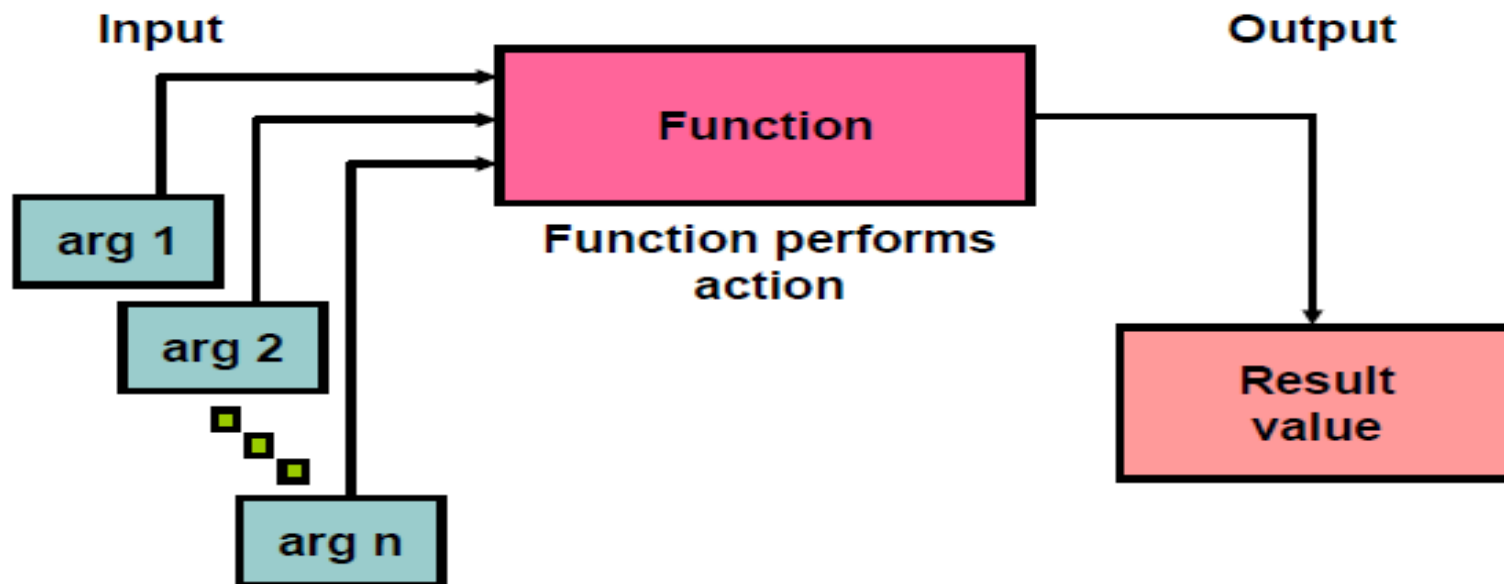


# Using Single-Row Functions to Customize Output

# Tujuan Pembelajaran:

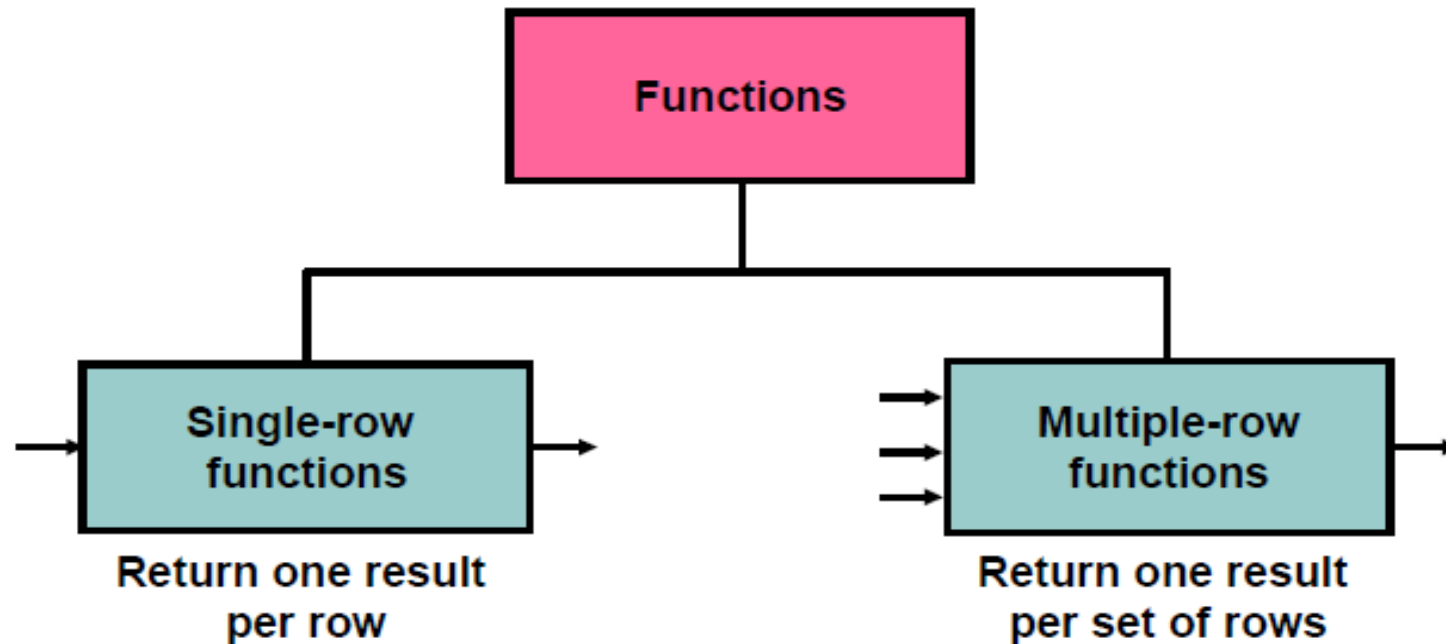
- Variasi Fungsi dalam Sql
- Penggunaan fungsi Character, Date, Number, Konversi
- Penggunaan fungsi General

# SQL Functions



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

## Two Types of SQL Functions



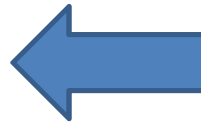
Single-row function : tiap record menghasilkan satu output

Multiple-row function : Beberapa record menghasilkan satu output

## Contoh Single-row Function:

Select Upper(last\_name) from employees;

UPPER(LAST_NAME)
ABEL
ANDE
ATKINSON
AUSTIN
BAER
BAIDA
BANDA
BATES

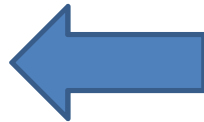


LAST_NAME
Abel
Ande
Atkinson
Austin
Baer
Baida
Banda
Bates

## Contoh Multiple-row Function:

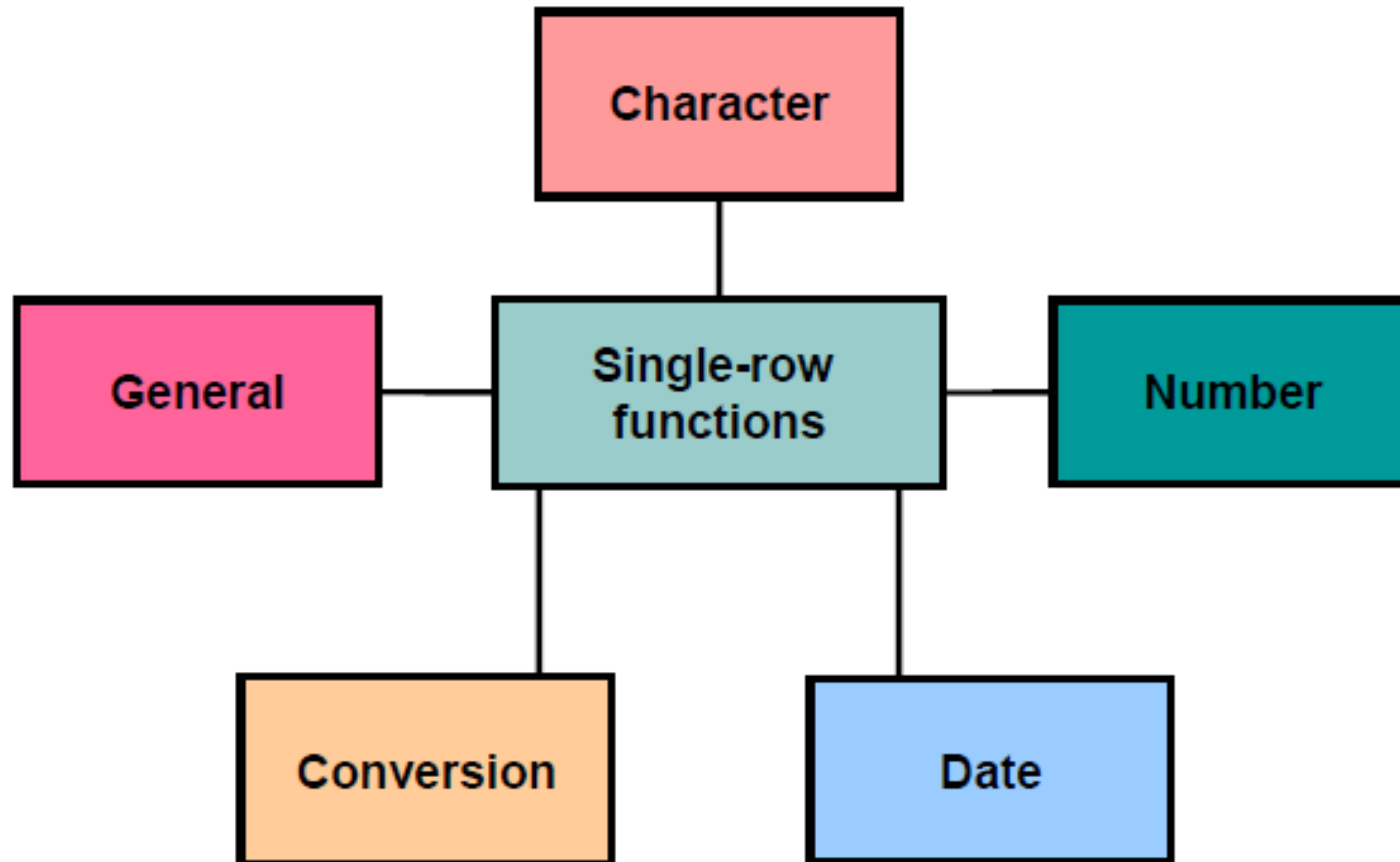
Select max(salary) from employees;

MAX(SALARY)
24000

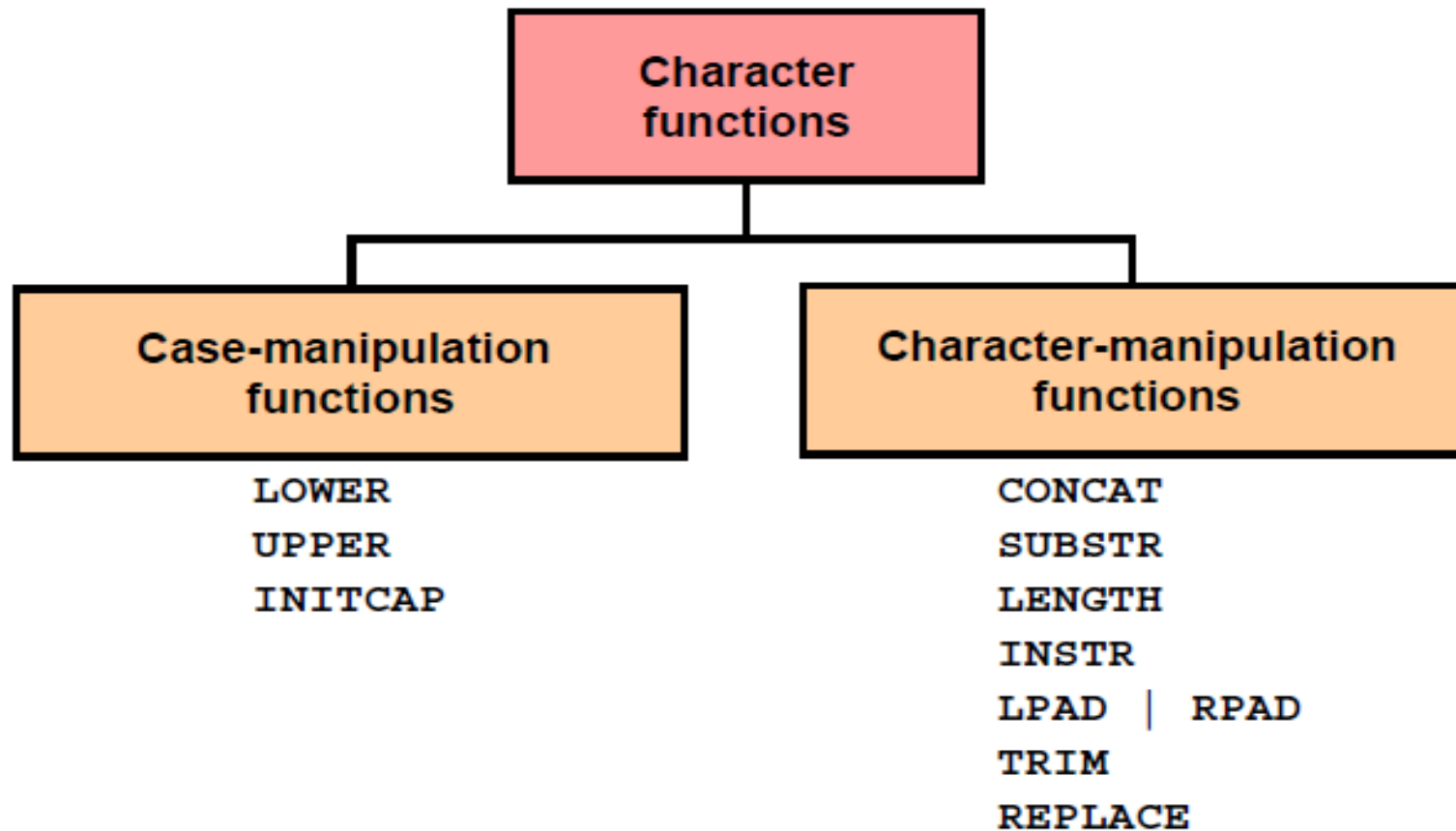


SALARY
24000
17000
17000
9000
6000
4800
4800
4200

# Single-Row Functions



# Character Functions





# Case-Manipulation Functions

These functions convert case for character strings:

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

Contoh:

```
Select last_name, upper(last_name), lower(last_name), initcap(last_name)
from employees;
```

LAST_NAME	UPPER(LAST_NAME)	LOWER(LAST_NAME)	INITCAP(LAST_NAME)
Abel	ABEL	abel	Abel
Ande	ANDE	ande	Ande
Atkinson	ATKINSON	atkinson	Atkinson
Austin	AUSTIN	austin	Austin
Baer	BAER	baer	Baer
Baida	BAIDA	baida	Baida
Banda	BANDA	banda	Banda
Bates	BATES	bates	Bates

Apa maksud dari perbandingan kedua contoh perintah Select di bawah ini?

**Display the employee number, name, and department number for employee Higgins:**

```
SELECT employee_id, last_name, department_id
FROM   employees
WHERE  last_name = 'higgins';
no rows selected
```

```
SELECT employee_id, last_name, department_id
FROM   employees
WHERE  LOWER(last_name) = 'higgins';
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
205	Higgins	110

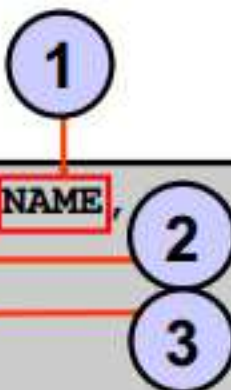
## Character-Manipulation Functions

These functions manipulate character strings:


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

# Using the Character-Manipulation Functions

```
SELECT employee_id, CONCAT(first_name, last_name) NAME,
       job_id, LENGTH(last_name),
       INSTR(last_name, 'a') 'Contains 'a'?'
FROM   employees
WHERE  SUBSTR(job_id, 4) = 'REP';
```



EMPLOYEE_ID	NAME	JOB_ID	LENGTH(LAST_NAME)	Contains 'a'?
174	EllenAbel	SA_REP	4	0
176	JonathonTaylor	SA_REP	6	2
178	KimberelyGrant	SA_REP	5	3
202	PatFay	MK_REP	3	2



# Apa maksud query berikut?

```
SELECT employee_id, CONCAT(first_name, last_name) NAME, LENGTH  
      (last_name), INSTR(last_name, 'a') "Contains'a'?"  
FROM employees  
WHERE SUBSTR(last_name, -1, 1) = 'n';
```

EMPLOYEE_ID	NAME	LENGTH(LAST_NAME)	Contains'a'?
102	LexDe Haan	7	5
105	DavidAustin	6	0
110	JohnChen	4	0
112	Jose ManuelUrman	5	4
123	ShantaVollman	7	6
130	MozheAtkinson	8	0
132	TJOlson	5	0
133	JasonMallin	6	2
151	DavidBernstein	9	0

Contoh:

Select first\_name, last\_name, concat(first\_name,last\_name),  
substr(first\_name,2,3), length(last\_name) ,instr(first\_name, 'a') from employees;

FIRST_NAME	LAST_NAME	CONCAT(FIRST_NAME, LAST_NAME)	SUBSTR(FIRST_NAME, 2, 3)	LENGTH(LAST_NAME)	INSTR(FIRST_NAME, 'A')
Ellen	Abel	EllenAbel	lle	4	0
Sundar	Ande	SundarAnde	und	4	5
Mozhe	Atkinson	MozheAtkinson	ozh	8	0
David	Austin	DavidAustin	avi	6	2
Hermann	Baer	HermannBaer	erm	4	5
Shelli	Baida	ShelliBaida	hel	5	0
Amit	Banda	AmitBanda	mit	5	0
Elizabeth	Bates	ElizabethBates	liz	5	5
Sarah	Bell	SarahBell	ara	4	2
David	Bernstein	DavidBernstein	avi	9	2

Contoh:

Select salary, lpad(salary,12,'&'), rpad(salary,12,'&') from employees;

SALARY	LPAD(SALARY,12,'&')	RPAD(SALARY,12,'&')
24000	&&&&&&&24000	24000&&&&&&&
17000	&&&&&&&17000	17000&&&&&&&
17000	&&&&&&&17000	17000&&&&&&&
9000	&&&&&&&&9000	9000&&&&&&&&
6000	&&&&&&&&6000	6000&&&&&&&&
4800	&&&&&&&&4800	4800&&&&&&&&
4800	&&&&&&&&4800	4800&&&&&&&&
4200	&&&&&&&&4200	4200&&&&&&&&
12000	&&&&&&&12000	12000&&&&&&&
9000	&&&&&&&&9000	9000&&&&&&&&



Contoh:

Select last\_name, replace(last\_name, 'e', 'o'), trim('a' from last\_name)  
from employees;

LAST_NAME	REPLACE(LAST_NAME,'E','O')	TRIM('A'FROMLAST_NAME)
Abel	Abol	Abel
Ande	Ando	Ande
Atkinson	Atkinson	Atkinson
Austin	Austin	Austin
Baer	Baor	Baer
Baida	Baida	Baid
Banda	Banda	Band
Bates	Batos	Bates

Contoh:

Select last\_name, replace(last\_name, 'e', 'o') GANTI,  
trim('a' from last\_name) HAPUS from employees;

LAST_NAME	GANTI	HAPUS
Abel	Abol	Abel
Ande	Ando	Ande
Atkinson	Atkinson	Atkinson
Austin	Austin	Austin
Baer	Baor	Baer
Baida	Baida	Baid
Banda	Banda	Band
Bates	Batos	Bates

Apa tujuan perintah di bawah ini:

```
SELECT employee_id, CONCAT(first_name, last_name) NAME, job_id, LENGTH  
      (last_name), INSTR(last_name, 'a')  
FROM employees
```

EMPLOYEE_ID	NAME	JOB_ID	LENGTH(LAST_NAME)	INSTR(LAST_NAME,'A')
100	StevenKing	AD_PRES	4	0
101	NeenaKochhar	AD_VP	7	6
102	LexDe Haan	AD_VP	7	5
103	AlexanderHunold	IT_PROG	6	0
104	BruceErnst	IT_PROG	5	0
105	DavidAustin	IT_PROG	6	0
106	ValliPataballa	IT_PROG	9	2
107	DianaLorentz	IT_PROG	7	0

Bagaimana perintah untuk membatasi baris (record) di atas yang memiliki Job-Id berawalan 'IT'?

Bagaimana perintah untuk membatasi baris (record) di atas yang memiliki Job-Id berawalan 'IT'?

```
SELECT employee_id, CONCAT(first_name, last_name) NAME, job_id, LENGTH  
      (last_name), INSTR(last_name, 'a')  
FROM employees  
Where upper(substr(job_id,1,2)) = 'IT'
```

EMPLOYEE_ID	NAME	JOB_ID	LENGTH(LAST_NAME)	INSTR(LAST_NAME,'A')
103	AlexanderHunold	IT_PROG	8	0
104	BruceErnst	IT_PROG	5	0
105	DavidAustin	IT_PROG	8	0
106	ValliPataballa	IT_PROG	9	2
107	DianaLorentz	IT_PROG	7	0

# Number Function

- ROUND: Pembulatan nilai ke atas, dengan angka desimal yang sudah ditentukan
- TRUNC: Pembulatan nilai ke bawah, dengan angka desimal yang sudah ditentukan
- MOD: Mengembalikan nilai sisa dari nilai pembagian

Function	Result
ROUND ( 45 . 926 , 2 )	45 . 93
TRUNC ( 45 . 926 , 2 )	45 . 92
MOD ( 1600 , 300 )	100

```
SELECT ROUND(45.923,2), ROUND(45.923,0),
ROUND(45.923,-1) FROM DUAL;
```

ROUND(45.923,2)	ROUND(45.923,0)	ROUND(45.923,-1)
45.92	46	50

```
SELECT TRUNC(45.923,2), TRUNC(45.923,0),
TRUNC(45.923,-1) FROM DUAL;
```

TRUNC(45.923,2)	TRUNC(45.923,0)	TRUNC(45.923,-1)
45.92	45	40

```
SELECT Job_id, last_name, salary, MOD(salary, 5000)
FROM employees WHERE job_id = 'SA_REP';
```

JOB_ID	LAST_NAME	SALARY	MOD(SALARY,5000)
SA_REP	Tucker	10000	0
SA_REP	Bernstein	9500	4500
SA_REP	Hall	9000	4000
SA_REP	Olsen	8000	3000
SA_REP	Cambault	7500	2500
SA_REP	Tuvault	7000	2000
SA_REP	King	10000	0
SA_REP	Sully	9500	4500
SA_REP	McEwen	9000	4000
SA_REP	Smith	8000	3000

## Bekerja dengan Tanggal

Sql mengenali format tanggal dimana tahun yang dituliskan dua digit akan dikenali seperti berikut ini:

- Untuk tahun sistem yg berlaku 0 – 49, maka jika dua digit tahun pada tanggal berkisar 0-49, dimaknai tahun pada abad sama dengan abad tahun sistem berkisar 50-99, dimaknai tahun pada abad sebelum abad tahun sistem
- Untuk tahun sistem yg berlaku 50 – 99, maka jika dua digit tahun pada tanggal berkisar 0-49, dimaknai tahun pada abad setelah abad thn sistem berkisar 50-99, dimaknai tahun pada abad sama dengan abad tahun sistem

### Contoh:

```
SELECT last_name, hire_date FROM employees  
WHERE hire_date < '01-FEB-88';
```

LAST_NAME	HIRE_DATE
King	17-JUN-87
Whalen	17-SEP-87



Thn sistem 2011 (0-49)  
abad 2001, maka Thn 88  
(50-99) adalah Tahun  
1988 (abad 2000)

# Bekerja dengan Tanggal

Sql mengenali format tanggal dimana tahun yang dituliskan dua digit akan dikenali seperti berikut ini:

- Untuk tahun sistem yg berlaku 0 – 49, maka jika dua digit tahun pada tanggal berkisar 0-49, dimaknai tahun pada abad sama dengan abad tahun sistem berkisar 50-99, dimaknai tahun pada abad sebelum abad tahun sistem
- Untuk tahun sistem yg berlaku 50 – 99, maka jika dua digit tahun pada tanggal berkisar 0-49, dimaknai tahun pada abad setelah abad thn sistem berkisar 50-99, dimaknai tahun pada abad sama dengan abad tahun sistem

## Contoh:

```
SELECT last_name, hire_date FROM employees  
WHERE hire_date < '01-FEB-05;
```

LAST_NAME	HIRE_DATE
King	17-JUN-87
Kochhar	21-SEP-89
De Haan	13-JAN-93
Hunold	03-JAN-90
Ernst	21-MAY-91
Austin	25-JUN-97
Pataballa	05-FEB-98
Lorentz	07-FEB-99



Thn sistem 2011 (0-49)  
abad 2001, maka Thn 05  
(0-49) adalah Tahun 2005  
(abad 2001)

# Bekerja dengan Tanggal

Jika Tahun Sistem yang berlaku misalnya 1999, maka perintah yang sama dengan Contoh sebelumnya akan menghasilkan makna thn sebagai berikut.

## Contoh:

```
SELECT last_name, hire_date FROM employees  
WHERE hire_date < '01-FEB-88';
```

LAST_NAME	HIRE_DATE
King	17-JUN-87
Whalen	17-SEP-87



Thn sistem 1999 (50-99)  
abad 2000, maka Thn 88  
(50-99) adalah Tahun  
1988 (abad 2000)

## Contoh:

```
SELECT last_name, hire_date FROM employees  
WHERE hire_date < '01-FEB-05';
```

LAST_NAME	HIRE_DATE
King	17-JUN-87
Kochhar	21-SEP-89
De Haan	13-JAN-93
Hunold	03-JAN-90
Ernst	21-MAY-91
Austin	25-JUN-97
Pataballa	05-FEB-98
Lorentz	07-FEB-99



Thn sistem 1999 (50-99)  
abad 2000, maka Thn 05  
(0-49) adalah Tahun 2005  
(abad 2001)



# Arithmetic with DATE

- Menambahkan suatu angka dari hari ke suatu tanggal
- Mengurangi suatu angka dari hari dari suatu tanggal
- Mengurangkan suatu tanggal dari tanggal yang lain
- Menambahkan suatu angka dari jam ke suatu tanggal

```
SELECT hire_date, hire_date+2, hire_date-2, Round((Sysdate - hire_date) / 7,2) AS
WEEKS, hire_date + 90/24 FROM employees
```

HIRE_DATE	HIRE_DATE+2	HIRE_DATE-2	WEEKS	HIRE_DATE+90/24
17-JUN-87	19-JUN-87	15-JUN-87	614.04	20-JUN-87
21-SEP-89	23-SEP-89	19-SEP-89	495.9	24-SEP-89
13-JAN-93	15-JAN-93	11-JAN-93	323.04	16-JAN-93
03-JAN-90	05-JAN-90	01-JAN-90	481.04	06-JAN-90
21-MAY-91	23-MAY-91	19-MAY-91	409.19	24-MAY-91
25-JUN-97	27-JUN-97	23-JUN-97	91.04	28-JUN-97
05-FEB-98	07-FEB-98	03-FEB-98	58.9	08-FEB-98
07-FEB-99	09-FEB-99	05-FEB-99	6.47	10-FEB-99

## Date Functions

Function	Result
<b>MONTHS_BETWEEN</b>	Number of months between two dates
<b>ADD_MONTHS</b>	Add calendar months to date
<b>NEXT_DAY</b>	Next day of the date specified
<b>LAST_DAY</b>	Last day of the month
<b>ROUND</b>	Round date
<b>TRUNC</b>	Truncate date

Function	Result
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'

Assume SYSDATE = '25-JUL-03':

Function	Result
ROUND (SYSDATE, 'MONTH')	01-AUG-03
ROUND (SYSDATE , 'YEAR')	01-JAN-04
TRUNC (SYSDATE , 'MONTH')	01-JUL-03
TRUNC (SYSDATE , 'YEAR')	01-JAN-03

Latihan :

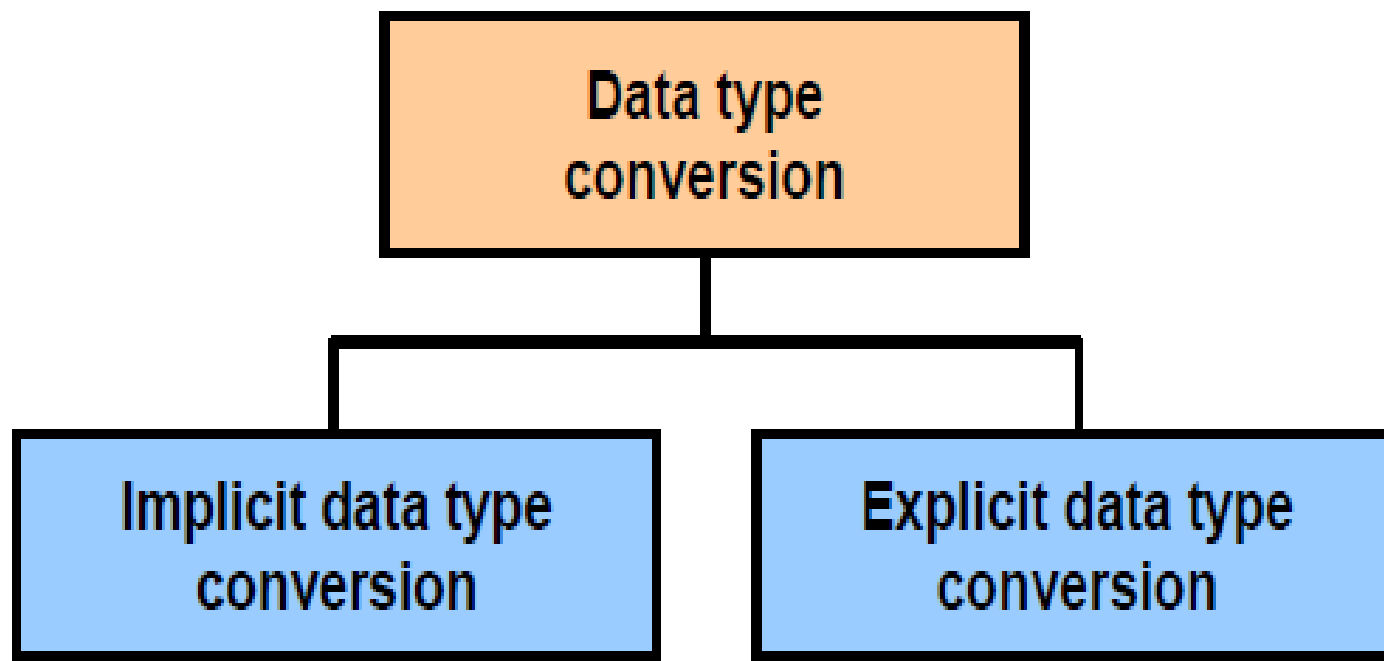
1. Tuliskan perintah dalam SQL untuk menampilkan hire\_date, menampilkan gabungan first\_name dan last\_name dimana first\_name dalam huruf kecil, dan last\_name dalam huruf kapital dari tabel Employees untuk baris (record) yang memiliki hire\_date diantara '2-Feb-1980' s/d '2-Feb-1990'.

HIRE_DATE	NAMA
17-JUN-87	stevenKING
21-SEP-89	neenaKOCHHAR
03-JAN-90	alexanderHUNOLD
17-SEP-87	jenniferWHALEN

2. Tuliskan perintah dalam SQL untuk menampilkan first\_name, dua huruf pertama dari Job\_id, panjang karakter pada Email dari tabel Employees untuk baris (record) yang memiliki karakter 'e' di tengah first\_name.

FIRST_NAME	JOB	EMAIL
Steven	AD	5
Neena	AD	8
Lex	AD	7
Alexander	IT	7
Daniel	FI	7
Ismael	FI	8
Jose Manuel	FI	7
Den	PU	8
Alexander	PU	5
Shelli	PU	6

# Conversion Functions



# Implicit Data Type Conversion

For assignments, the Oracle server can automatically convert the following:

From	To
VARCHAR2 or CHAR	NUMBER
VARCHAR2 or CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2

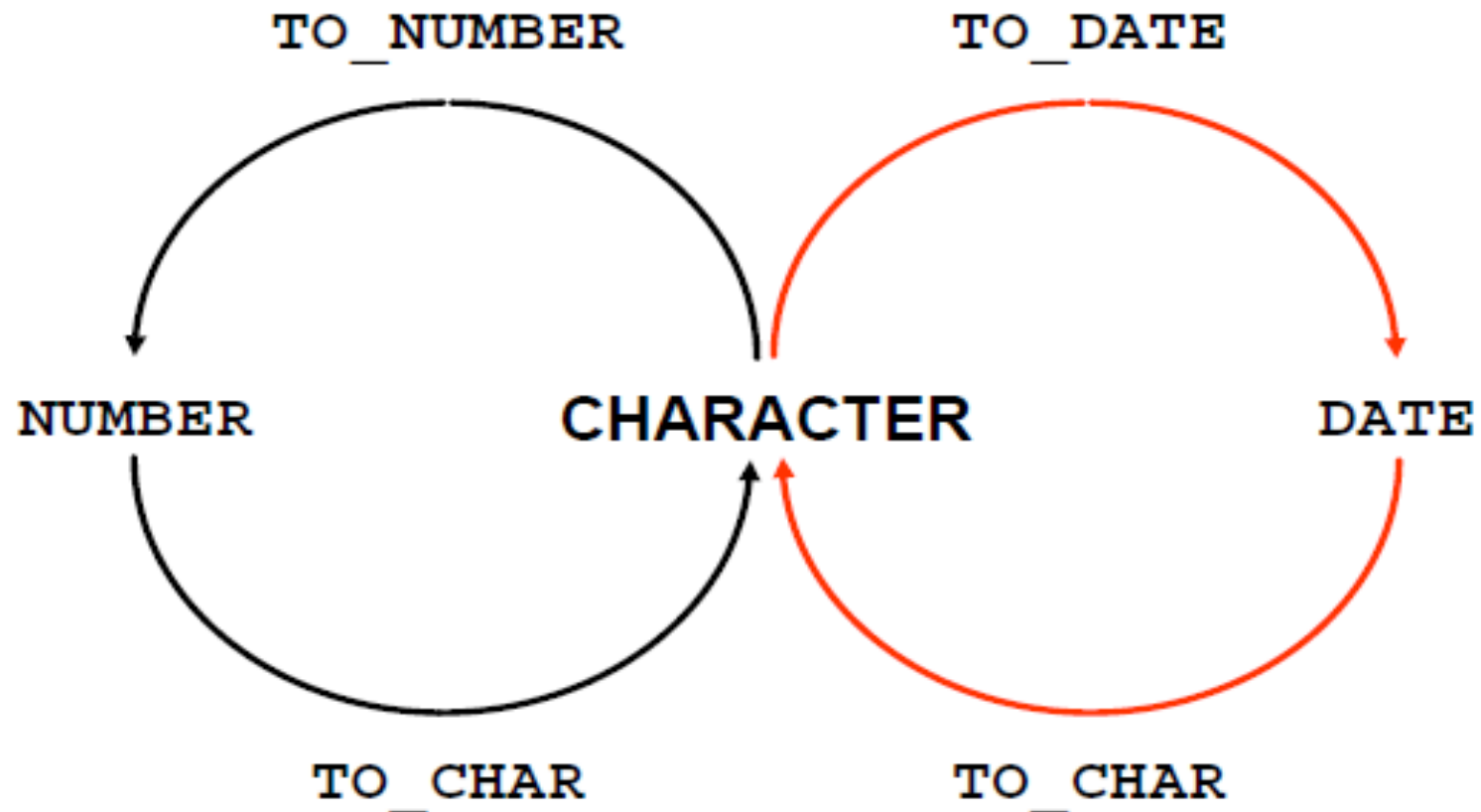
```
select 'Gaji ' || First_name || ' adalah ' || salary from employees
```

'GAJI'    FIRST_NAME    'ADALAH'    SALARY
Gaji Steven adalah 24000
Gaji Neena adalah 17000
Gaji Lex adalah 17000
Gaji Alexander adalah 9000
Gaji Bruce adalah 6000
Gaji David adalah 4800
Gaji Valli adalah 4800
Gaji Diana adalah 4200

```
select first_name || ' ' || hire_date from employees
```

FIRST_NAME    ' '    HIRE_DATE
Steven 17-JUN-87
Neena 21-SEP-89
Lex 13-JAN-93
Alexander 03-JAN-90
Bruce 21-MAY-91
David 25-JUN-97
Valli 05-FEB-98
Diana 07-FEB-99

# Explicit Data Type Conversion





```
select 'Gaji ' || First_name || ' adalah ' || to_char(salary) from employees
```

'GAJI'    FIRST_NAME    'ADALAH'    SALARY
Gaji Steven adalah 24000
Gaji Neena adalah 17000
Gaji Lex adalah 17000
Gaji Alexander adalah 9000
Gaji Bruce adalah 6000
Gaji David adalah 4800
Gaji Valli adalah 4800
Gaji Diana adalah 4200

```
select first_name || ' ' || to_char(hire_date ) from employees
```

FIRST_NAME    ' '    HIRE_DATE
Steven 17-JUN-87
Neena 21-SEP-89
Lex 13-JAN-93
Alexander 03-JAN-90
Bruce 21-MAY-91
David 25-JUN-97
Valli 05-FEB-98
Diana 07-FEB-99

## Using the TO\_CHAR Function with Dates

```
TO_CHAR(date, 'format_model')
```

Element	Result
YYYY	Full year in numbers
YEAR	Year spelled out (in English)
MM	Two-digit value for month
MONTH	Full name of the month
MON	Three-letter abbreviation of the month
DY	Three-letter abbreviation of the day of the week
DAY	Full name of the day of the week
DD	Numeric day of the month

```

SELECT first_name, hire_date,
TO_CHAR(hire_date, 'DD Month YYYY') AS HIREDATE1,
TO_CHAR(hire_date, 'DD Mon Year') AS HIREDATE2,
TO_CHAR(hire_date, 'Day') AS HIREDATE3,
TO_CHAR(hire_date, 'DY DD-MM-YY') AS HIREDATE4
FROM employees;

```

FIRST_NAME	HIRE_DATE	HIREDATE1	HIREDATE2	HIREDATE3	HIREDATE4
Steven	06/17/2003	17 June 2003	17 Jun Two Thousand Three	Tuesday	TUE 17-06-03
Neena	09/21/2005	21 September 2005	21 Sep Two Thousand Five	Wednesday	WED 21-09-05
Lex	01/13/2001	13 January 2001	13 Jan Two Thousand One	Saturday	SAT 13-01-01
Alexander	01/03/2006	03 January 2006	03 Jan Two Thousand Six	Tuesday	TUE 03-01-06
Bruce	05/21/2007	21 May 2007	21 May Two Thousand Seven	Monday	MON 21-05-07
David	06/25/2005	25 June 2005	25 Jun Two Thousand Five	Saturday	SAT 25-06-05
Valli	02/05/2006	05 February 2006	05 Feb Two Thousand Six	Sunday	SUN 05-02-06
Diana	02/07/2007	07 February 2007	07 Feb Two Thousand Seven	Wednesday	WED 07-02-07
Nancy	08/17/2002	17 August 2002	17 Aug Two Thousand Two	Saturday	SAT 17-08-02
Daniel	08/16/2002	16 August 2002	16 Aug Two Thousand Two	Friday	FRI 16-08-02
More than 10 rows available. Increase rows selector to view more rows.					

10 rows returned in 0.05 seconds

[Download](#)

# Using the TO\_CHAR Function with Numbers

```
TO_CHAR(number, 'format_model') 
```

These are some of the format elements that you can use with the TO\_CHAR function to display a number value as a character:

Element	Result
9	Represents a number
0	Forces a zero to be displayed
\$	Places a floating dollar sign
L	Uses the floating local currency symbol
.	Prints a decimal point
,	Prints a comma as thousands indicator

```

SELECT first_name, salary,
       TO_CHAR(salary, '$99,999.00') AS SALARY1,
       TO_CHAR(salary, '$00,000.00') AS SALARY2
FROM employees

```

FIRST_NAME	SALARY	SALARY1	SALARY2
Steven	24000	\$24,000.00	\$24,000.00
Neena	17000	\$17,000.00	\$17,000.00
Lex	17000	\$17,000.00	\$17,000.00
Alexander	9000	\$9,000.00	\$09,000.00
Bruce	6000	\$6,000.00	\$06,000.00
David	4800	\$4,800.00	\$04,800.00
Valli	4800	\$4,800.00	\$04,800.00
Diana	4200	\$4,200.00	\$04,200.00
Nancy	12008	\$12,008.00	\$12,008.00
Daniel	9000	\$9,000.00	\$09,000.00


## TO\_NUMBER & TO\_DATE

```
SELECT first_name, salary, salary+TO_NUMBER('10000'), hire_date, hire_date -  
TO_DATE('2-Feb-80') FROM employees
```

FIRST_NAME	SALARY	SALARY+TO_NUMBER('10000')	HIRE_DATE	HIRE_DATE-TO_DATE('2-FEB-80')
Steven	24000	34000	17-JUN-87	2892
Neena	17000	27000	21-SEP-89	3519
Lex	17000	27000	13-JAN-93	4729
Alexander	9000	19000	03-JAN-90	3823
Bruce	8000	18000	21-MAY-91	4128
David	4800	14800	25-JUN-97	6353
Valli	4800	14800	05-FEB-98	6578
Diana	4200	14200	07-FEB-99	6945

# General Functions

The following functions work with any data type and pertain to using nulls:

- NVL (expr1, expr2)
  - NVL2 (expr1, expr2, expr3)
  - NULLIF (expr1, expr2)
  - COALESCE (expr1, expr2, ..., exprn)
- 

## GENERAL FUNCTIONS

Fungsi-Fungsi yang berkaitan dengan berbagai tipe data mengenai nilai Null.

- **NVL (expr1, expr2)**

Jika Expr1 bukan Null, maka Output berupa Expr1

Jika Expr1 bernilai Null, maka Output berupa Expr2

- **NVL2 (expr1, expr2, expr3)**

Jika Expr1 bukan Null, maka Output berupa Expr2

Jika Expr1 bernilai Null, maka Output berupa Expr3

- **NULLIF (expr1, expr2)**

Jika Expr1 = Expr2, maka Output berupa Null

Jika Expr1 ≠ Expr2, maka Output berupa Expr1

- **COALESCE (expr1, expr2, ..., exprn)**

Jika Expr1, Expr2, ... Exprn-1 semuanya Null, maka Output adl Exprn

Sebaliknya, maka Output berupa salah satu Expr1, Expr2, .... Exprn-1 yang tidak null (diprioritaskan Expr yang terkiri)



```

SELECT first_name,last_name,manager_id, commission_pct,
NVL (commission_pct, 0) AS KOMISI1,
NVL2 (commission_pct, Salary+Salary*Commission_pct ,0) AS KOMISI2,
NULLIF (Length(first_name),Length (last_name)) AS HASIL1,
COALESCE(manager_id, commission_pct, -1) AS HASIL2
FROM employees

```

FIRST_NAME	LAST_NAME	MANAGER_ID	COMMISSION_PCT	KOMISI1	KOMISI2	HASIL1	HASIL2
Steven	King	-	-	0	0	6	-1
Neena	Kochhar	100	-	0	0	5	100
Lex	De Haan	100	-	0	0	3	100
Alexander	Hunold	102	-	0	0	9	102
Bruce	Ernst	103	-	0	0	-	103
David	Austin	103	-	0	0	5	103
Valli	Pataballa	103	-	0	0	5	103
Diana	Lorentz	103	-	0	0	5	103
Nancy	Greenberg	101	-	0	0	5	101
Daniel	Faviet	108	-	0	0	-	108
John	Chen	108	-	0	0	-	108

# Conditional Expressions

- Provide the use of **IF-THEN-ELSE** logic within a SQL statement
- Use two methods:
  - **CASE** expression
  - **DECODE** function



# CASE Expression

Facilitates conditional inquiries by doing the work of an IF-THEN-ELSE statement:

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

## Using the CASE Expression

Facilitates conditional inquiries by doing the work of an IF-THEN-ELSE statement:

```
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	5800	5800
Rajs	ST_CLERK	3500	4025
...			
Gietz	AC_ACCOUNT	8300	8300

## DECODE Function

Facilitates conditional inquiries by doing the work of a CASE expression or an IF-THEN-ELSE statement:

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

## Using the DECODE Function

```
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;
```

LAST_NAME	JOB_ID	SALARY	REVISED_SALARY
...			
Lorentz	IT_PROG	4200	4620
Mourgos	ST_MAN	5800	5800
Rajs	ST_CLERK	3600	4025
...			
Gietz	AC_ACCOUNT	8300	8300

Latihan :

1. Tuliskan query untuk menampilkan manager\_id, menampilkan hire\_date dalam format seperti 17/Jun/99 lalu digabungkan dengan Email (dalam huruf kecil), menampilkan salary dalam format seperti Rp. 07,000 dari tabel employees untuk baris (record) yang memiliki manager\_id bukan 100,102 dan memiliki Email yang mengandung huruf 'K' di awal Email.

MANAGER_ID	GABUNG1	GABUNG2
122	14/JUN/05kchung	Rp. 03,800
114	10/AUG/07kcolmena	Rp. 02,500
124	23/MAY/06kfeeney	Rp. 03,000
122	12/DEC/07kgee	Rp. 02,400
149	24/MAY/07kgrant	Rp. 07,000

2. Tuliskan perintah dalam SQL untuk menampilkan employee\_id, first\_name, salary, job\_id, bonus dengan ketentuan sebagai berikut:  
Untuk dua huruf pertama dari Job\_id jika bernilai  
    'IT' diberi bonus = 10% \* salary  
    'AD' diberi bonus = 20% \* salary  
    Selain 'IT' dan 'AD' diberi bonus = 0  
dari tabel employees untuk baris (record) yang memiliki employee\_id < 110

EMPLOYEE_ID	FIRST_NAME	SALARY	JOB_ID	BONUS
100	Steven	24000	AD_PRES	4800
101	Neena	17000	AD_VP	3400
102	Lex	17000	AD_VP	3400
103	Alexander	9000	IT_PROG	900
104	Bruce	6000	IT_PROG	600
105	David	4800	IT_PROG	480
106	Valli	4800	IT_PROG	480
107	Diana	4200	IT_PROG	420
108	Nancy	12008	FI_MGR	0
109	Daniel	9000	FI_ACCOUNT	0