Institución Universitaria

Acreditada en Alta Calidad

ANDRES
MARTINEZ
GUTIERREZ

2018-2

DATABASE ADMINISTRATION ADVANCED

AGENDA

- Analytics Functions
- Views

Analytics Functions

https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions004.htm#SQLRF06174

http://bit.ly/2P2bMs3

ANALYTICS FUNCTIONS

- The group of rows is called a **Window**
  - For each row, a sliding window of rows is defined.
  - The window determines the range of rows used to perform the calculation.

ANALYTICS FUNCTIONS

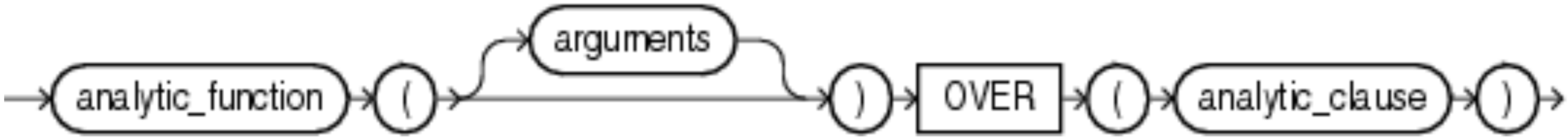
- Compute an aggregate value based on a group of rows
- They differ from aggregate functions in that they return multiple rows for each group
- They are executed just before of **ORDER BY** clause

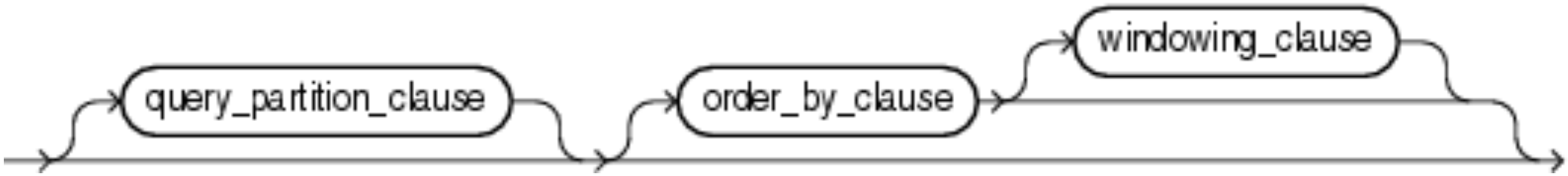
ANALYTICS FUNCTIONS

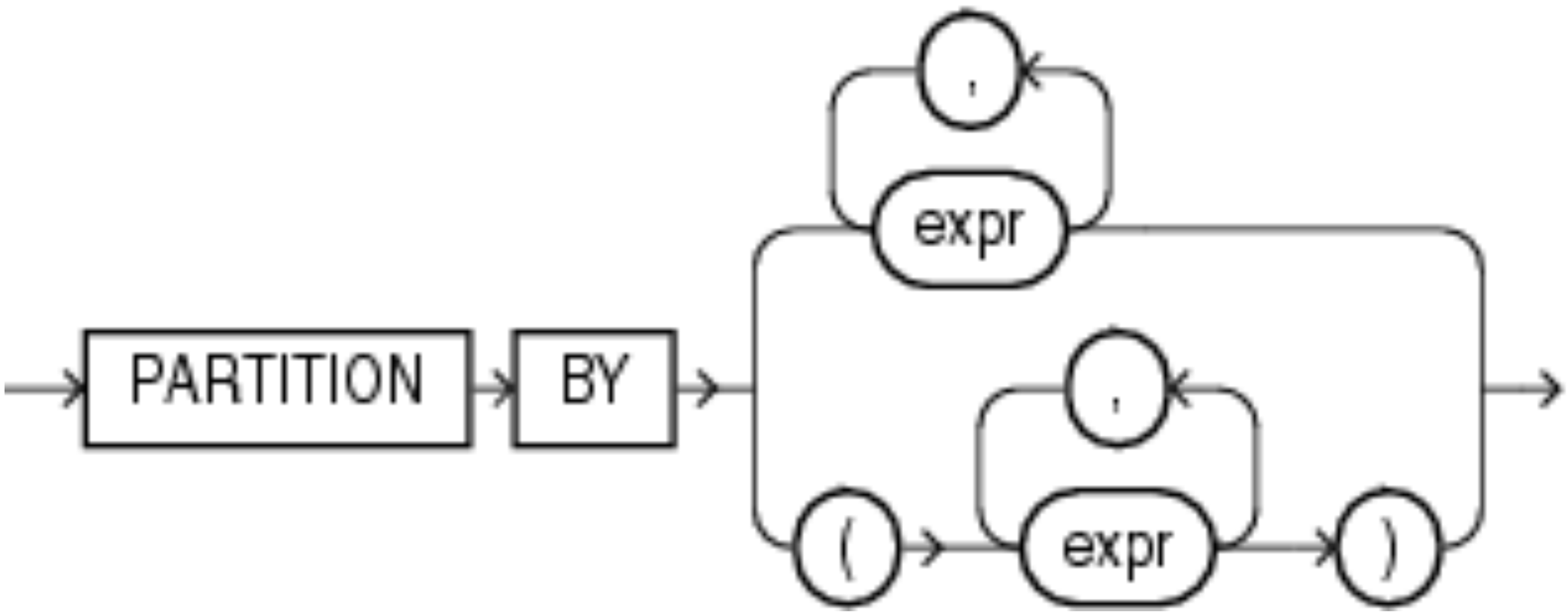
- Commonly used to compute cumulative, moving, centered and reporting aggregates.
- A window function performs a calculation across a set of table rows that are somehow related to the current row.

ANALYTICS FUNCTIONS

- Unlike regular aggregate functions, use of a window function does not cause rows to become grouped into a single output row
- Behind the scenes, the window function is able to access more than just the current row of the query result


analytic_function
(
arguments
)
OVER
(
analytic_clause
)


query_partition_clause
order_by_clause
windowing_clause


PARTITION
BY
,
expr
(
,
expr
)

AVG	CORR	COUNT	COVAR_POP
COVAR_SAMP	CUME_DIST	DENSE_RANK	FIRST
FIRST_VALUE	LAG	LAST	LAST_VALUE
LEAD	LISTAGG	MAX	MEDIAN
MIN	NTH_VALUE	NTILE	PERCENT_RANK

PERCENTILE_CONT	PERCENTILE_DISC	RANK
RATIO_TO_REPORT	REGR_ (Linear Regression) Functions	ROW_NUMBER
STDDEV	STDDEV_POP	STDDEV_SAMP
SUM	VAR_POP	VAR_SAMP
VARIANCE		

1. Count the number of employees by department order by department

```sql
select department, count(department)
from salaries
group by department order by department;

DEPARTMENT                                          COUNT(DEPARTMENT)
--------------------------------------------------- -----------------
Accounting                                                         87
Business Development                                               76
Engineering                                                        82
Human Resources                                                    85
Legal                                                              80
Marketing                                                          91
Product Management                                                 85
Research and Development                                           89
Sales                                                              73
Services                                                           88
Support                                                            76
Training                                                           88
```

2. List salaries and count the number of employees by department order by department

```sql
SELECT id, first_name, last_name,  department,
COUNT(*) OVER (PARTITION BY
department) department_count
FROM salaries
order by department;


        ID FIRST_NAME       DEPARTMENT                DEPARTMENT_COUNT
---------- ---------------- ---------------------     ----------------
       780 Roz              Accounting                87
       786 Bowie            Accounting                87
       788 Shannan          Accounting                87
       790 Marlon           Accounting                87
       799 Pinchas          Accounting                87
       807 Gib              Accounting                87
       816 Trude            Accounting                87
       819 Mick             Accounting                87
       821 Sharline         Accounting                87
       844 Charity          Accounting                87
       849 Garwin           Accounting                87
       822 Haroun           Business Development      76
       848 Minna            Business Development      76
       882 Darin            Business Development      76
       894 Dilly            Business Development      76
       904 Bessie           Business Development      76
       913 Daile            Business Development      76
       914 Sascha           Business Development      76
       923 Devondra         Business Development      76
       944 Hendrik          Business Development      76
       977 Manny            Business Development      76
       994 Baillie          Business Development      76
```

2. List salaries and the average salary by department order by department if the salary is under 150USD (round average to 3 decimals)

```sql
SELECT id, first_name, salary, department,
ROUND(AVG(salary) OVER (PARTITION BY department), 3) as AVG_by_department
FROM salaries
order by department;

ID    FIRST_NAME                                              SALARY DEPARTMENT      AVG_BY_DEPARTMENT
--    ------------------------------------------------- ---------- ---------------- -----------------
780   Roz                                                   591,66     Accounting      544,799
786   Bowie                                                 132,52     Accounting      544,799
788   Shannan                                               113,54     Accounting      544,799
790   Marlon                                                520,73     Accounting      544,799
799   Pinchas                                               765,04     Accounting      544,799
807   Gib                                                   483,03     Accounting      544,799
816   Trude                                                 774,18     Accounting      544,799
819   Mick                                                  285,17     Accounting      544,799
821   Sharline                                              505,23     Accounting      544,799
844   Charity                                               656,3      Accounting      544,799
849   Garwin                                                161,36     Accounting      544,799
```

```sql
SELECT id, first_name, salary, department,
ROUND(AVG(salary) OVER (PARTITION BY department), 3) as AVG_by_department
FROM salaries
order by department;

ID    FIRST_NAME                                              SALARY DEPARTMENT      AVG_BY_DEPARTMENT
--    ------------------------------------------------- ---------- ---------------- -----------------
780   Roz                                                   591,66     Accounting      544,799
786   Bowie                                                 132,52     Accounting      544,799
788   Shannan                                               113,54     Accounting      544,799
790   Marlon                                                520,73     Accounting      544,799
799   Pinchas                                               765,04     Accounting      544,799
807   Gib                                                   483,03     Accounting      544,799
816   Trude                                                 774,18     Accounting      544,799
819   Mick                                                  285,17     Accounting      544,799
821   Sharline                                              505,23     Accounting      544,799
844   Charity                                               656,3      Accounting      544,799
849   Garwin                                                161,36     Accounting      544,799
```

One more example

```sql
SELECT CARTS.ID,
CARTS.CUSTOMER_NAME,
ITEMS.ID AS ITEM_ID,
ITEMS.PRICE,
SUM(PRICE) OVER (PARTITION BY CARTS.ID) TOTAL_CART,
AVG(PRICE) OVER (PARTITION BY CARTS.ID) AVERAGE_PRICE,
MIN(PRICE) OVER (PARTITION BY CARTS.ID) MIN_PRICE
FROM CARTS
INNER JOIN CART_DETAILS ON CARTS.ID = CART_DETAILS.CART_ID
INNER JOIN ITEMS ON ITEMS.ID = CART_DETAILS.ITEM_ID
WHERE CARTS.ID IN (2,7);
```

Views

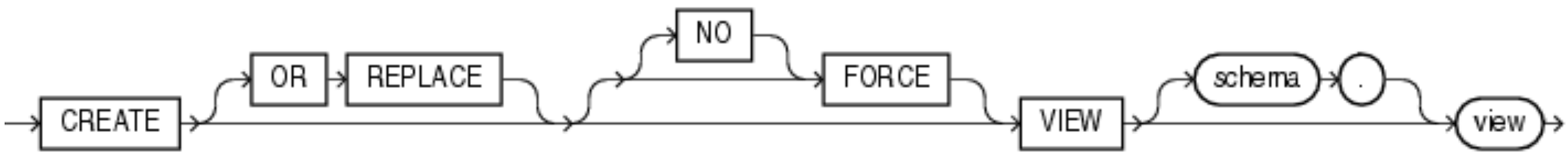
https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_8004.htm

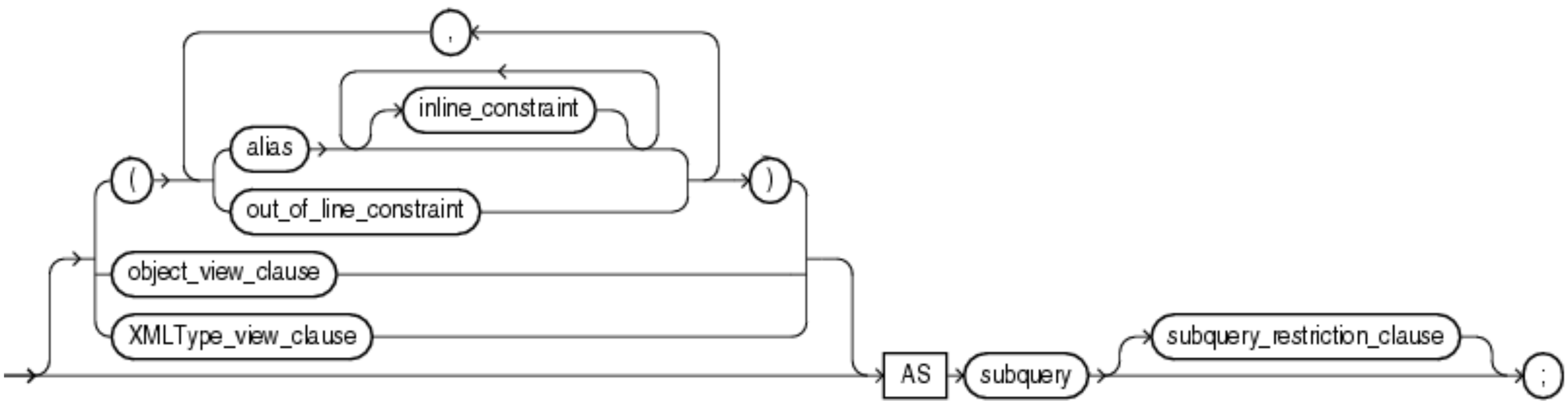
Views

- *"Logical table based on one or more tables or views. A view contains no data itself. The tables upon which a view is based are called* ***base tables****".*

Views (Pre-requisites)

- CREATE VIEW system privilege **(own schema)** or CREATE ANY VIEW **(any schema)**
- Privileges necessary to either select, insert, update, or delete rows from all the tables or views on which the view is based.


CREATE
OR
REPLACE
NO
FORCE
VIEW
schema
.
view


(
,
alias
inline_constraint
out_of_line_constraint
)
object_view_clause
XMLType_view_clause
AS
subquery
subquery_restriction_clause
;

Views (To consider)

- Specify **OR REPLACE** to re-create the view if it already exists
- Specify **NOFORCE** if you want to create the view only if the base tables exist and the owner of the schema containing the view has privileges on them. This is the default.

Views (Restrictions)

- **AS subquery** identifies columns and rows of the table(s) that the view is based on. The select list of the subquery can contain up to **1000** expressions.

- The subquery cannot select the **CURRVAL** or **NEXTVAL** pseudocolumns.

Views (Restrictions)

- If the subquery selects the ROWID, ROWNUM, or LEVEL pseudocolumns, then those columns must have aliases in the view subquery

Views (Restrictions)

- If the subquery uses an **asterisk (*)** to select all columns of a table, and you later add new columns to the table, then ***the view will not contain those columns until you re-create the view*** by issuing a **CREATE OR REPLACE VIEW** statement

Updatable Views

- Each column in the view must map to a column of a single table

- If you want a join view to be updatable, the DML statement must affect only one table underlying the join.

Updatable Views

- **Must not contain:**
  - A DISTINCT operator
  - An aggregate or analytic function
  - A GROUP BY, ORDER BY, MODEL, CONNECT BY, or START WITH clause

Updatable Views

```sql
CREATE VIEW clerk AS
   SELECT employee_id, last_name, department_id, job_id
   FROM employees
   WHERE job_id = 'PU_CLERK'
      or job_id = 'SH_CLERK'
      or job_id = 'ST_CLERK';

UPDATE clerk SET job_id = 'PU_MAN' WHERE employee_id = 118;
```

Updatable Views (With Check option)

- You cannot subsequently insert a new row into clerk if the new employee is not a clerk.
- *You can update an employee's job_id from one type of clerk to another type of clerk.*
  - The update in the preceding statement would fail, because the view cannot access employees with non-clerk job_id.

Updatable Views (With Check option)

```sql
CREATE VIEW clerk AS
   SELECT employee_id, last_name, department_id, job_id
   FROM employees
   WHERE job_id = 'PU_CLERK'
      or job_id = 'SH_CLERK'
      or job_id = 'ST_CLERK'
   WITH CHECK OPTION;
```

Example

```sql
CREATE VIEW EMPLOYEE_VIEW AS
  SELECT LAST_NAME, SALARY*12 ANNUAL_SALARY
  FROM EMPLOYEES
  WHERE DEPARTMENT_ID = 20;

CREATE OR REPLACE VIEW ACCOUNTS_AVG_BALANCE_GT_50000 AS
  SELECT ACCOUNTS.TYPE, LOCATIONS.CITY, AVG(BALANCE) FROM ACCOUNTS
  INNER JOIN LOCATIONS ON ACCOUNTS.LOCATION_ID = LOCATIONS.ID
  HAVING AVG(BALANCE) >= 50000
  GROUP BY ACCOUNTS.TYPE, LOCATIONS.CITY
  ORDER BY CITY;
```

Let’s practice

http://bit.ly/2AevyLW

Thank you!