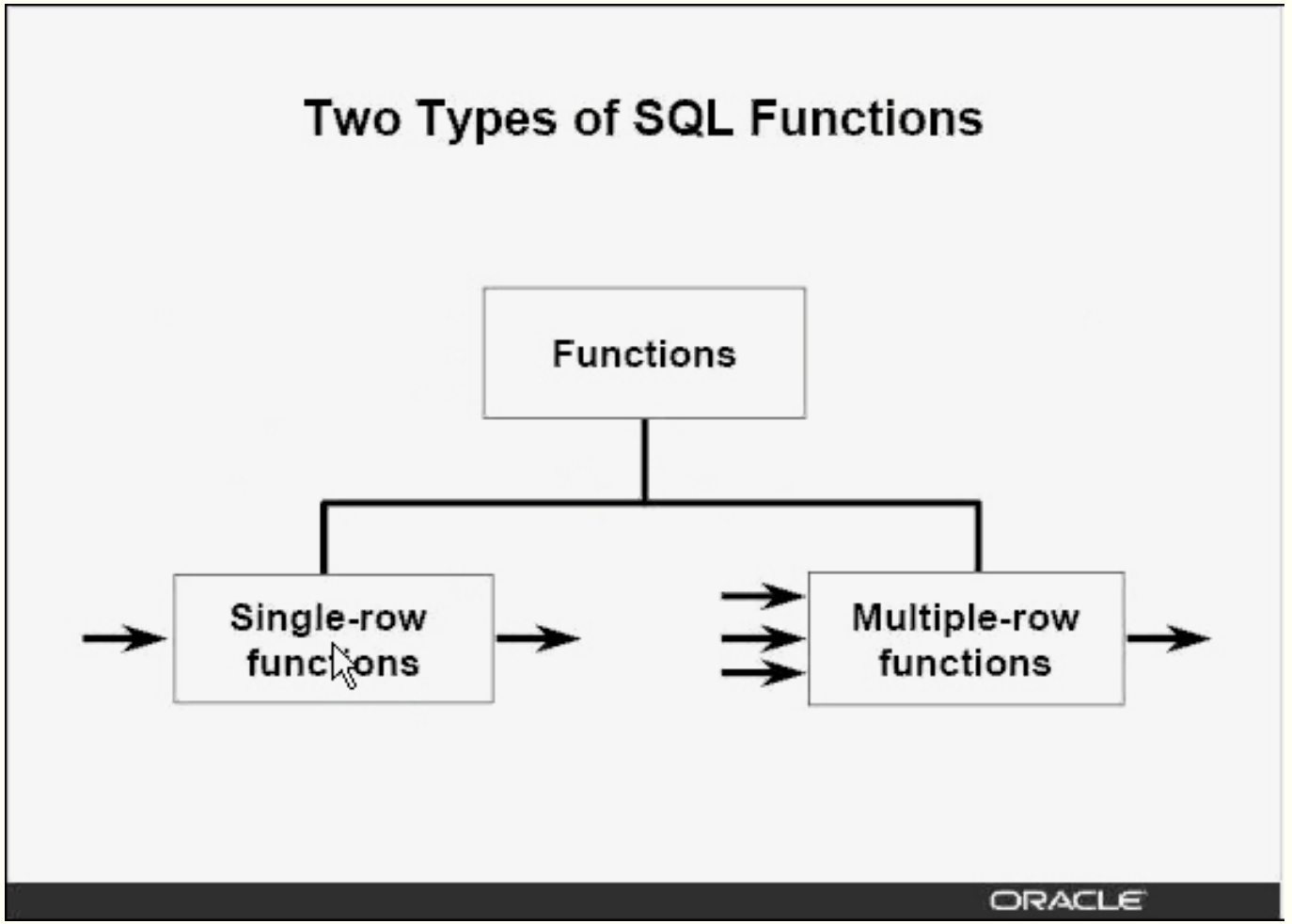
FUNCTIONS

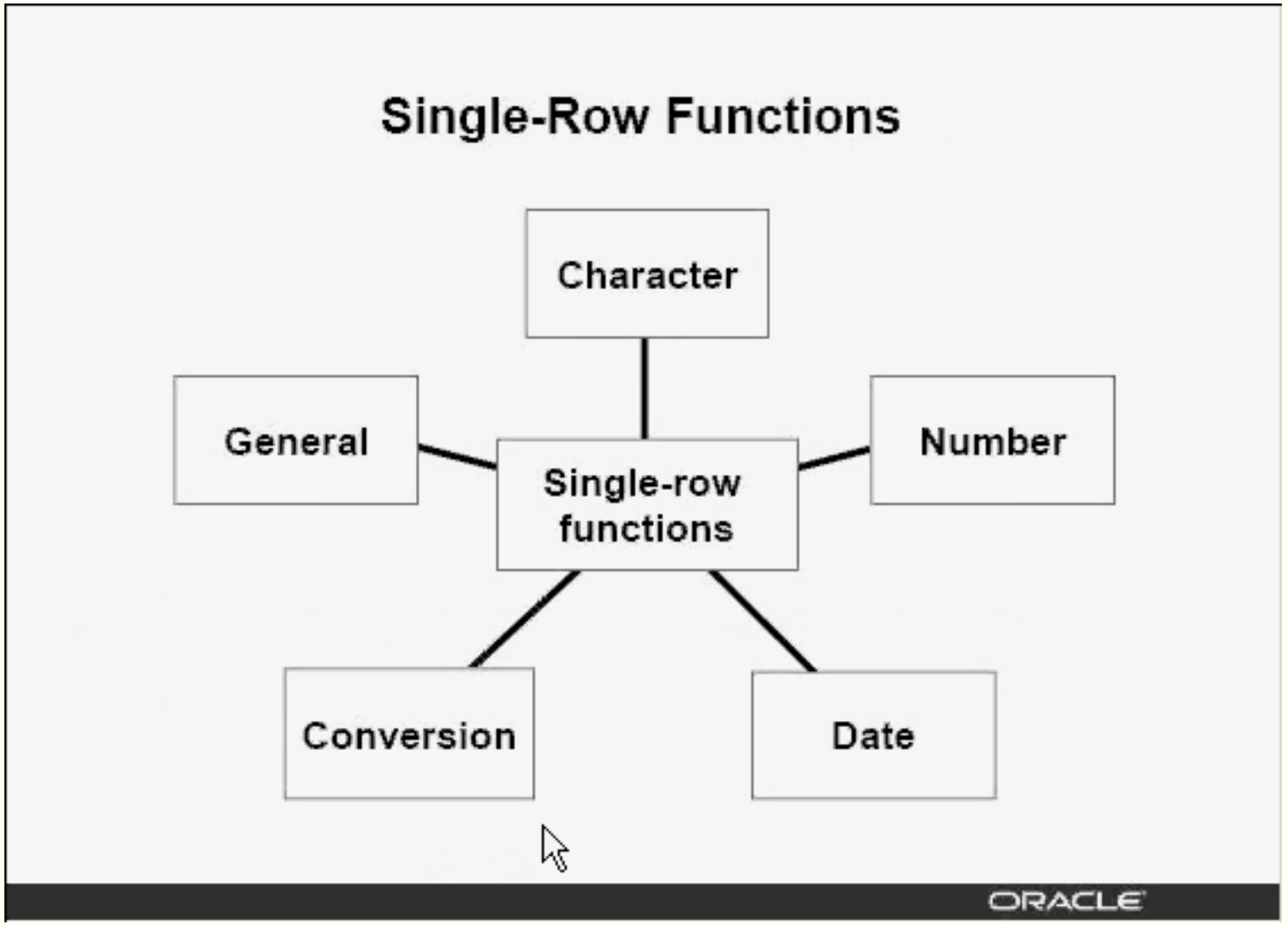
Dr. Mohammed A. Mohammed

University of Sulaimani | College of science | Computer dept.

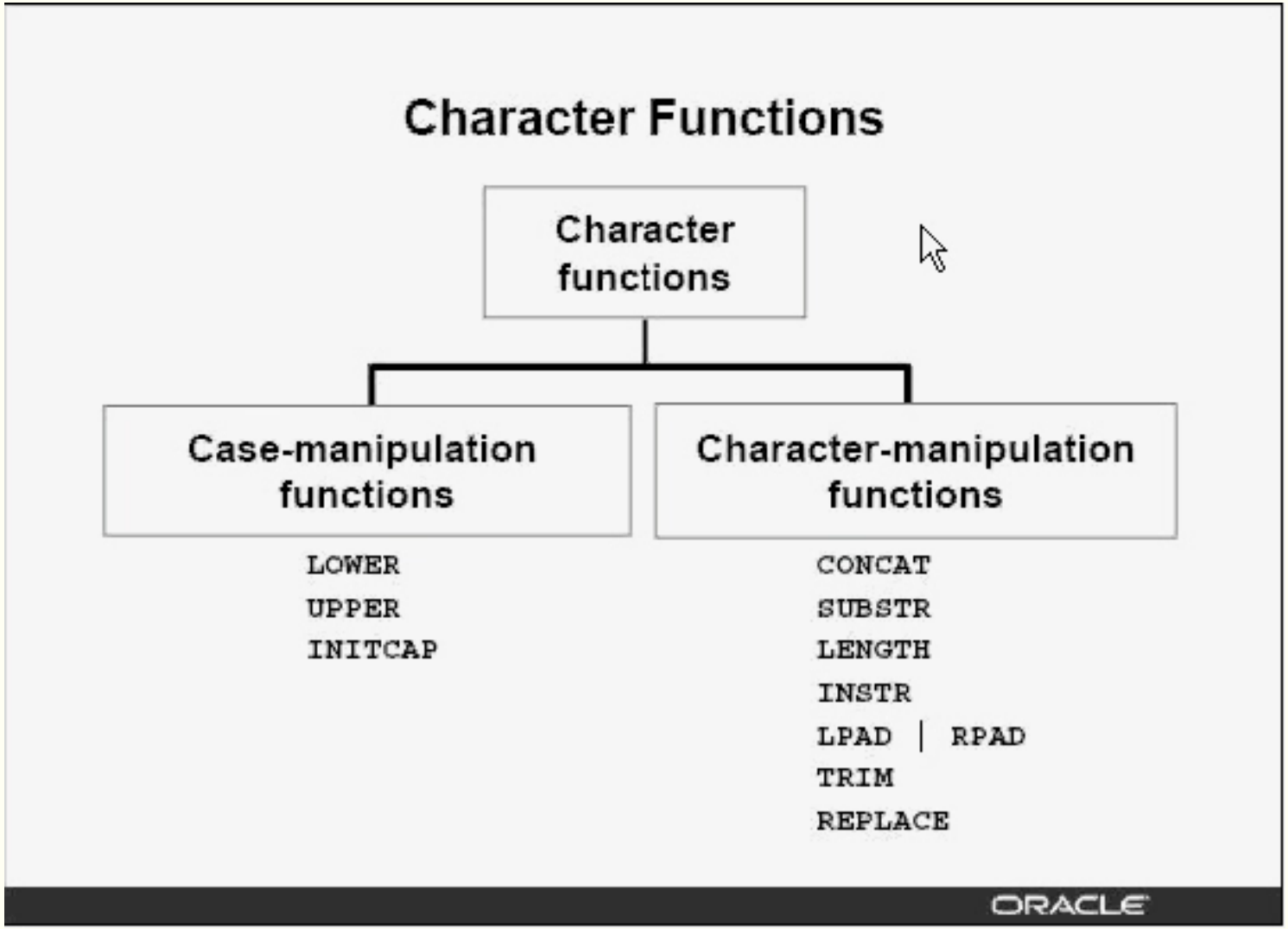
TYPES OF SQL FUNCTIONS



SINGLE ROW FUNCTIONS



CHARACTER FUNCTIONS



Case Manipulation Functions

These functions convert case for character strings.

Function	Result
`LOWER('SQL Course')`	`sql course`
`UPPER('SQL Course')`	`SQL COURSE`
`INITCAP('SQL Course')`	`Sql Course`

```
SELECT employee_id,upper(last_name),department_id FROM employees
WHERE last_name='higgins‘
```

no data found

```
SELECT employee_id,upper(last_name),department_id FROM employees
WHERE lower(last_name)='higgins‘
```

EMPLOYEE_ID	UPPER(LAST_NAME)	DEPARTMENT_ID
205	HIGGINS	110

Character-Manipulation Functions

These functions manipulate character strings:

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*****
TRIM('H' FROM 'HelloWorld')	elloWorld

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'?
150	PeterTucker	SA_REP	6	0
151	DavidBernstein	SA_REP	9	0
152	PeterHall	SA_REP	4	2

SELECT employee_id,CONCAT(first_name,last_name) NAME, job_id,LENGTH(last_name), INSTR(last_name,'a') "contains 'a'?“ FROM employees WHERE substr(last_name,-1,1)=‘n'

EMPLOYEE_ID	NAME	JOB_ID	LENGTH(LAST_NAME)	Contains 'a'?
102	LexDe Haan	AD_VP	7	5
105	DavidAustin	IT_PROG	6	0
110	JohnChen	FI_ACCOUNT	4	0

Number Functions

- ROUND: Rounds value to specified decimal

  ROUND(45.926, 2) → 45.93

- TRUNC: Truncates value to specified decimal

  TRUNC(45.926, 2) → 45.92

- MOD: Returns remainder of division

  MOD(1600, 300) → 100

NUMBER FUNCTIONS

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 MOD(9,2) FROM dual

MOD(9,2)
1

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

LAST_NAME	SALARY	MOD(SALARY,5000)
Tucker	10000	0
Bernstein	9500	4500
Hall	9000	4000

DATE

Operation	Result	Description
date + number	Date	Adds a number of days to a date
date - number	Date	Subtracts a number of days from a date
date - date	Number of days	Subtracts one date from another
date + number/24	Date	Adds a number of hours to a date

SELECT sysdate FROM dual

SYSDATE
09-03-15

SELECT last_name,(sysdate-hire_date)/7 as week FROM employees WHERE department_id=90

LAST_NAME	WEEK
King	1134,62952380952380952380952380952380952
Kochhar	1016,48666666666666666666666666666666667
De Haan	843,629523809523809523809523809523809524

Date Functions

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

Using Date Functions

- 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'

Using Date Functions

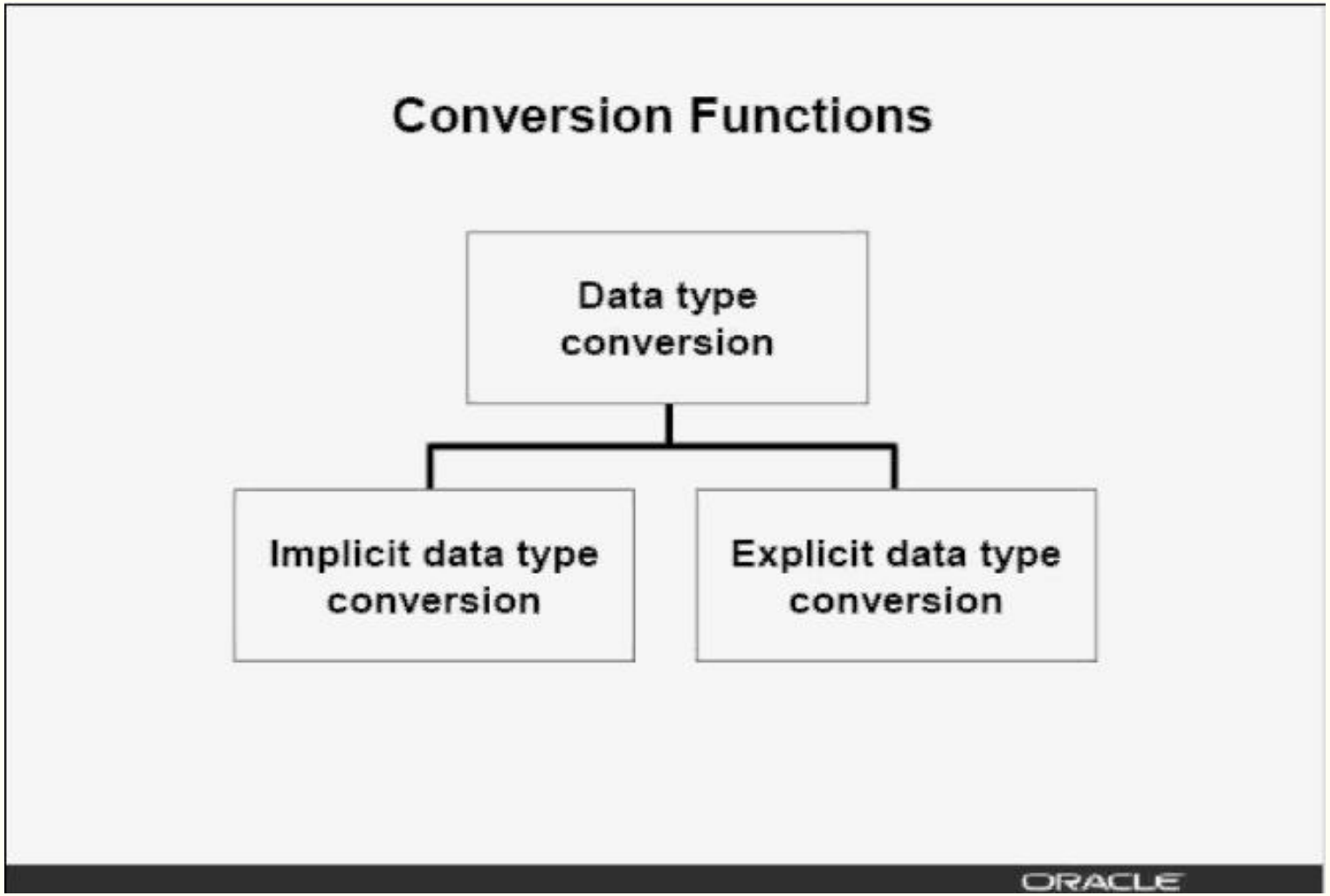
Assume SYSDATE = '25-JUL-95':

- ROUND(SYSDATE,'MONTH') → 01-AUG-95
- ROUND(SYSDATE ,'YEAR') → 01-JAN-96
- TRUNC(SYSDATE ,'MONTH') → 01-JUL-95
- TRUNC(SYSDATE ,'YEAR') → 01-JAN-95

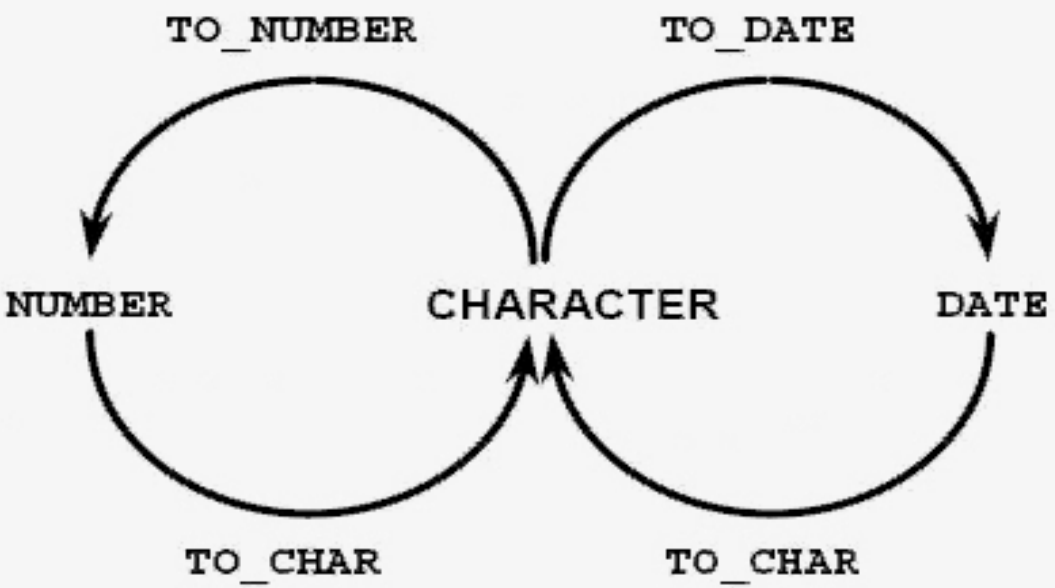
```
SELECT employee_id,hire_date,MONTHS_BETWEEN(sysdate,hire_date),
ADD_MONTHS(hire_date,6),NEXT_DAY(hire_date,7), LAST_DAY(hire_date)
FROM employees WHERE MONTHS_BETWEEN(sysdate,hire_date)>36
```

EMPLOYEE_ID	HIRE_DATE	MONTHS_BETWEEN(SYSDATE,HIRE_DATE)	ADD_MONTHS(HIRE_DATE,6)	NEXT_DAY(HIRE_DATE,7)	LAST_DAY(HIRE_DAT
100	87-06-17	260,952710946833930704898446833930704898	87-12-17	87-06-20	87-06-30
101	89-09-21	233,823678688769414575866188769414575866	90-03-21	89-09-23	89-09-30
102	93-01-13	194,081743204898446833930704898446833931	93-07-13	93-01-16	93-01-31

CONVERSION FUNCTIONS



Explicit Data Type Conversion



CONVERSION FUNCTIONS

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

EMPLOYEE_ID	TO_CHAR(HIRE_DATE,'MM/YY')
205	06/94

Elements of the Date Format Model

YYYY	Full year in numbers
YEAR	Year spelled out
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

VALID DATE FORMATS

Sample Format Elements of Valid Date Formats

Element	Description
SCC or CC	Century; server prefixes B.C. date with -
Years in dates YYYY or SYYYY	Year; server prefixes B.C. date with -
YYY or YY or Y	Last three, two, or one digits of year
Y,YYY	Year with comma in this position
IYYY, IYY, IY, I	Four, three, two, or one digit year based on the ISO standard
SYEAR or YEAR	Year spelled out; server prefixes B.C. date with -
BC or AD	B.C./A.D. indicator
B.C. or A.D.	B.C./A.D. indicator with periods
Q	Quarter of year
MM	Month: two-digit value
MONTH	Name of month padded with blanks to length of nine characters
MON	Name of month, three-letter abbreviation
RM	Roman numeral month
WW or W	Week of year or month
DDD or DD or D	Day of year, month, or week
DAY	Name of day padded with blanks to a length of nine characters
DY	Name of day; three-letter abbreviation
J	Julian day; the number of days since 31 December 4713 B.C.

VALID DATE FORMATS (CONT.)

Element	Description
AM or PM	Meridian indicator
A.M. or P.M.	Meridian indicator with periods
HH or HH12 or HH24	Hour of day, or hour (1–12), or hour (0–23)
MI	Minute (0–59)
SS	Second (0–59)
SSSSS	Seconds past midnight (0–86399)

OTHER FORMATS

Other Formats	
Element	**Description**
/ . ,	Punctuation is reproduced in the result
"of the"	Quoted string is reproduced in the result

Specifying Suffixes to Influence Number Display	
Element	**Description**
TH	Ordinal number (for example, DDTH for 4TH)
SP	Spelled-out number (for example, DDSP for FOUR)
SPTH or THSP	Spelled-out ordinal numbers (for example, DDSPTH for FOURTH)

EXAMPLE

SELECT last_name,to_char(hire_date,'DD MONTH YYYY')
FROM employees

LAST_NAME	TO_CHAR(HIRE_DATE,'DDMONTHYYYY')
King	17 JUNE 1987
Kochhar	21 SEPTEMBER 1989
De Haan	13 JANUARY 1993

SELECT last_name,to_char(hire_date,'Ddspth "of" Month YYYY HH:MI:ss AM') FROM employees

LAST_NAME	TO_CHAR(HIRE_DATE,'DDSPTH"OF"MONTHYYYYHH:MI:SSAM')
King	Seventeenth of June 1987 12:00:00 AM
Kochhar	Twenty-First of September 1989 12:00:00 AM
De Haan	Thirteenth of January 1993 12:00:00 AM

TO_CHAR

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 thousand indicator

OTHER FORMATS

SELECT TO_CHAR(salary,’$99,999.00’),salary FROM employees WHERE last_name=‘Ernst’

TO_CHAR(SALARY,'$99,999.00')	SALARY
$6,000.00	6000

SELECT last_name,hire_date FROM employees WHERE hire_date = TO_DATE('05 24, 1999','MM DD, YY')

LAST_NAME	HIRE_DATE
Grant	99-05-24

General Functions

These 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)`

- **NVL(**String1, expr**)**

✓If **string1** is null then replace it with **expr.**

- **NVL2(**String1, value_if_not_null, value_if_null**)**

- **NULLIF(**expr1, expr2**)**

✓It compares **expr1** and **expr2**. If **expr1** and **expr2** are equal it returns **null**. Otherwise, it returns **expr1**.

✓Parameters or arguments must be either numeric values or values that are of the same datatype.

- **COALESCE** is a generalization of the **NVL** function. It is used as a variety of the CASE expression.

OTHER FORMATS

SELECTlast_name,salary,commission_pct,(salary*12),
(salary*12*commission_pct) FROM employees

LAST_NAME	SALARY	COMMISSION_PCT	(SALARY*12)	(SALARY*12*COMMISSION_PCT)
King	24000	-	288000	-
Kochhar	17000	-	204000	-
De Haan	17000	-	204000	-

SELECT last_name,salary,commission_pct,(salary*12),
(salary*12*NVL(commission_pct,0)) FROM employees

LAST_NAME	SALARY	COMMISSION_PCT	(SALARY*12)	(SALARY*12*NVL(COMMISSION_PCT,0))
King	24000	-	288000	0
Kochhar	17000	-	204000	0
De Haan	17000	-	204000	0

OTHER FORMATS

```
SELECT last_name,salary,commission_pct,
NVL2(commission_pct,'SAL+COMM','SAL') income FROM employees
WHERE department_id in (50,80)
```

LAST_NAME	SALARY	COMMISSION_PCT	INCOME
Weiss	8000	-	SAL
Fripp	8200	-	SAL
Kaufling	7900	-	SAL

OTHER FORMATS

```
SELECT first_name,LENGTH(first_name) "expr1", last_name,
LENGTH(last_name) "expr2", NULLIF(LENGTH(first_name),
LENGTH(last_name)) result FROM employees
```

FIRST_NAME	Expr1	LAST_NAME	Expr2	RESULT
Ellen	5	Abel	4	5
Sundar	6	Ande	4	6
Mozhe	5	Atkinson	8	5

```
SELECT last_name,COALESCE(commission_pct,salary,10)
FROM employees
```

LAST_NAME	COALESCE(COMMISSION_PCT,SALARY,10)
King	24000
Kochhar	17000
De Haan	17000