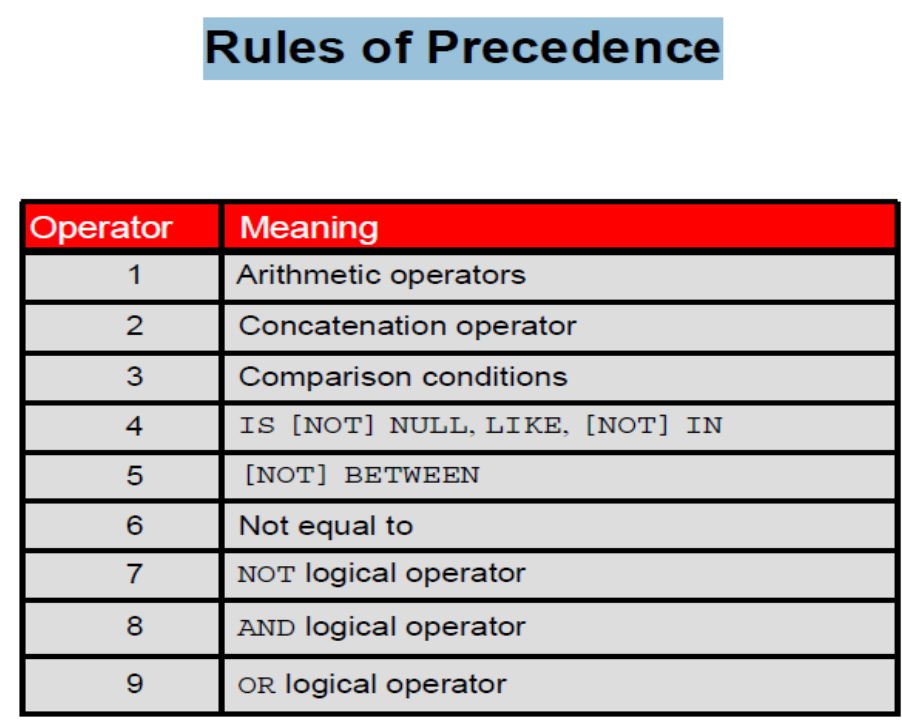
* Oracle standard edition is for small organizations
* Oracle enterprise editions is for huge organizations
* To search for string with underscore

Select \* from Empolyees

Wehre first\_name like ‘s/\_%’ escape ‘/’;

* NOT BETWEEN … AND … , NOT LIKE , IS NOT NULL , NOT IN ( … , … )
* <> and != are both the same and they mean logical not equal and valid in SQL
* When we have successive AND and OR operators in the same level of brackets , SQL looks for AND operator and consider it as one condition
* Precedence rules



* Null values are last in order records
* If you want null first in order records

SELECT \* FROM EMPLOYEES ORDER BY SALARY NULL FIRST;

* We can order by excepression
* We can order by different columns and each one can be DESC or ASC
* Select \* FROM EMPLOYEE ORDER BY FRIST\_NAME

FETCH FIRST 5 ROWS ONLY;

* Select \* FROM EMPLOYEE ORDER BY FRIST\_NAME

FETCH FIRST 50 PERCENT ROWS ONLY;

* Select \* FROM EMPLOYEE ORDER BY FRIST\_NAME

OFFSET 5 FETCH FIRST 5 ROWS ONLY;

* PERCEENT HERE WILL TAKE PERCENT OVER THE TABLE NOT AFTER THE OFFSET
* Select \* FROM EMPLOYEE ORDER BY FRIST\_NAME

OFFSET 5 FETCH FIRST 50 PERCENT ROWS ONLY;

* THIS SELECT WILL SELECT FIRST 2 ITEMS AND IF MORE THAN ONE HAS SAME VALUE WILL ALSO APPEAR SO IT MAY SELECT MORE THAN TWO RECORDS
* Select \* FROM EMPLOYEE ORDER BY FRIST\_NAME

FETCH FIRST 2 ROWS WITH TIES;

* Substitution variables:

SELECT employee\_id, last\_name, salary, department\_id FROM EMPLOYEES

WHERE EMPLOYEE\_ID = &EMPLOYEE\_NUM ;

SELECT employee\_id, first\_name,last\_name, salary, department\_id FROM EMPLOYEES

where FIRST\_NAME='&ename';

SELECT employee\_id, last\_name, job\_id,&column\_name FROM employees

WHERE &condition ORDER BY &ORDER\_COLUMN ;

* We can define substitution vaiable for a whole session with 3 methods:

1. DEFINE key word

DEFINE EMPLOYEE\_NUM = 200;

1. ACCEPT key word – with option to change prompt text

ACCEPT DEPT\_ID PROMPT 'please enter dept id' ;

1. && operator

SELECT employee\_id, last\_name, job\_id, &&column\_name --this =define column\_name

FROM EMPLOYEES ORDER BY &COLUMN\_NAME ;

* Use UNDEFINE to delete pre-defined substitution variable
* Can stop substitution variable definition

SET DEFINE OFF;

SELECT \* FROM DEPARTMENTS WHERE DEPARTMENT\_NAME LIKE '%&t%';

* In SQL , string index starts from 1
* In SQL , string index can be negative to count from end of string

INSTR( string, substring [, start\_position [, th\_appearance ] ] )

Parameters or Arguments

**string**

The string to search. *string* can be CHAR, VARCHAR2, NCHAR, NVARCHAR2, CLOB, or NCLOB.

**substring**

The substring to search for in *string*. *substring* can be CHAR, VARCHAR2, NCHAR, NVARCHAR2, CLOB, or NCLOB.

**start\_position**

Optional. The position in *string* where the search will start. If omitted, it defaults to 1. The first position in the string is 1. If the *start\_position* is negative, the INSTR function counts back *start\_position* number of characters from the end of *string* and then searches towards the beginning of *string*.

**nth\_appearance**

Optional. The nth appearance of *substring*. If omitted, it defaults to 1.

LPAD( string1, padded\_length [, pad\_string] )

Parameters or Arguments

**string1**

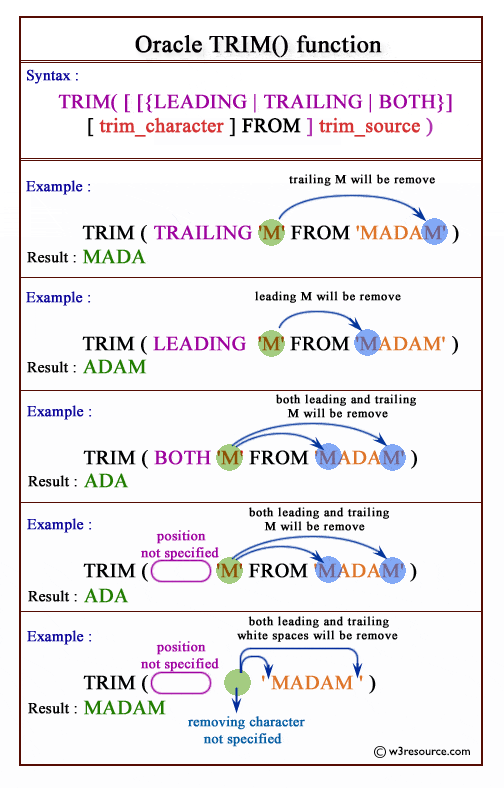
The string to pad characters to (the left-hand side).

**padded\_length**

The number of characters to return. If the *padded\_length* is smaller than the original string, the LPAD function will truncate the string to the size of *padded\_length*.

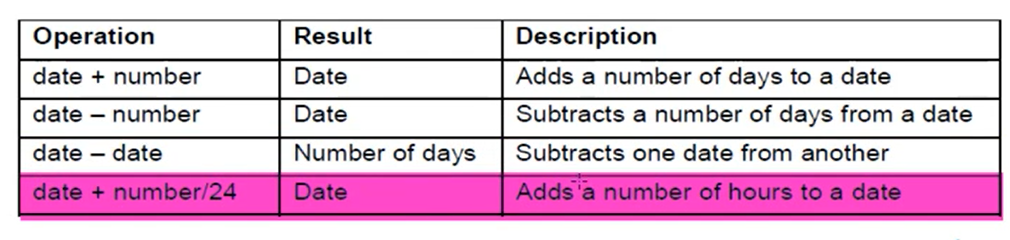
**pad\_string**

Optional. This is the string that will be padded to the left-hand side of *string1*. If this parameter is omitted, the LPAD function will pad spaces to the left-side of *string1*.

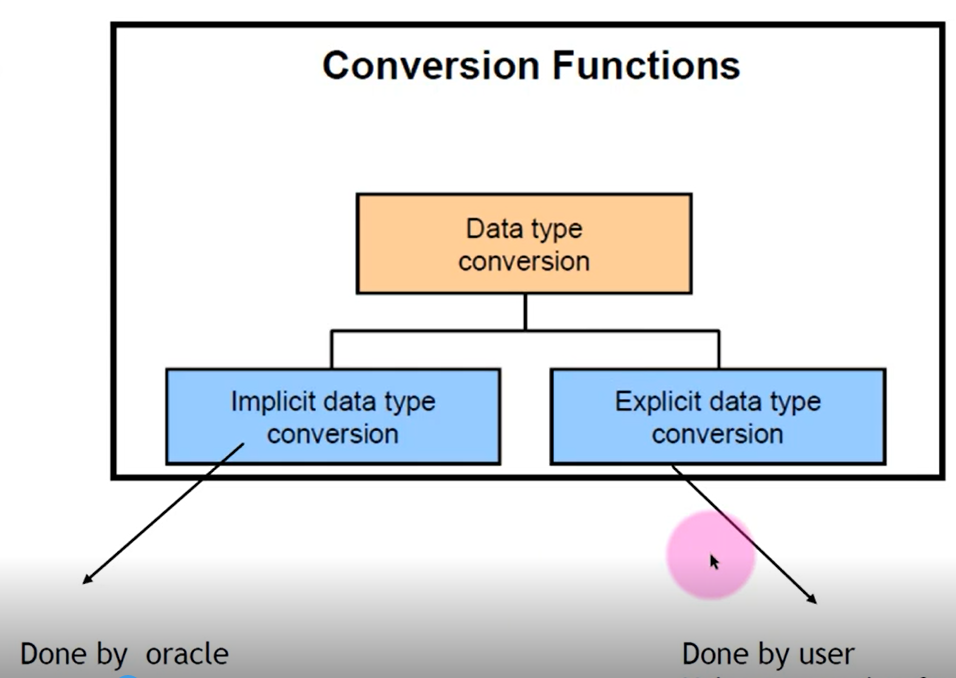


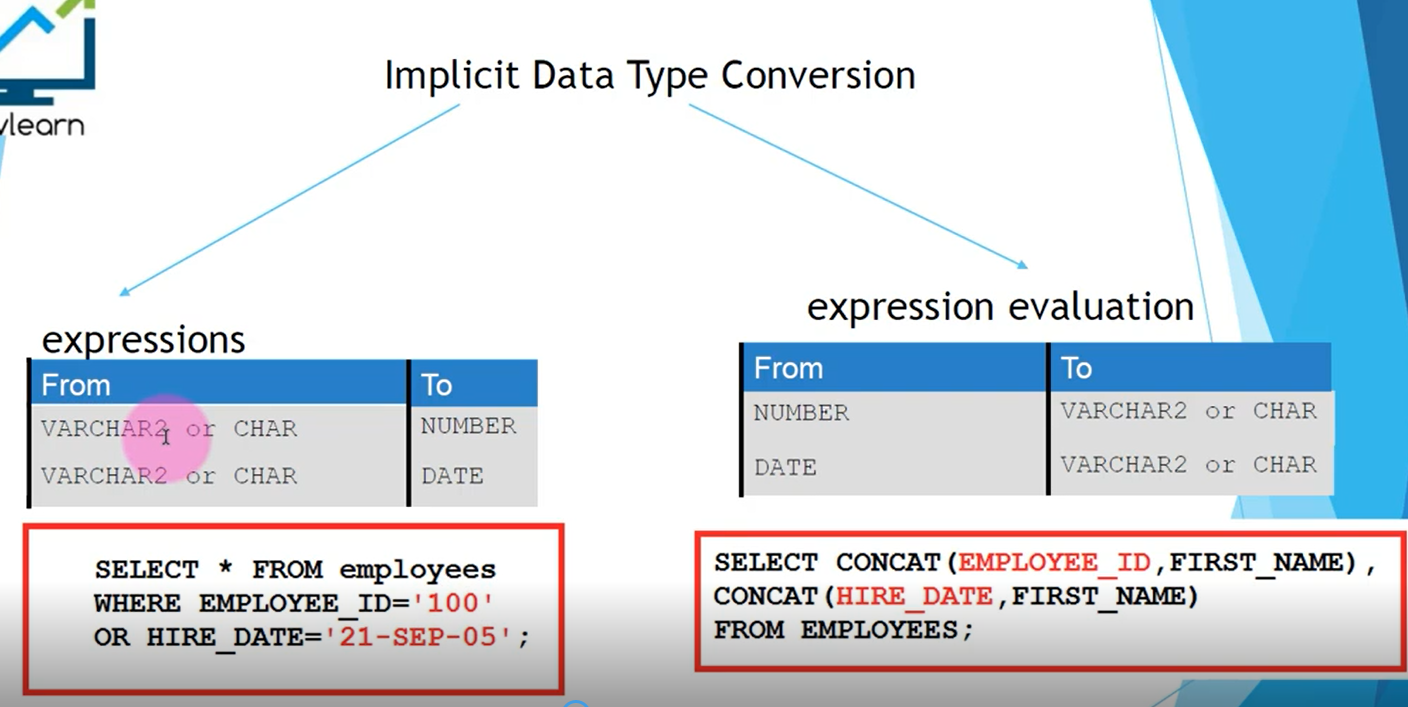
* We can use negative index with Round and TRUNC to indicated hanling digits before the decimal point

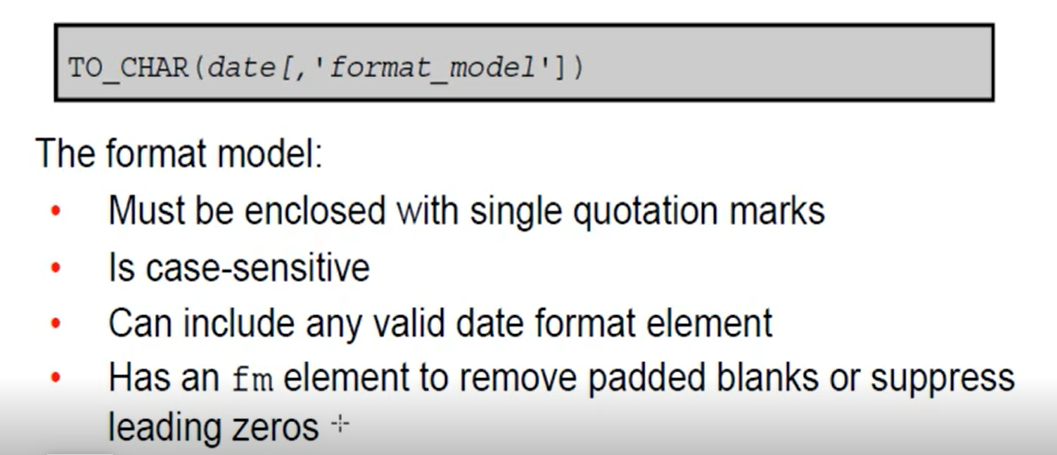
Date conversions

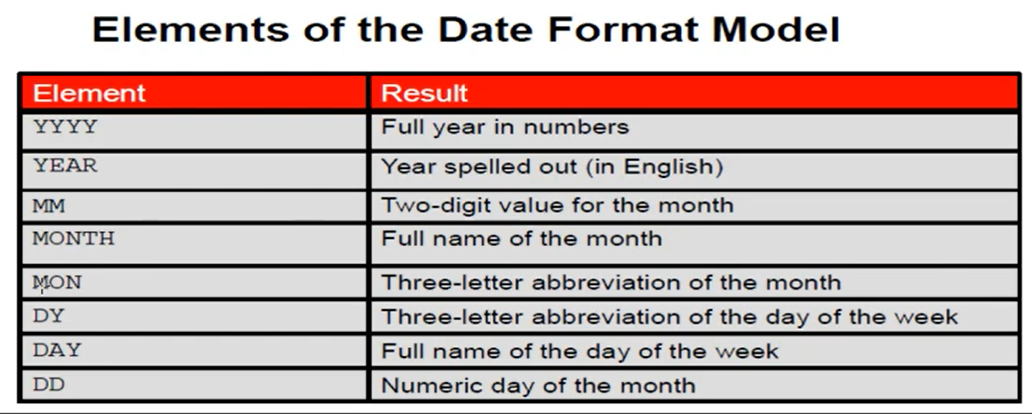


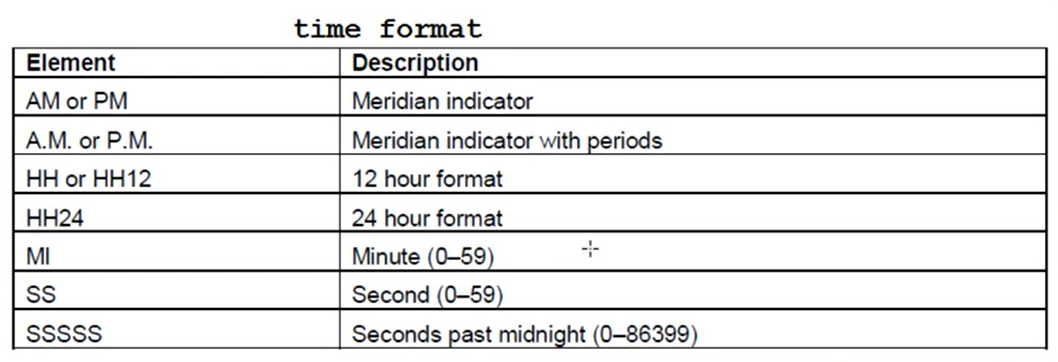
* RR format
  + If the year between 50 – 99 it’s 20th century
  + If the year between 0-49 it’s 21st century
* Months\_between
  + Function that calculate number of months between 2 dates
  + We cannot divide by 30 as not all months are 30 days so it’s not accurate
* Next\_day
  + Returns date of the next valid week day
  + Select next\_day(sysdate , ‘FRIDAY’) from dual;
* LAST\_DAY
  + Returns the last day of the month
* Round ( SYSDATE , ‘MONTH’);
* TRUNC ( SYSDATE , ‘YEAR);
* Round ( SYSDATE , ‘MONTH’);
* TRUNC ( SYSDATE , ‘YEAR);

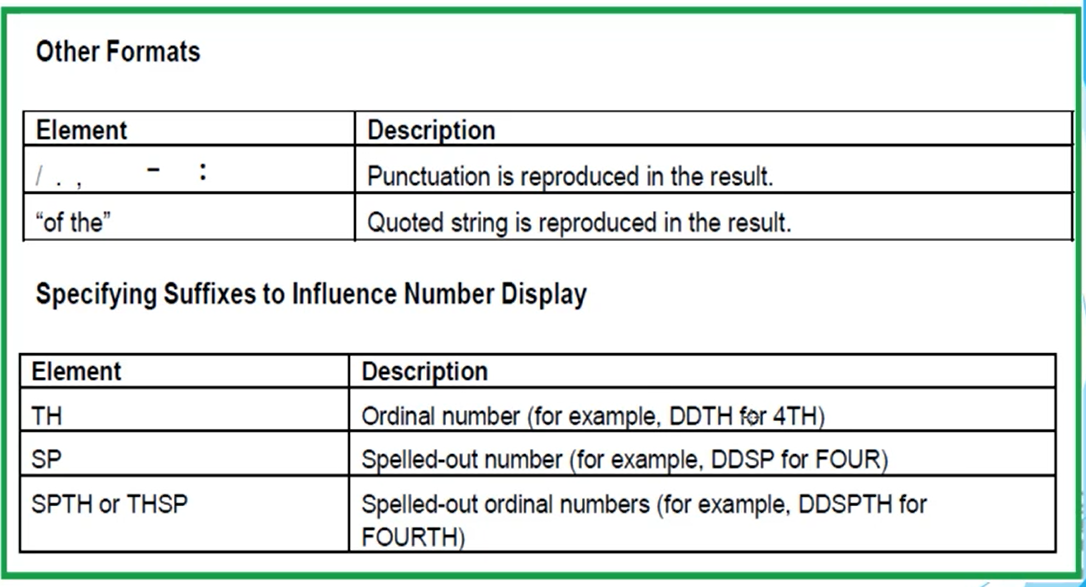












SELECT TO\_CHAR(SYSDATE,'dd-mm-yyyy hh:mi:ss AM')

FROM DUAL;--AM AND PM are the same

SELECT TO\_CHAR(HIRE\_DATE, 'fmDD Month YYYY')

--SO USING FM IS BETTER TO REMOVE SPAES

FROM EMPLOYEES;

--list all the employees who employeed in 2003

SELECT \* FROM EMPLOYEES

WHERE TO\_CHAR(HIRE\_DATE,'yyyy')='2003';

--list all the employees who employeed in feb

SELECT \* FROM EMPLOYEES

WHERE TO\_CHAR(HIRE\_DATE,'mm')='02'; --if you put only '2' this not coorect

SELECT \* FROM EMPLOYEES

WHERE TO\_CHAR(HIRE\_DATE,'fmmm')='2'; --you should use fm if you want to put only '2' cz fm remove 0 and space

SELECT \* FROM EMPLOYEES

WHERE HIRE\_DATE> TO\_DATE('10-11-2003','dd-mm-yyyy');

SELECT \* FROM EMPLOYEES

WHERE HIRE\_DATE> TO\_DATE('10-11- 2003','dd-mm-yyyy'); --oracle remove spaces

SELECT \* FROM EMPLOYEES

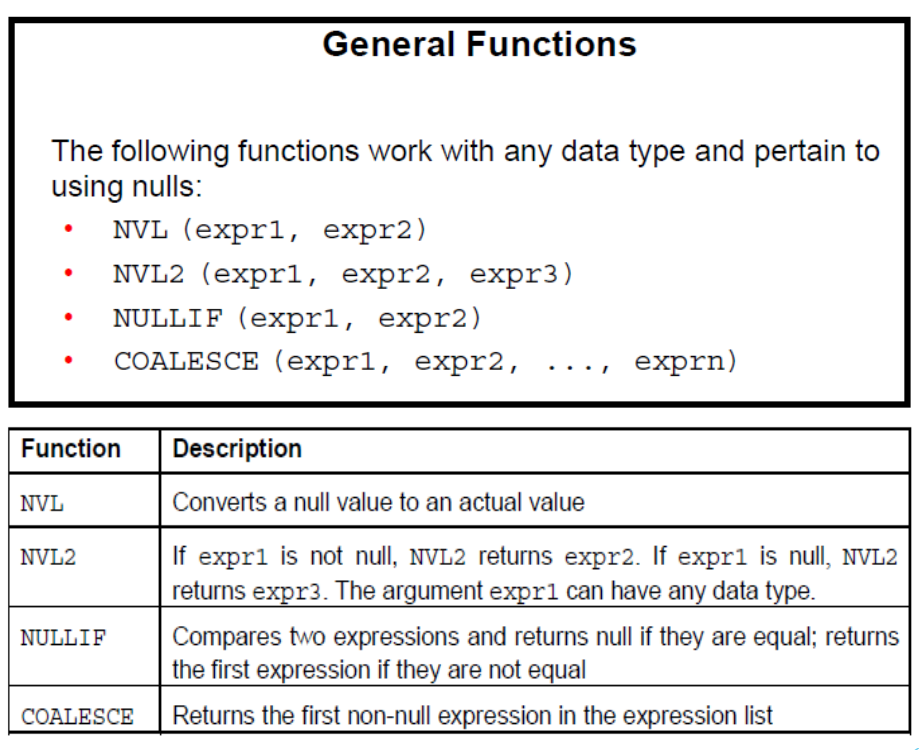
WHERE HIRE\_DATE> TO\_DATE('10-11- 2003','fxdd-mm- yyyy'); --when you put fx then exact should be mach

* Returinig 1985

SELECT TO\_CHAR( TO\_DATE('1-1-85','DD-MM-RR'),'YYYY') from dual;

* Returning 2085

SELECT TO\_CHAR(TO\_DATE('1-1-85','DD-MM-YY'),'YYYY') from dual;



--if exp1=exp2 then it return null, else it return exp1

SELECT FIRST\_NAME, LENGTH(FIRST\_NAME), LAST\_NAME, LENGTH(LAST\_NAME),

nullif(LENGTH(FIRST\_NAME), LENGTH(LAST\_NAME) ) results

FROM EMPLOYEES;

-- coalesce function 🡪 it return the first non-null value

SELECT EMPLOYEE\_ID,FIRST\_NAME, COMMISSION\_PCT, MANAGER\_ID, SALARY,

COALESCE(COMMISSION\_PCT,MANAGER\_ID,SALARY),

nvl( nvl(COMMISSION\_PCT,MANAGER\_ID), SALARY ) --nested nvl equal to COALESCE

FROM EMPLOYEES;

* Case 🡪 if you don’t add else , null will be shown if no case is met

SELECT first\_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;

--you can make the condition after when

SELECT FIRST\_NAME, JOB\_ID, SALARY,

CASE WHEN JOB\_ID='IT\_PROG' THEN 1.10\*SALARY

WHEN JOB\_ID='ST\_CLERK' THEN 1.15\*SALARY

WHEN job\_id='SA\_REP' THEN 1.20\*salary

ELSE SALARY

END "REVISED\_SALARY"

FROM EMPLOYEES;

* If you don’t add last odd value , null will be returned if no case is met

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;