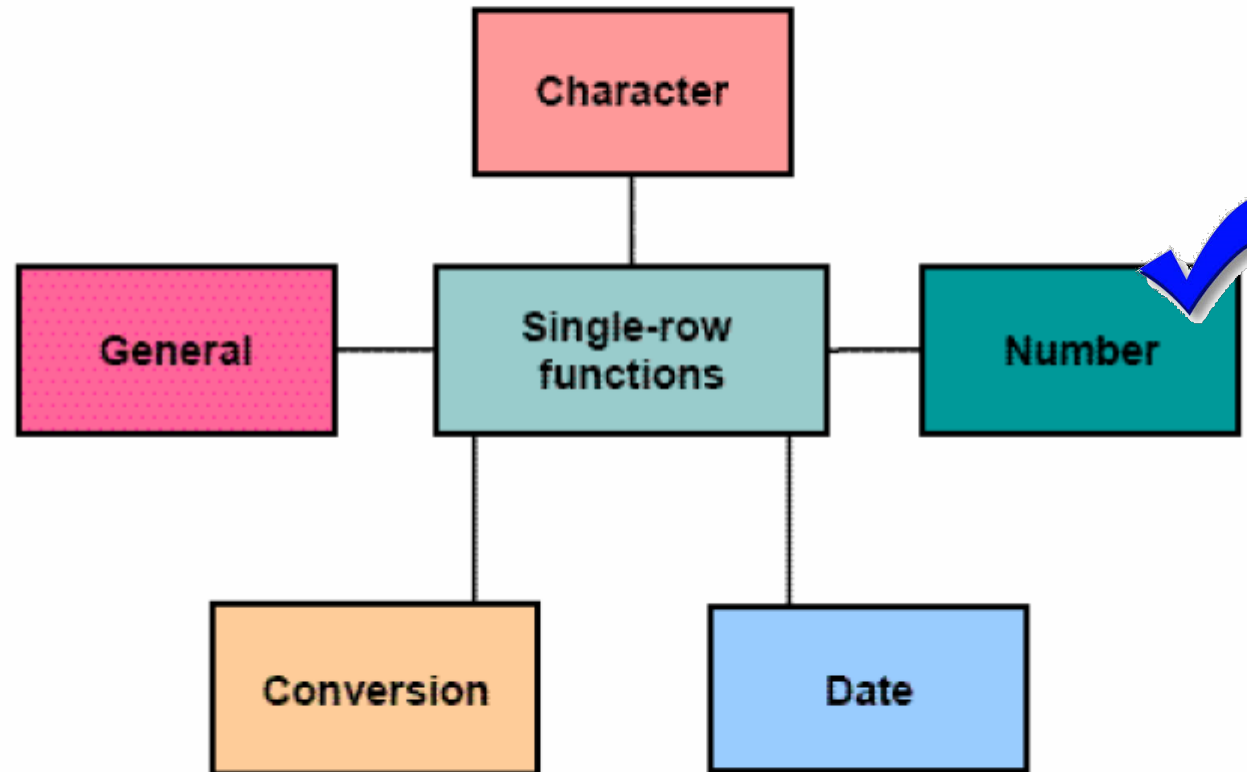
Fungsi-fungsi manipulasi karakter-karakter *string* :

Function	Result
CONCAT('Hello', 'World')	HelloWorld
SUBSTR('HelloWorld',1,5)	Hello
LENGTH('HelloWorld')	10
INSTR('HelloWorld', 'W')	6
LPAD(salary,10, '*')	*****24000
RPAD(salary, 10, '*')	24000*****
REPLACE ('JACK and JUE', 'J', 'BL')	BLACK and BLUE
TRIM('H' FROM 'HelloWorld')	elloWorld

Contoh Sintax :

- `SELECT kd_brg, lower(nama_barang) FROM Barang;`
- `SELECT upper(nama_barang), Initcap(nama_barang) FROM Barang;`
- `SELECT concat(kd_brg,jeniskemasan) "Gabungan" FROM Barang;`
- `SELECT nm_supp, substr(nm_supp,1,7) potongan, length(nm_supp) panjang FROM Supplier;`
- `SELECT nm_supp, instr(nm_supp,'m') "Posisi M" FROM Supplier;`
- `SELECT nama_barang, lpad(stok,6,'x'), rpad(stok,6,'x') FROM Barang;`
- `SELECT nama_barang, replace(nama_barang,'a','u') ubah FROM Barang;`
- `SELECT trim(nama_barang) FROM Barang;`
- `SELECT nama_barang,trim ('K' from nama_barang) FROM Barang;`

Single-Row Functions



ORACLE

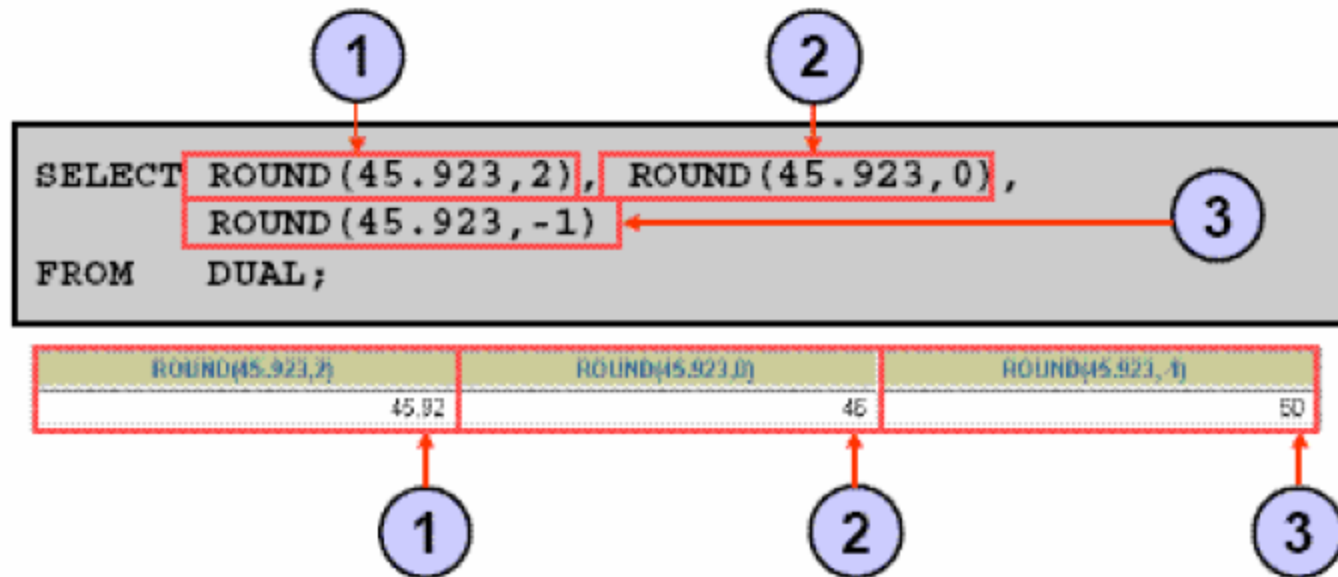
Copyright © 2004, Oracle. All rights reserved.

Fungsi-Fungsi *Number*

- **ROUND** : Pembulatan nilai ke desimal tertentu
- **TRUNC** : Memotong nilai ke desimal tertentu
- **MOD** : Mengembalikan sisa bagi

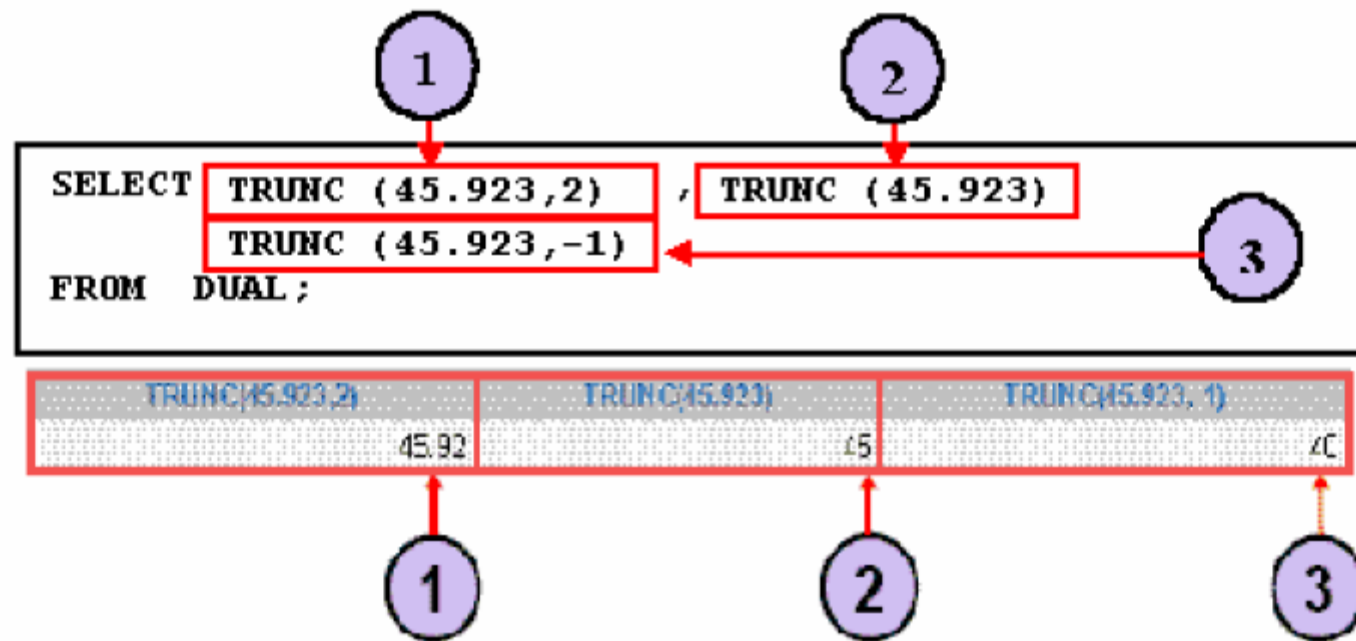
Function	Result
ROUND (45.926, 2)	45.93
TRUNC (45.926, 2)	45.92
MOD (1600, 300)	100

Menggunakan Fungsi ROUND



DUAL adalah *dummy table* (tabel contoh) yang dapat anda gunakan untuk melihat hasil-hasil dari fungsi-fungsi dan perhitungan-perhitungan.

Menggunakan Fungsi TRUNC



Menggunakan Operator Aritmatika *Dates*

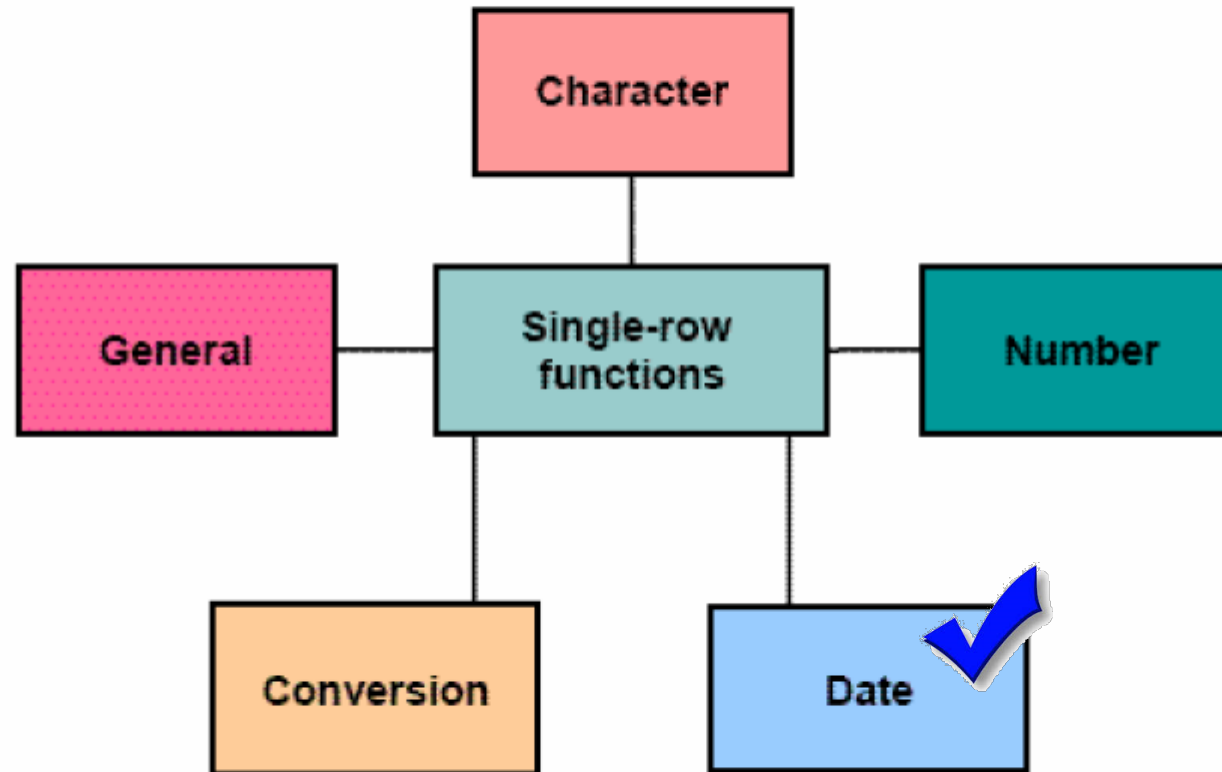
```
SELECT last_name, (SYSDATE-hire_date)/7 AS WEEKS  
FROM employees  
WHERE department_id = 90;
```

LAST_NAME	WEEKS
King	744.245385
Kochhar	626.102538
De Haan	453.245385

Contoh Sintax :

- `SELECT 42.766, round(42.766,2), round(42.766,0), round(42.766,-1), round(42.766,-2), round(42.766,-2) FROM Dual;`
- `SELECT 42.766, trunc(42.766,2), trunc(42.766,0), trunc(42.766,-1), trunc(42.766,-2), trunc(42.766,-2) FROM Dual;`
- `SELECT 90.268, round(90.268,2), round(90.268,0), round(90.268,-1), round(90.268,-2), round(90.268,-2) FROM Dual;`
- `SELECT 90.268, trunc(90.268,2), trunc(90.268,0), trunc(90.268,-1), trunc(90.268,-2), trunc(90.268,-2) FROM Dual;`
- `SELECT nama_barang, stok, stok/3 ,round(stok/3,2),mod(stok,3) sisa FROM Barang;`
- `SELECT nm_supp, lhr_supp,round((sysdate-lhr_supp)/365,0) umur FROM Supplier;`

Single-Row Functions



ORACLE

Copyright © 2004, Oracle. All rights reserved.

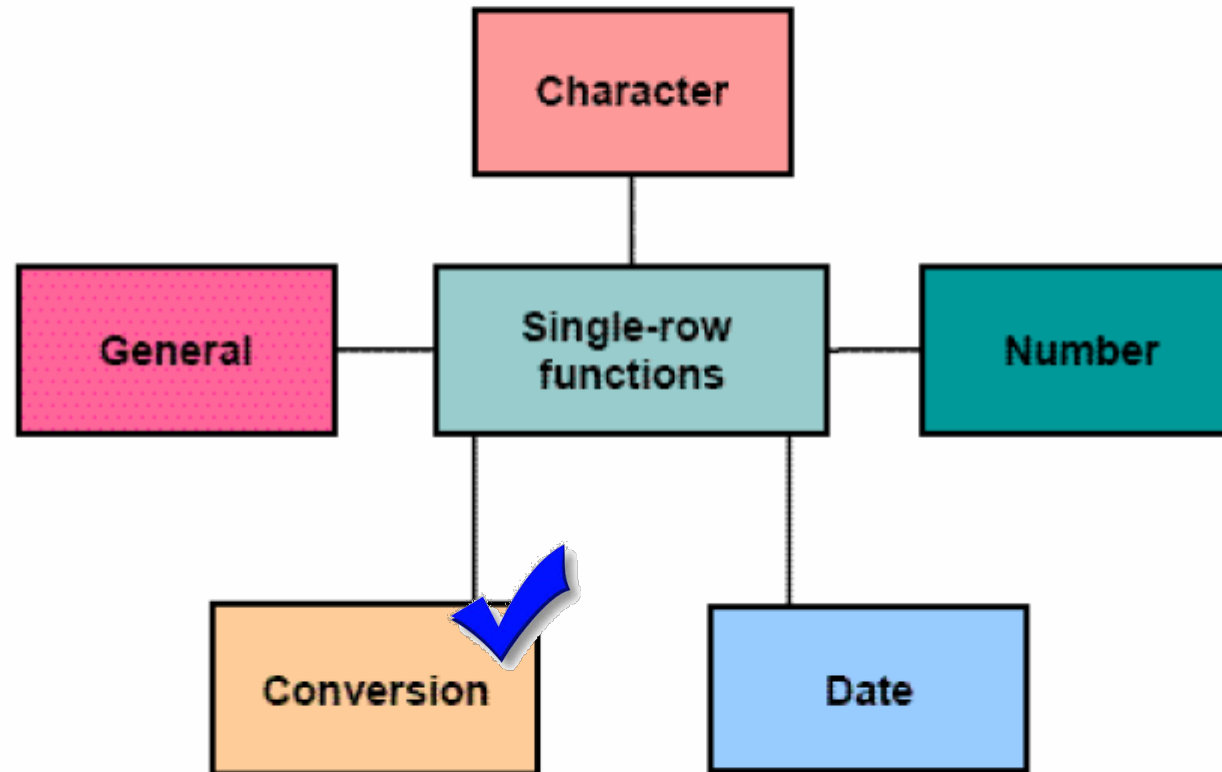
Menggunakan Fungsi-Fungsi *Date*

Fungsi	Hasil
MONTHS_BETWEEN ('01-SEP-95', '11-JAN-94')	19.6774194
ADD_MONTHS ('11-JAN-94', 6)	'11-JUL-94'
NEXT_DAY ('01-SEP-95', 'FRIDAY')	'08-SEP-95'
LAST_DAY ('01-FEB-95')	'28-FEB-95'

Contoh Sintax :

- `SELECT kd_supp, kd_brg, jml_pasok, months_between(sysdate, tgl_pasok) FROM PasokanBarang;`
- `SELECT nm_supp, lhr_supp, add_months(lhr_supp, 4) FROM Supplier;`
- `SELECT kd_supp, kd_brg, jml_pasok, tgl_pasok, next_day(tgl_pasok, 'Friday') FROM PasokanBarang;`
- `SELECT nm_supp, lhr_supp, last_day(lhr_supp) FROM Supplier;`

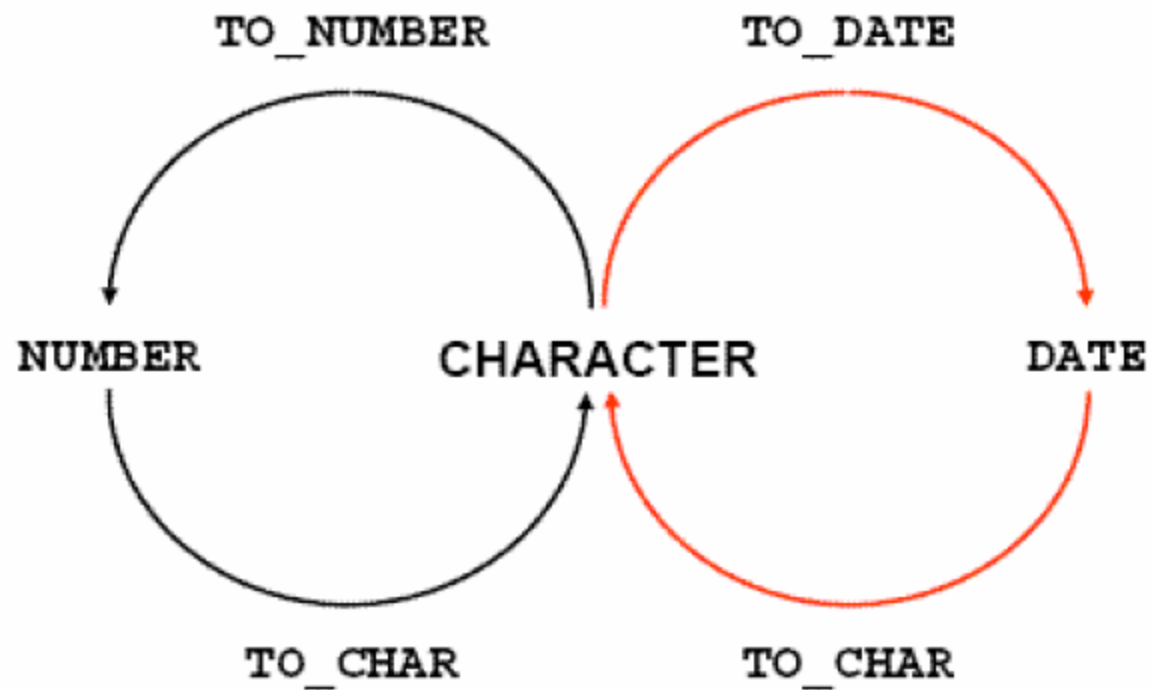
Single-Row Functions



ORACLE

Copyright © 2004, Oracle. All rights reserved.

Konversi Tipe Data Eksplisit



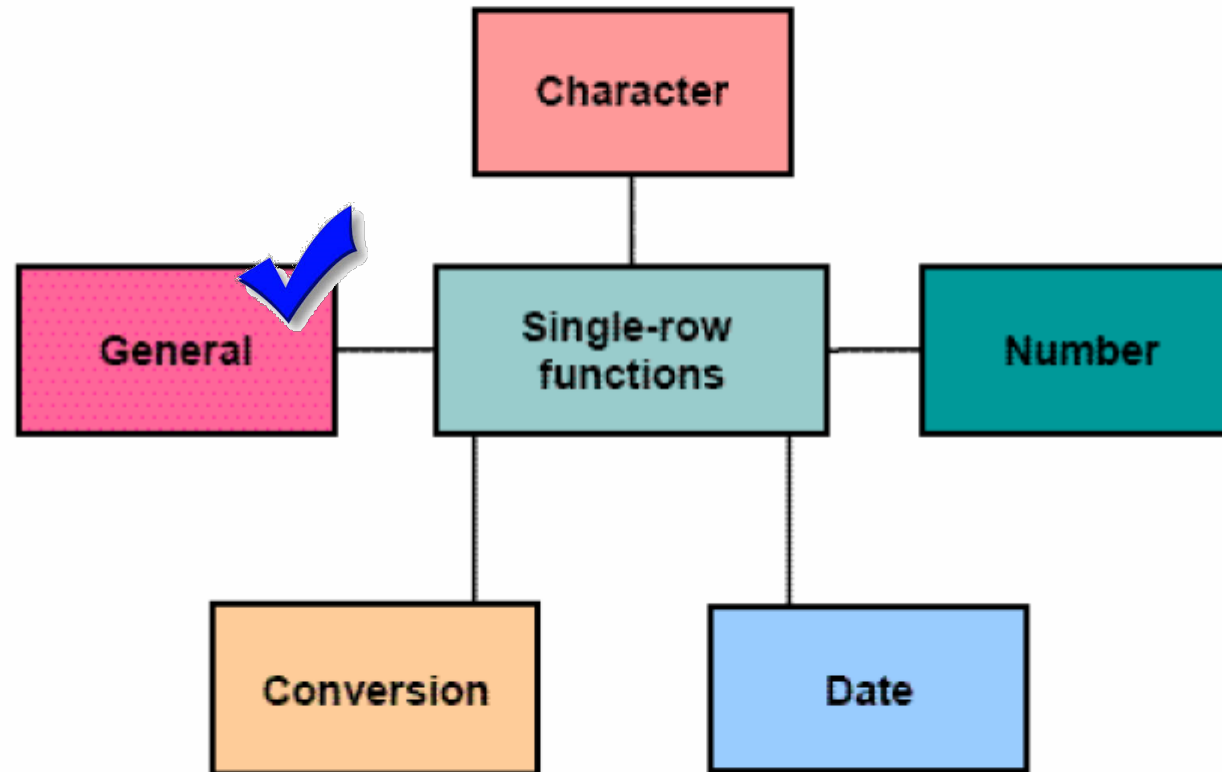
ORACLE

Copyright © 2004, Oracle. All rights reserved.

Contoh Sintax :

- `SELECT to_number('123.456','999.999') FROM Dual;`
- `SELECT to_number('56','99') FROM Dual;`
- `SELECT to_date('23-JAN-2021','DD-MON-YYYY') FROM Dual;`
- `SELECT nm_supp, lhr_supp, to_char(lhr_supp,'MON') bulan,
to_char(lhr_supp,'MON-YYYY') bulantahun,
to_char(lhr_supp,'YYYY') tahun FROM Supplier;`

Single-Row Functions



ORACLE

Copyright © 2004, Oracle. All rights reserved.

Fungsi NVL

Mengubah nilai *null* menjadi nilai aktual :

- Tipe data – tipe data yang dapat digunakan adalah tanggal, karakter, dan angka.
- Tipe data-tipe data harus sesuai :
 - NVL (commission_pct, 0)
 - NVL (hire_date, '01-Jan-97')
 - NVL (job_id, 'No Job Yet')

Menggunakan Ekspresi CASE

Kemudahan-kemudahan pencarian kondisional dengan melakukan suatu pekerjaan pada suatu pernyataan IF-THEN-ELSE :

```
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;
```

LAST_NAME	JOB_ID	SALARY	REVISED_SALARY

Lorentz	IT_PROG	4200	4620
Mourgos	ST_MAN	5900	6785
Rajs	ST_CLERK	3500	4025

Gietz	AC_ACCOUNT	8300	8300

20 rows selected.

ORACLE

Copyright © 2004, Oracle. All rights reserved.

Fungsi DECODE

Kemudahan-kemudahan pencarian kondisional dengan melakukan suatu pekerjaan pada suatu ekspresi CASE atau suatu pernyataan IF-THEN-ELSE :

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

Contoh Sintax :

- `SELECT nm_supp, almt_supp, jkel_supp, email, nvl(email, 'joshua@gmail.com') email_nvl FROM Supplier;`
- `SELECT nama_barang, jenis, stok, nvl(stok, 999) stok_nvl FROM Barang;`
- `SELECT nm_supp, lhr_supp, nvl(lhr_supp, '31-JAN-2021') tgl_nvl FROM Supplier;`

Contoh Sintax :

```
SELECT nama_barang, jenis, stok,  
case jenis when 'Peralatan' then stok+9  
      when 'Alat Tulis' then stok+7  
      when 'Asesoris' then stok+6  
else stok+3 end "Stok Baru"  
FROM Barang;
```

```
SELECT nama_barang, jenis, stok,  
decode(jenis, 'Peralatan', stok+9, 'Alat Tulis', stok+7, 'Asesoris', stok+6,  
stok+3) "Stok Baru"  
FROM Barang;
```

Contoh Single Row Function sebagai kondisi/syarat menampilkan data

- `SELECT upper(nama_barang) nama, lower(jenis) jenis, stok
FROM Barang WHERE substr(jenis,1,1)='P';`
- `SELECT lower(nm_supp) nama, length(almt_supp)
panjang, lhr_sup FROM Supplier WHERE
to_char(lhr_supp,'YYYY') = '2001'`

Praktekkan Fungsi Single Row Function!

1. Tampilkan data Supplier dengan format, nama dengan huruf kecil, alamat dengan huruf besar, jkel dan telepon untuk Supplier pria !
2. Tampilkan data Supplier yaitu kode, nama, dan alamatnya. Kode supplier yang ditampilkan tanpa hurup "Sp". Urutkan tampilan berdasarkan namanya (Z-A) !
3. Tampilkan nama Supplier dan berapa panjang karakter namanya. Urutkan berdasarkan jumlah karakter nama terpanjang !
4. Tampilkan data nama, jkel, tgl lahir dan umur Supplier. Urutkan dari Supplier paling tua !
5. Tampilkan semua data Supplier yang lahir di bulan februari !
6. **Buat tangkapan layar** untuk hasilnya fungsi Single Row Function diatas. Hasil kumpulan tangkapan layar diunggah melalui Brilian dengan nama file : **Demo3_P1_NimPanjang_Nama.PDF paling lambat hari Sabtu, 05 Juni 2021 pukul 23:59 WIB → Kelas P1**
Demo3_Q1_NimPanjang_Nama.PDF paling lambat hari Senin, 07 Juni 2021 pukul 12:00 WIB → Kelas P1

Tetap
Semangat