To implement various Online Analytical Processing (OLAP) operations for multidimensional analysis for a given data warehouse on Sales History Schema of Oracle 11g and comparison of results with the following set of operations.

a. Roll up and Partial Roll Up

b. Cube and Partial Cube

c. Grouping Sets

d. Grouping ID’s

Q1. Find the total sales by country\_id and channel\_desc for the US and GB through the

Internet and direct sales in September 2000 and October 2000 using ROLL-UP Extension. The

query should return the following:

 The aggregation rows that would be produced by GROUP BY ,

 The First-level subtotals aggregating across country\_iso\_code for each combination

of channel\_desc and calendar\_month.

 Second-level subtotals aggregating

across calendar\_month\_desc and country\_iso\_code for each channel\_desc value.

 A grand total row.

select sum(AMOUNT\_SOLD),country\_iso\_code,CHANNEL\_DESC,CALENDAR\_MONTH\_NUMBER

from sh.channels natural join sh.countries natural join sh.times natural join sh.sales

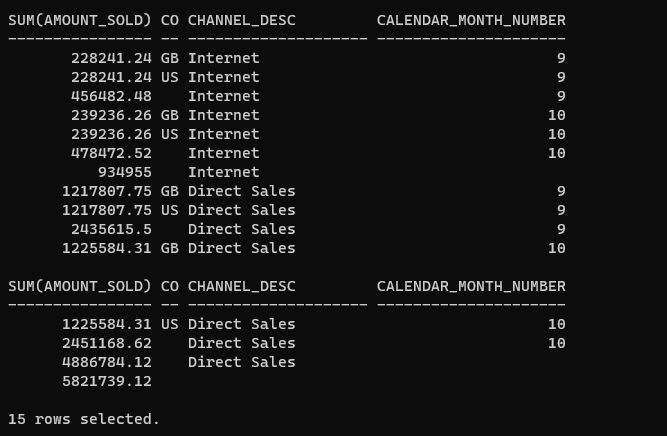
where country\_iso\_code in ('US','GB')

and channel\_desc in ('Direct Sales','Internet')

and CALENDAR\_MONTH\_NAME in ('September','October')

and CALENDAR\_YEAR='2000'

group by rollup (CHANNEL\_DESC,CALENDAR\_MONTH\_NUMBER,country\_iso\_code);



Q2. Find the total sales by country\_iso\_code and channel\_desc for the US and GB through the Internet and direct sales in September 2000 and October 2009 using CUBE aggregation across three dimensions- channel\_desc, calendar\_month\_desc, countries. country\_iso\_code.

select sum(AMOUNT\_SOLD),country\_iso\_code,CHANNEL\_DESC,calendar\_month\_desc

from sh.channels natural join sh.countries natural join sh.times natural join sh.sales

where country\_iso\_code in ('US','GB')

and channel\_desc in ('Direct Sales','Internet')

and CALENDAR\_MONTH\_DESC in ('2000-09','2000-10')

group by cube (CHANNEL\_DESC,calendar\_month\_desc,country\_iso\_code);



Q3. Find the total sales by country\_iso and channel\_desc for the US and France through the Internet and direct sales in September 2000

SELECT channels.channel\_desc, countries.country\_iso\_code,

TO\_CHAR(SUM(amount\_sold), '9,999,999,999') SALES$

FROM sh.sales, sh.customers, sh.times, sh.channels, sh.countries

WHERE sales.time\_id=times.time\_id AND sales.cust\_id=customers.cust\_id AND

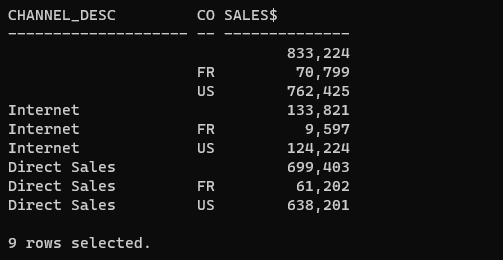
sales.channel\_id= channels.channel\_id AND channels.channel\_desc IN

('Direct Sales', 'Internet') AND times.calendar\_month\_desc='2000-09'

AND customers.country\_id=countries.country\_id

AND countries.country\_iso\_code IN ('US','FR')

GROUP BY CUBE(channels.channel\_desc, countries.country\_iso\_code);



Q4. Find the total sales by country\_iso and channel\_desc for the US and GB through the Internet and direct sales in September 2000 and October 2009 using PARTIAL ROLL-UP. The query should return the following:

 Regular aggregation rows that would be produced by GROUP BY without

using ROLLUP.

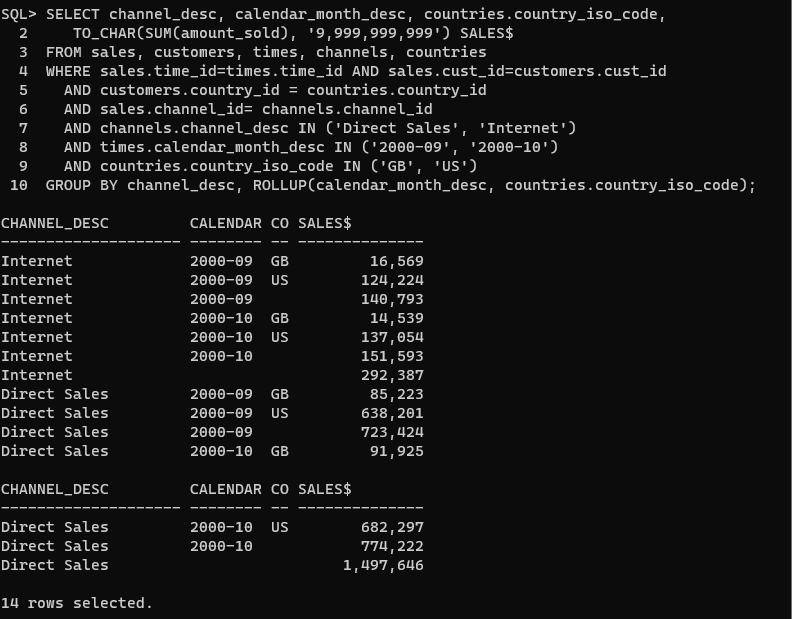
 First-level subtotals aggregating across country\_iso for each combination

of channel\_desc and calendar\_month\_desc.

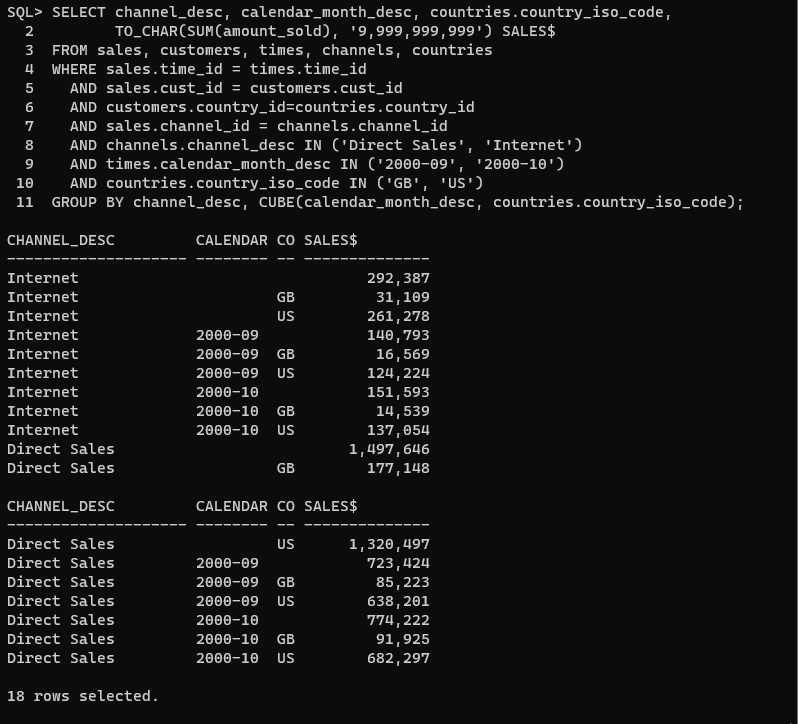
 Second-level subtotals aggregating across calendar\_month\_desc and country\_iso for

each channel\_desc value.

 It does not produce a grand total row.



Q5. Find the total sales by country\_id and channel\_desc for the US and GB through the Internet and direct sales in September 2000 and October 2009 using PARTIAL CUBE aggregation on month and country code and GROUP BY on channel\_desc.

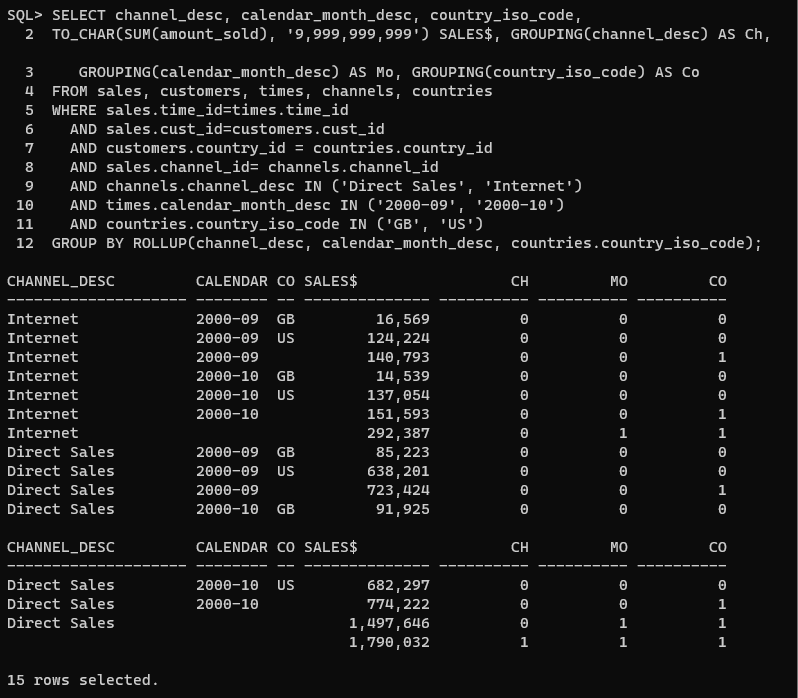


Q6. Use GROUPING to create a set of mask columns for the result set of Q1.

 Create grouping on channel\_desc and name it as CH

 Create grouping calendar\_month\_desc and name it as MO

 Create grouping on country\_iso\_code and name it as CO



Q7.  Find the total sales by country\_id and channel\_desc for the US and GB through the

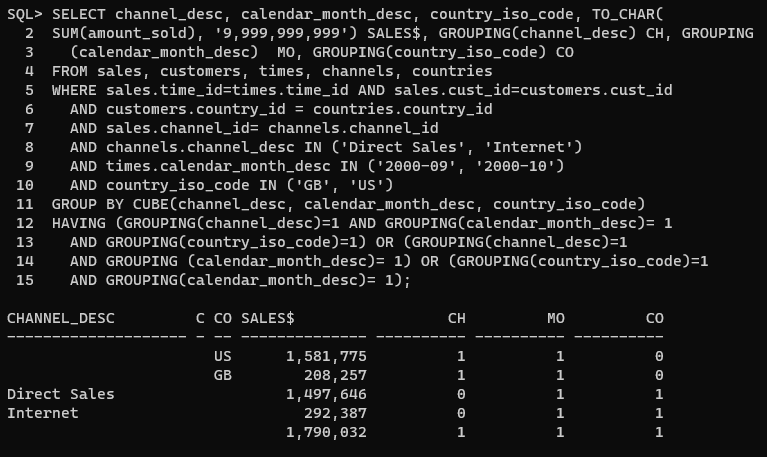
Internet and direct sales in September 2000 and October 2009 using GROUPING SETS.

Calculate aggregates over three groupings:

 (channel\_desc, calendar\_month\_desc, country\_iso\_code)

 (channel\_desc, country\_iso\_code)

 (calendar\_month\_desc, country\_iso\_code)



Q8. Consider the following Query and make conclusion from the result obtained.

Query: (scott Schema)

SELECT deptno, job, SUM(sal)

FROM emp

GROUP BY CUBE(deptno, job)

SQL> SELECT deptno, job, SUM(sal)

2 FROM emp

3 GROUP BY CUBE(deptno, job);

Q9. Calculate the salary for each department present in different cities of hr schema using rollup.

select sum(salary),city,department\_id,department\_name

from hr.employees natural join hr.departments natural join hr.locations

group by rollup(city,department\_id,department\_name);



Q10. Calculate the salary for each department present in different cities of hr schema using cube.

select sum(salary),city,department\_id,department\_name

from hr.employees natural join hr.departments natural join hr.locations

group by cube(city,department\_id,department\_name);

