<https://www.mycompiler.io/new/sql>

* Substr:

e.g.

1. select substr("Welcome to oracle sql", 4);

o/p: come to oracle sql

2. select substr("Welcome to oracle sql", 4,7);

O/P : come to

3. select substr("Welcome to oracle sql", 4,10);

O/p: come to or

4. select substr("Welcome to oracle sql");

O/P: Error: near line 1: wrong number of arguments to function substr()

5. select substr("Welcome to oracle sql",0);

O/P: Welcome to oracle sql

6. select substr("Welcome to oracle sql",1);

O/P: Welcome to oracle sql

7. select substr("Welcome to oracle sql.",-1);

O/P: .

8. select substr("Welcome to oracle sql.",-7);

O/P: le sql.

9. select substr("Welcome to oracle sql.",-7,2);

O/P: le

10. select substr("Welcome to oracle sql.",-7,5);

O/P: le sq

11. select substr("Welcome to oracle sql.",-7,-3);

O/P: rac

12. select substr("Welcome to oracle sql.",-7,0);

O/P:

13. select substr("Welcome to oracle sql.",10,-1);

O/P: t

14. select substr("Welcome to oracle sql.",10,-3);

O/P: e t

* Union, Union all, intersect and minus:

Union :

Graphical user interface, application

Description automatically generated

Union all:Graphical user interface, diagram

Description automatically generated

Intersect:Graphical user interface, diagram

Description automatically generated

Minus:

Graphical user interface, diagram, application, Word

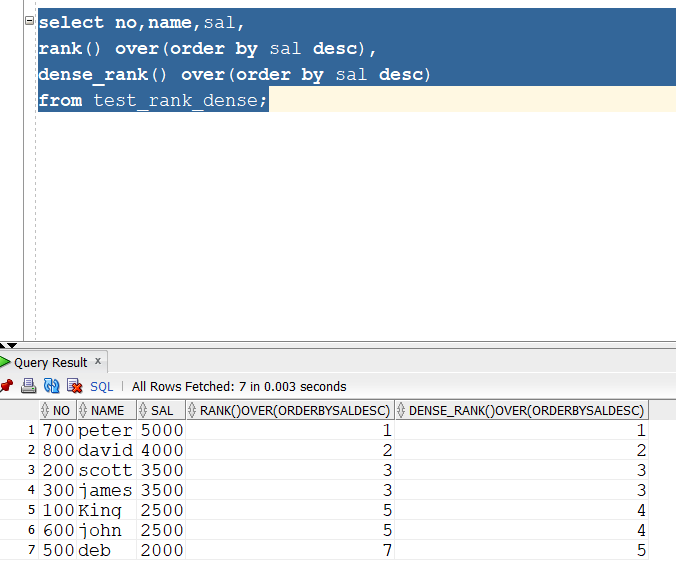
Description automatically generated

Rank and Dense rank: [practice more]

Rank and dense rank:

Diagram

Description automatically generated

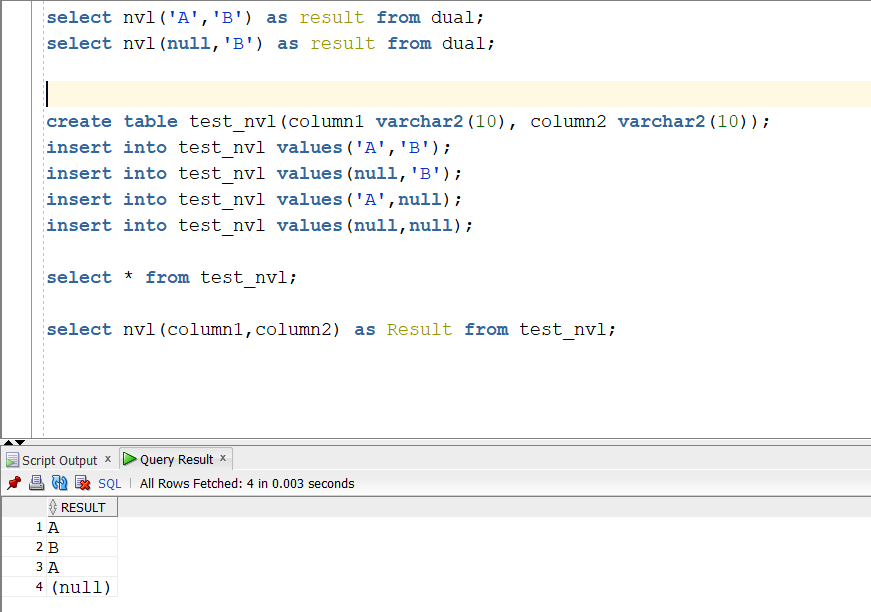


row\_number:

row\_num:

row\_id:

* NVL : Returns 1st not null value. [take only 2 parameters, if passed 3/less then Invalid number of arguments will be displayed]

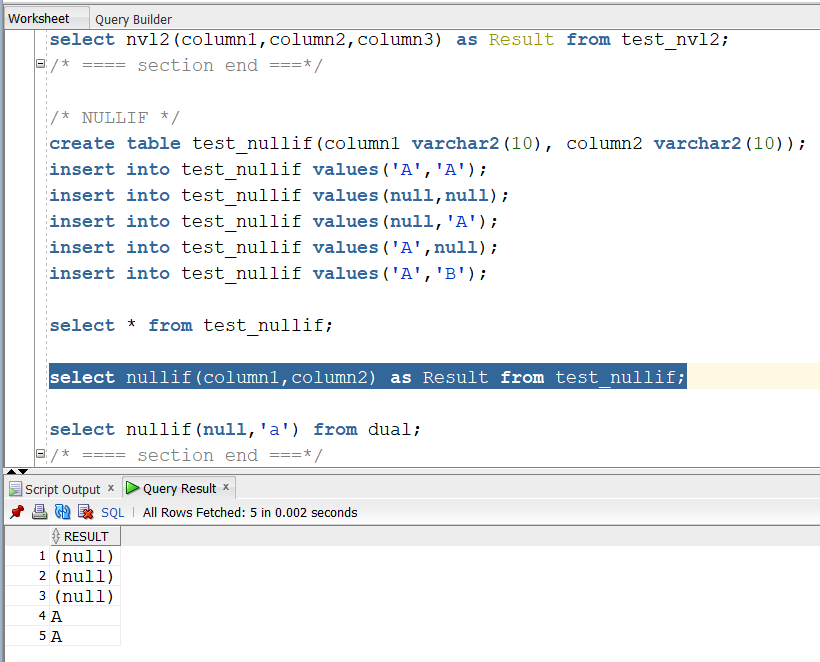


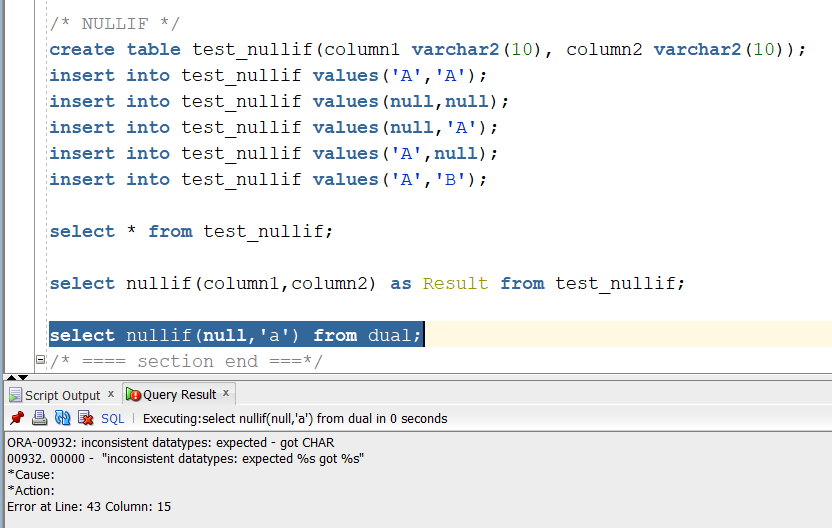
* NVL2: NVL2 will return=> if 1st value is not null will return 2nd value else 3rd value



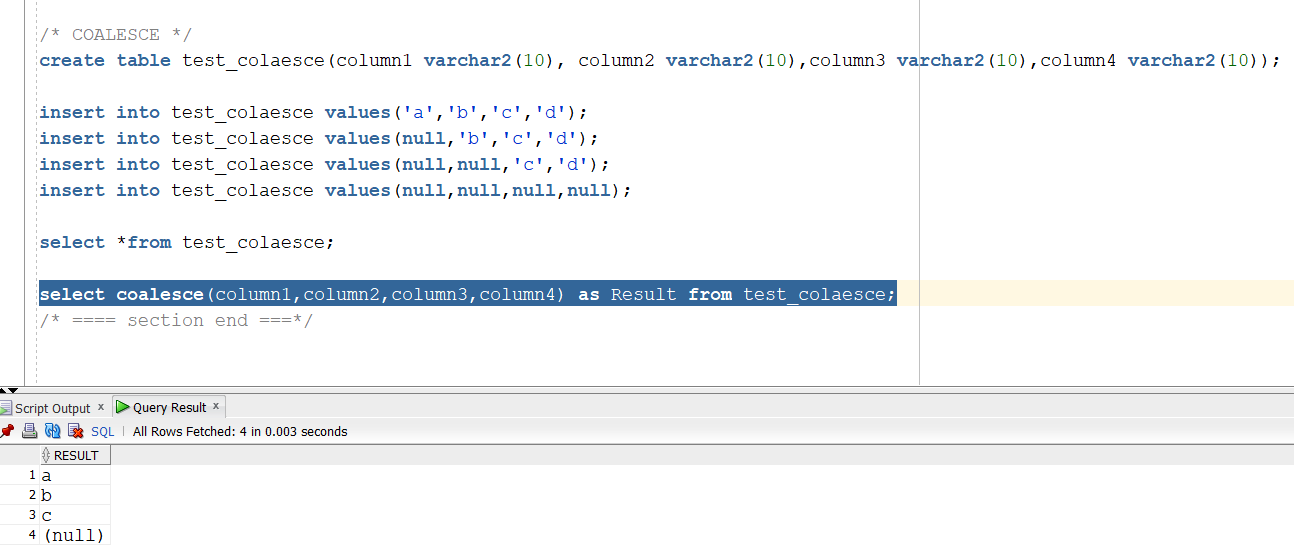
* NULLIF: The Oracle/PLSQL NULLIF function **compares expr1 and expr2**. If expr1 and expr2 are equal, the NULLIF function returns NULL. Otherwise, it returns expr1.

Note: if expr1 contains null then it will throw an error



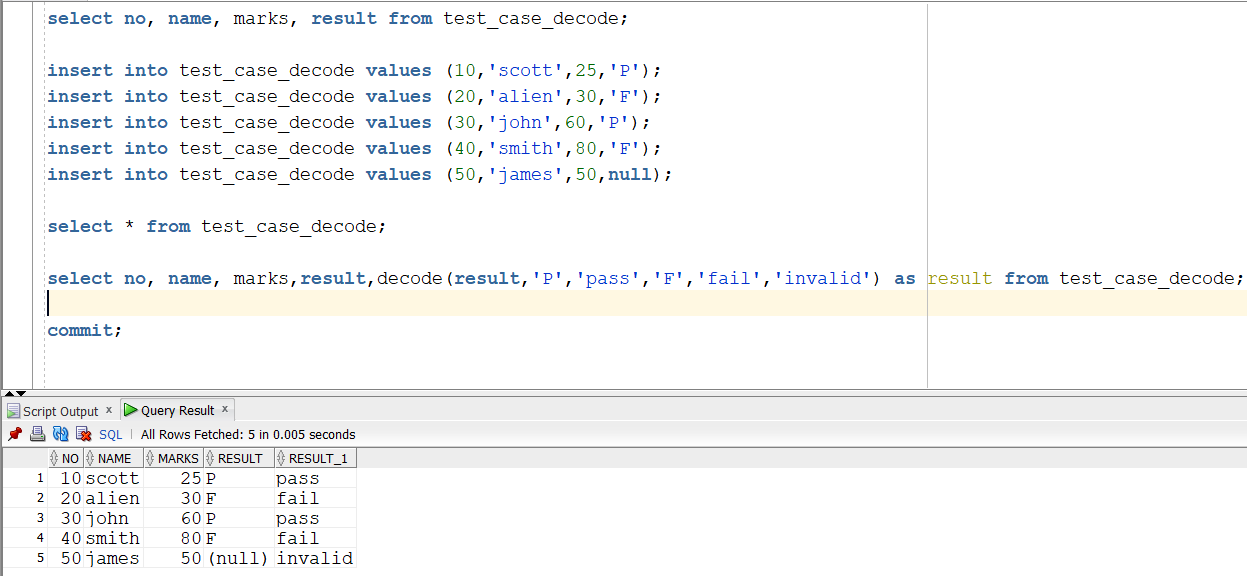


* COALESCE:   
  The Oracle/PLSQL COALESCE function **returns the first non-null expression in the list**. If all expressions evaluate to null, then the COALESCE function will return null.



* Decode and Case: Decode vs Case

**Decode:** DECODE compares *expr* to each *search* value one by one. If *expr* is equal to a *search*, then Oracle Database returns the corresponding *result*. If no match is found, then Oracle returns *default*. If *default* is omitted, then Oracle returns null.

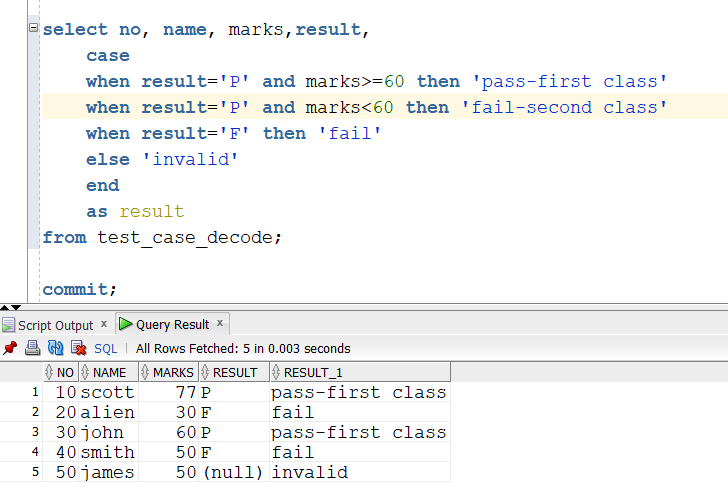


Result is invalid cz it is case-sensitive(p,f)



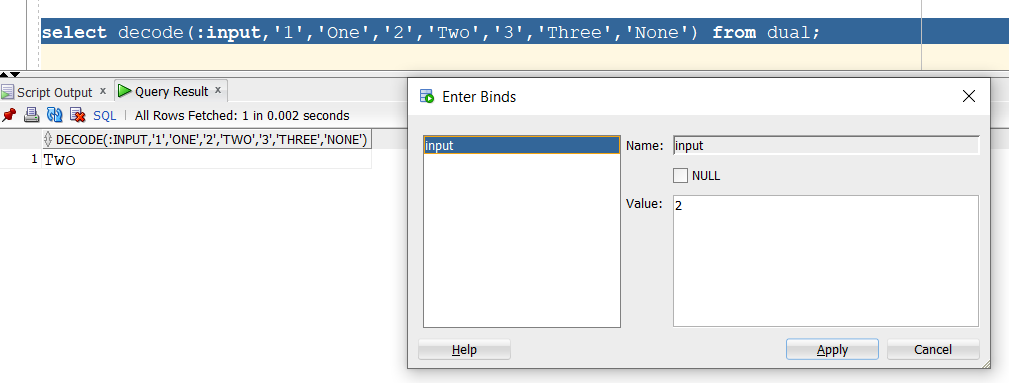
**Case:**

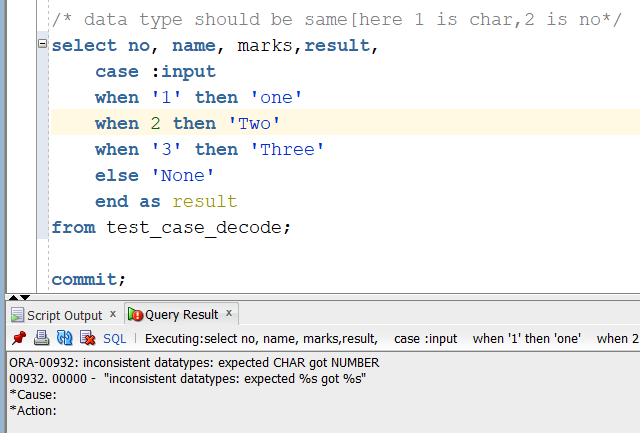
**Note: same as decode case matters(case-sensitive)**









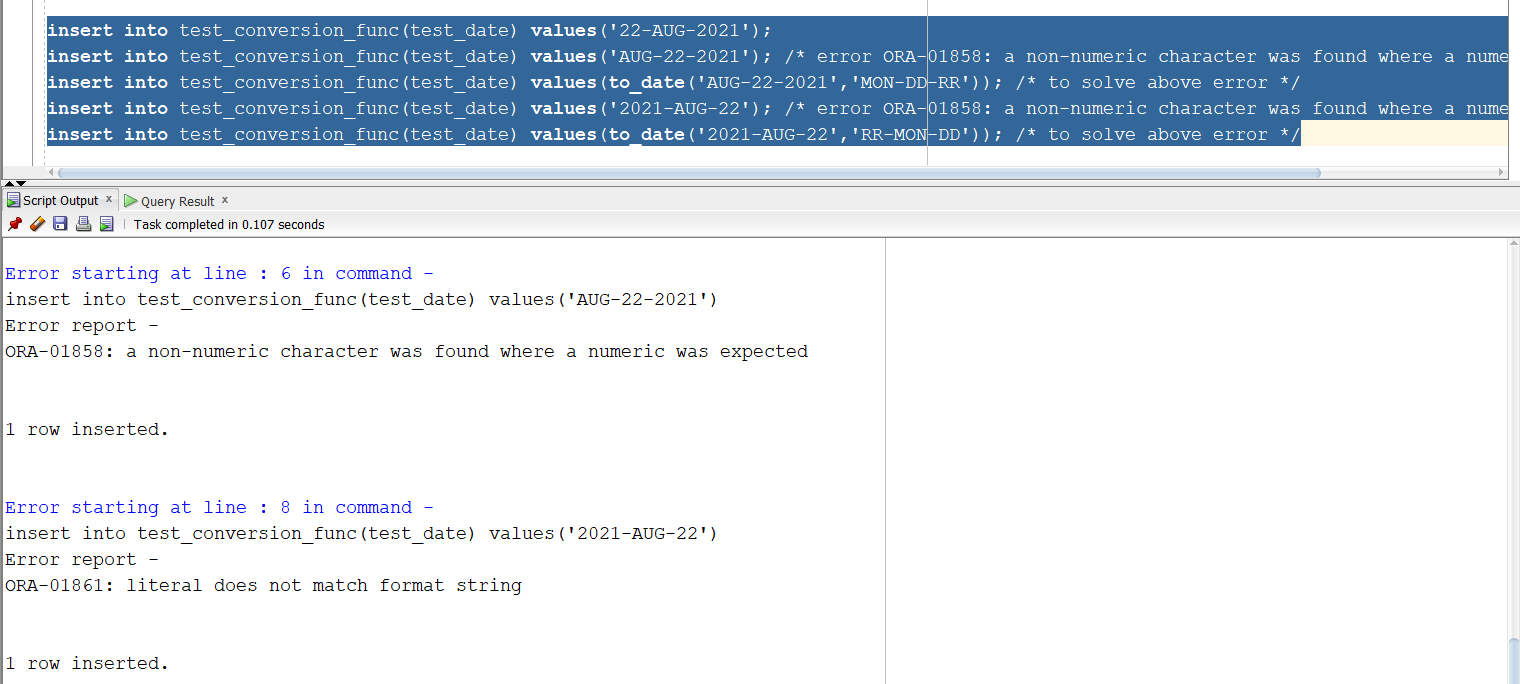


* To\_number, to\_char, to\_date

Oracle date format: DD-MON-RR

Note: while inserting date into table it will accept only DD-MON-RR format.[else ORA-01858: a non-numeric character was found where a numeric was expected error will be displayed] to solve this format issue we use to\_date()

Char to date:



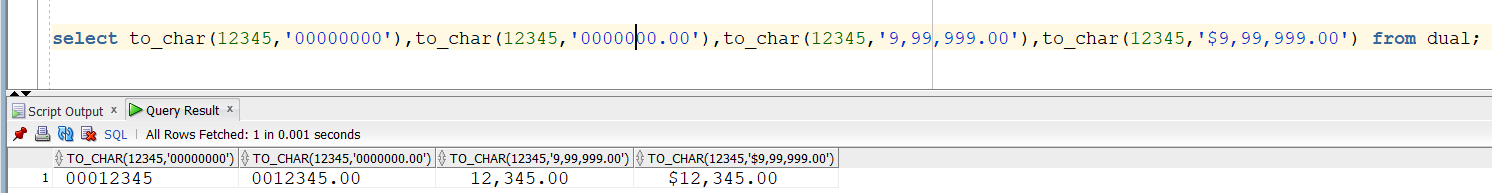
Char to number:

Oracle number format: 123456789.00

Note:



Number to char:



Number to date/date to number: not possible

Date to char – ok

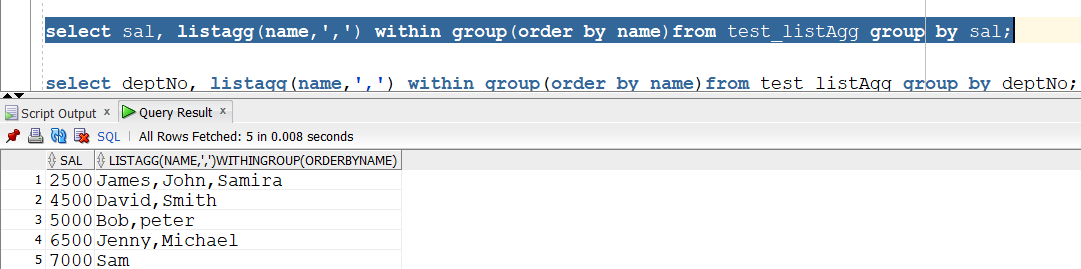
Date to numer- not possible

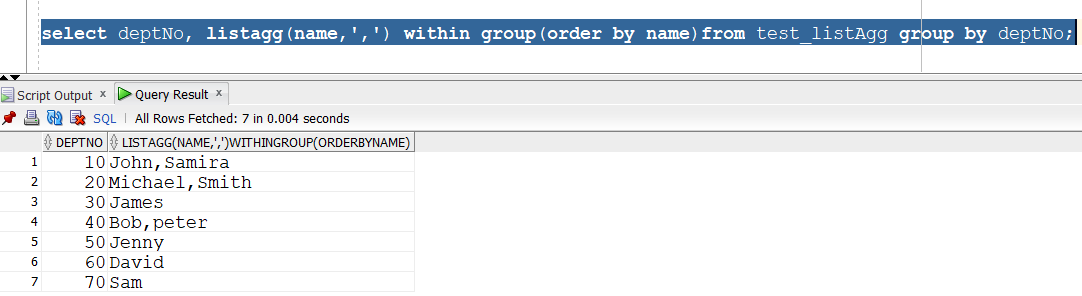
* Sysdate
* Last\_day
* Trunc
* Sysdate + sysdate
* listAgg:

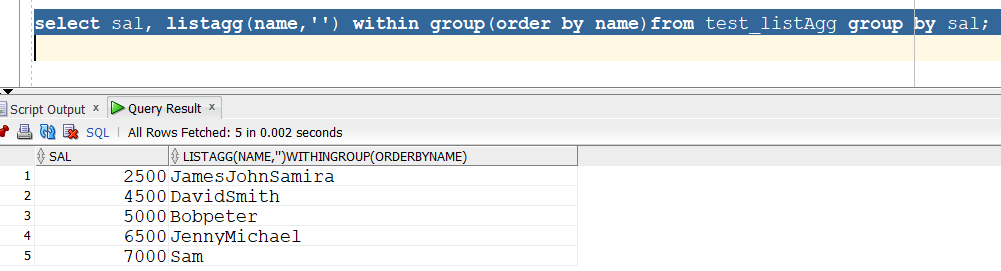
\*limitations of 11g, 12c, 19c

Syntax of 11g, 12c, 19c

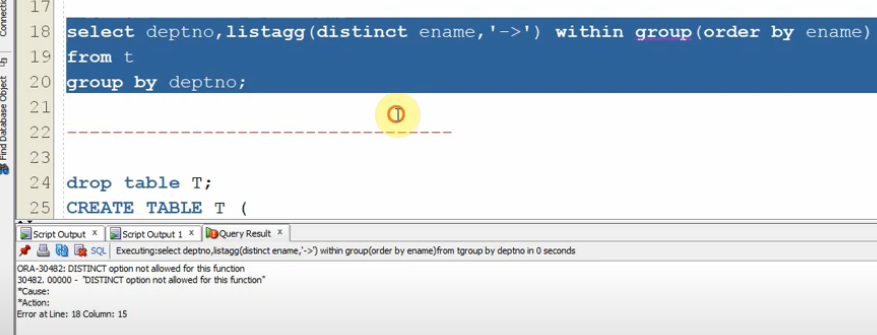
* Is aggregate function to convert row of strings to comma seperated list of values
* [introduced in 1st in oracle 11g then in sub-sequent version i.e. oracle 12c and oracle 19c there is lots of enhancements]
* If we don’t pass delimeter null will be considered as delimeter and string will be concatinated



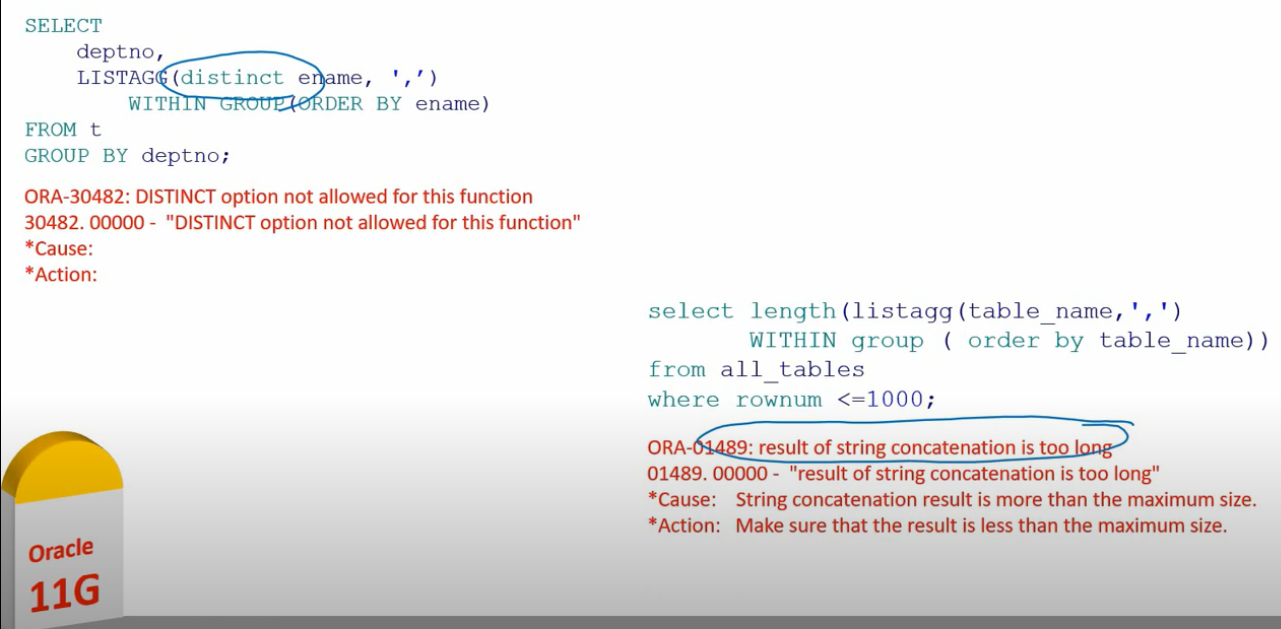




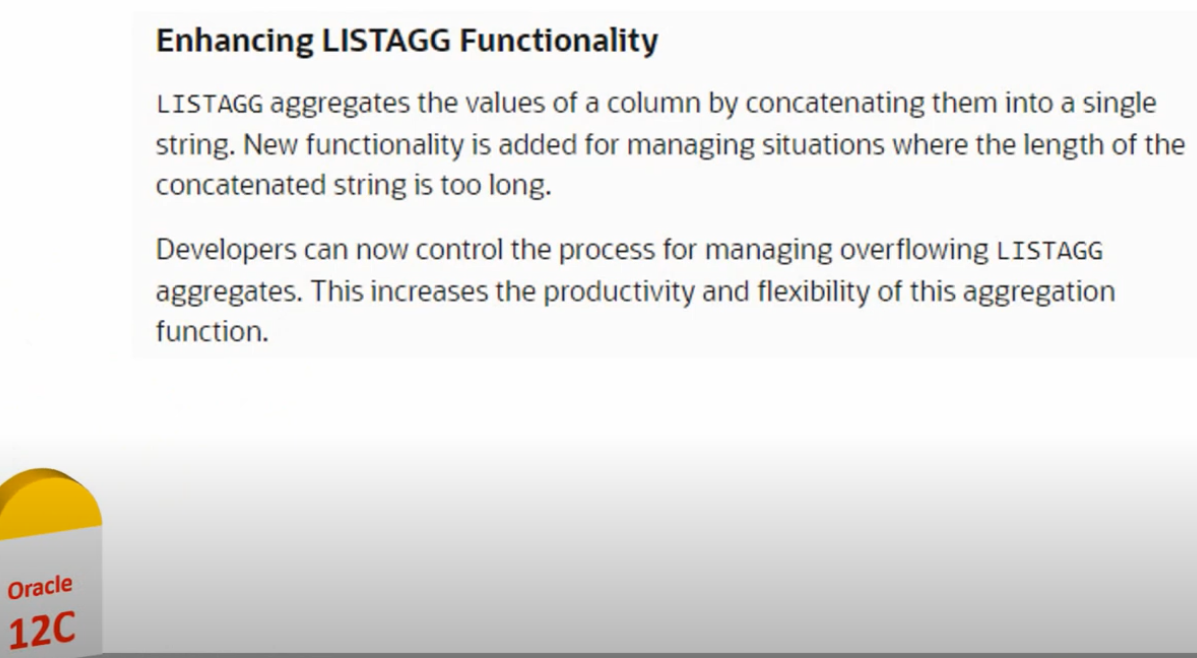
We can’t provide distinct keyword in 11 g



**Limitation of 11g (**distinct keyword and if we extend 4000 chars)



**Enhancing LISTAGG functionality:**



* lag & lead
* dual table