A

(1-a)

time dimension, PK: time\_key

item dimension, PK: item\_key

branch dimension, PK: branch\_key

location dimension, PK: location\_key

(1-b)

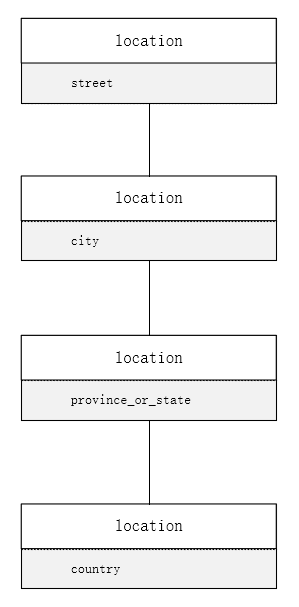
PK: Sales\_Fact(time\_key, item\_key, branch\_key, location\_key)

(1-c)

Measures: dollars\_sold, units\_sold

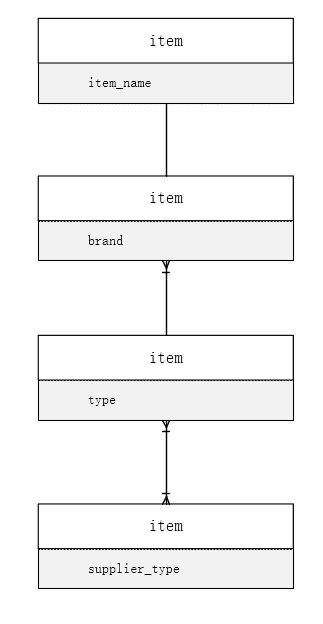
(1-d)

Location dimension: street → city → province\_or\_state → country



(1-e)

Item dimension: item\_name → brand → type → supplier\_type



(2-a)

Query: Find the total amount of dollars sold for all the branches for the years between 2014 to 2017.

Rollup operation summarizes data at a higher level by climbing up hierarchy or dimension reduction, in this case I implement dimension reduction by only have 2 dimensions in the query: branch dimension and year dimension..

(2-b)

Query: Find the total amount of dollars sold and the total number of units\_sold for all the branches for each city and each brand for the years between 2014 to 2017’s first 6 months.

Drilldown operation summarizes lower-level dimension hierarchy or introduces a new dimension. Since in the initial query, all four dimensions are included, no more new dimensions can be introduced. So the only way to drill-down is to summarize at a lower-level, which is the “first 6 month” because “month” is lower than “year” in time dimension.

(2-c)

Query: Find the total amount of dollars sold and the total number of units\_sold for all the branches for Washington DC and each brand for the years between 2014 to 2017.

Slice operation creates a new subcube by selecting a single value of one dimension. In this case, I select “Washington DC” as the single value in the location dimension.

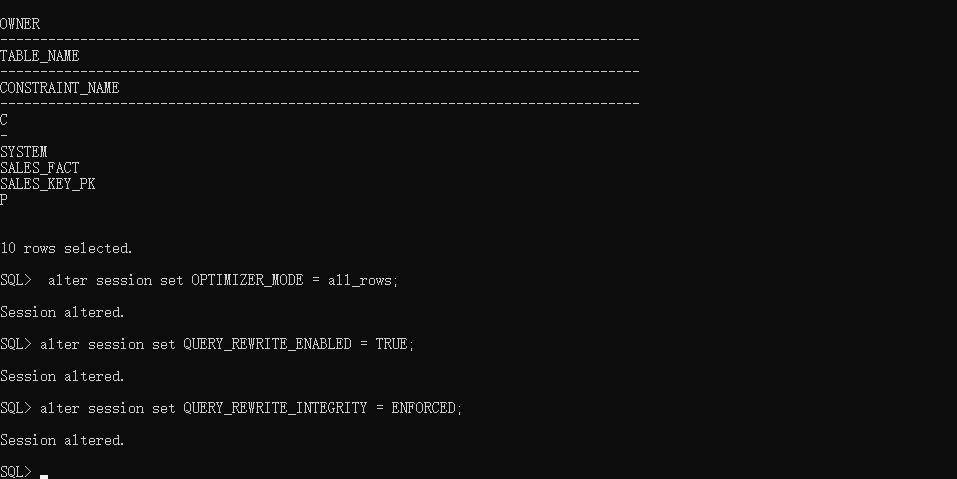
(2-d)

Query: Find the total amount of dollars sold and the total number of units\_sold for all the for all the branches for Philadelphia or Washington DC and Coca-Cola or PepsiCo for the years between 2014 to 2017 or 2017 to 2020.

Dice operation creates a new subcube by selecting a range of values of one or more dimensions. In this case, I select “Philadelphia or Washington DC” as multiple values in location dimension, “Coca-Cola or PepsiCo” as multiple values in item dimension and “2014 to 2017 or 2017 to 2020” as multiple values in time dimension.

B.

(1)



(2-a)

create sequence seq\_time start with 1 increment by 1;



(2-b)

create or replace procedure populate\_time\_dimension(p\_start\_date in DATE, p\_end\_date in DATE)

is

v\_actualDate date;

v\_dayofmonth number(2);

v\_month varchar2(9);

v\_quarter number(1);

v\_year number(4);

Begin

v\_actualdate := p\_start\_date;

LOOP

exit when v\_actualDate>p\_end\_date;

v\_dayofmonth:= to\_number(to\_char(v\_actualDate,'DD'));

v\_Month := to\_char(v\_actualDate,'Month');

v\_quarter := to\_number(to\_char(v\_actualDate,'Q'));

v\_year := to\_number(to\_char(v\_actualdate,'YYYY'));

insert into time\_dimension(day\_key, actual\_date, day\_of\_month, month, quarter, year)

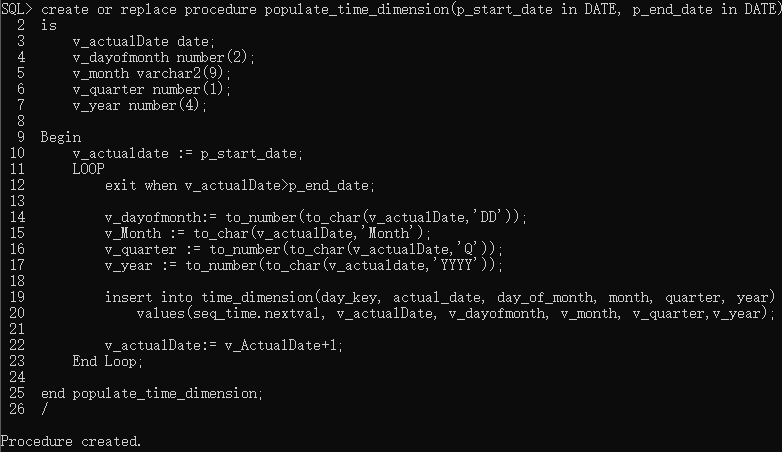
values(seq\_time.nextval, v\_actualDate, v\_dayofmonth, v\_month, v\_quarter,v\_year);

v\_actualDate:= v\_ActualDate+1;

End Loop;

end populate\_time\_dimension;

/



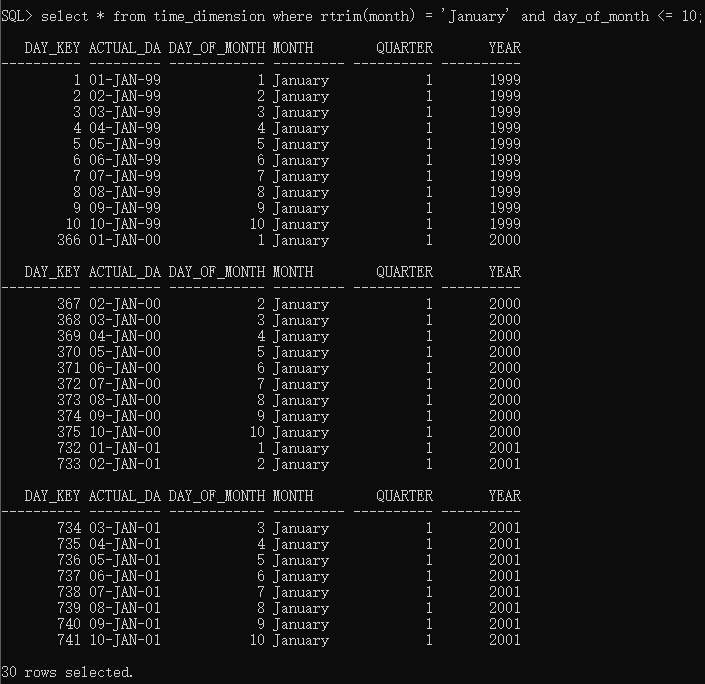
(2-c)

Exec populate\_time\_dimension ('01-JAN-1999','31-DEC-2001');



(2-d)

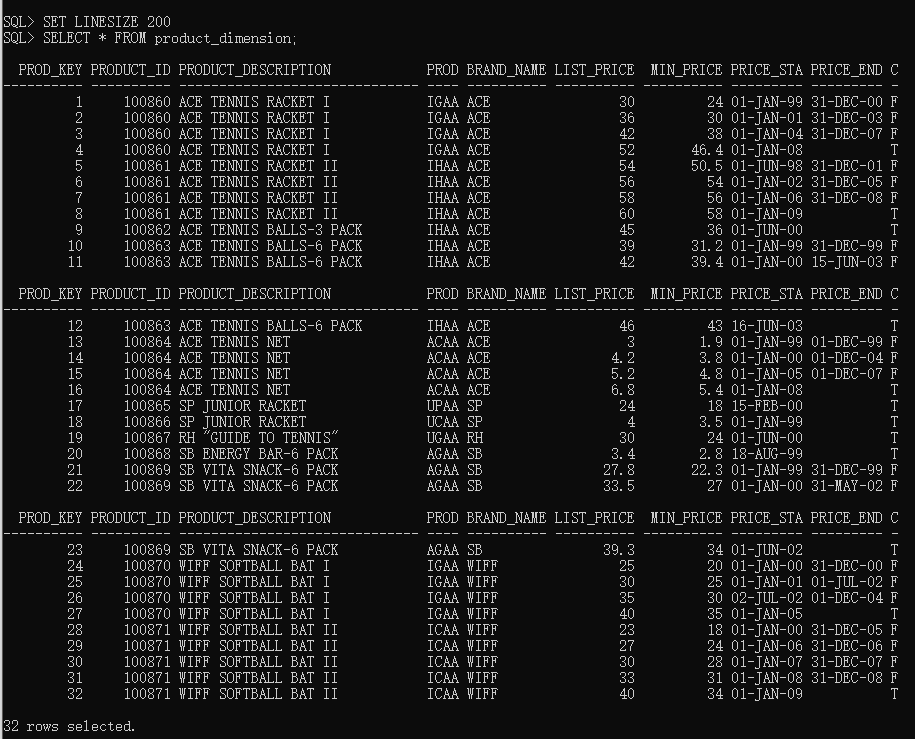
select \* from time\_dimension where rtrim(month) = 'January' and day\_of\_month <= 10;



(3)

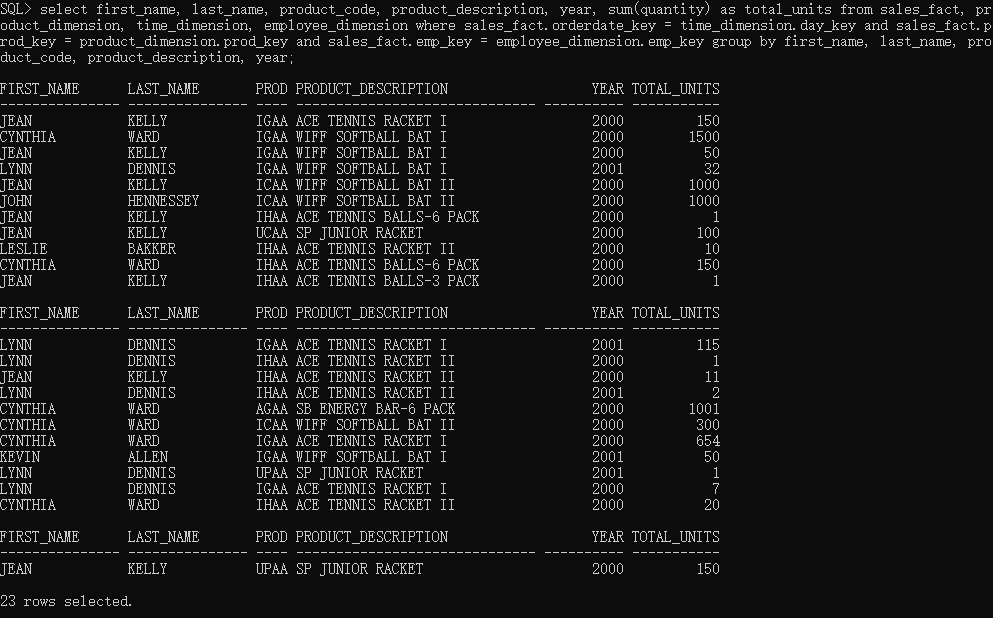
SET LINESIZE 200

SELECT \* FROM product\_dimension;



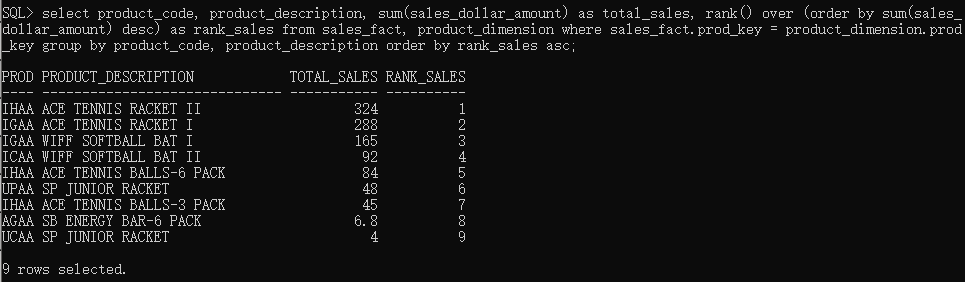
(4-a)

select first\_name, last\_name, product\_code, product\_description, year, sum(quantity) as total\_units from sales\_fact, product\_dimension, time\_dimension, employee\_dimension where sales\_fact.orderdate\_key = time\_dimension.day\_key and sales\_fact.prod\_key = product\_dimension.prod\_key and sales\_fact.emp\_key = employee\_dimension.emp\_key group by first\_name, last\_name, product\_code, product\_description, year;



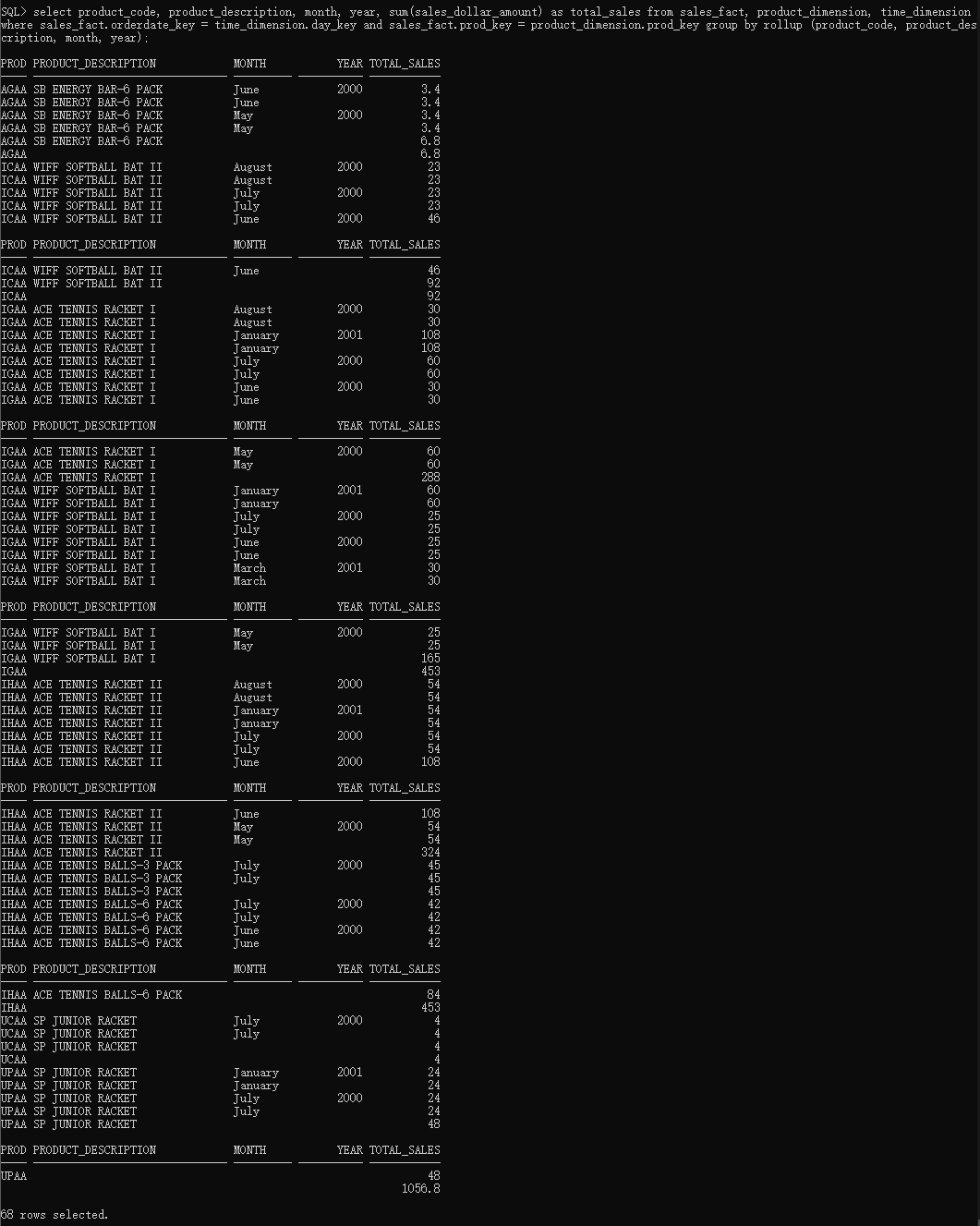
(4-b)

select product\_code, product\_description, sum(sales\_dollar\_amount) as total\_sales, rank() over (order by sum(sales\_dollar\_amount) desc) as rank\_sales from sales\_fact, product\_dimension where sales\_fact.prod\_key = product\_dimension.prod\_key group by product\_code, product\_description order by rank\_sales asc;



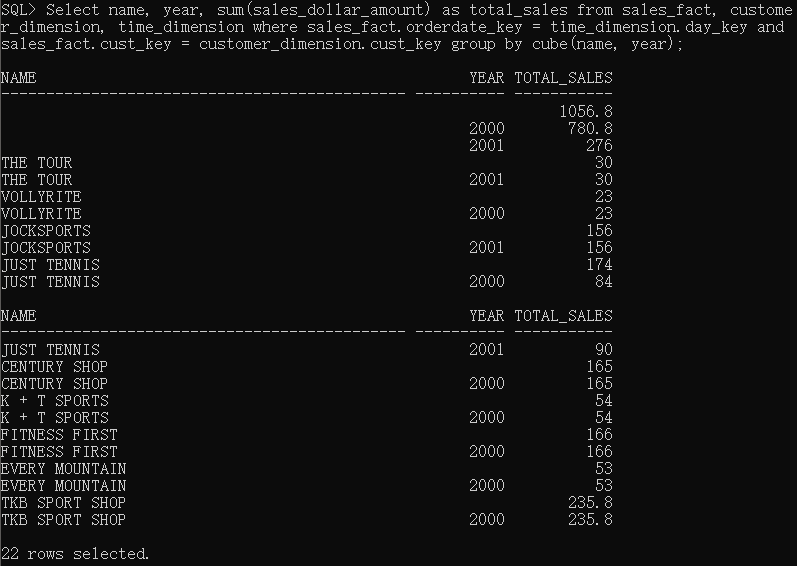
(4-c)

select product\_code, product\_description, month, year, sum(sales\_dollar\_amount) as total\_sales from sales\_fact, product\_dimension, time\_dimension where sales\_fact.orderdate\_key = time\_dimension.day\_key and sales\_fact.prod\_key = product\_dimension.prod\_key group by rollup (product\_code, product\_description, month, year);



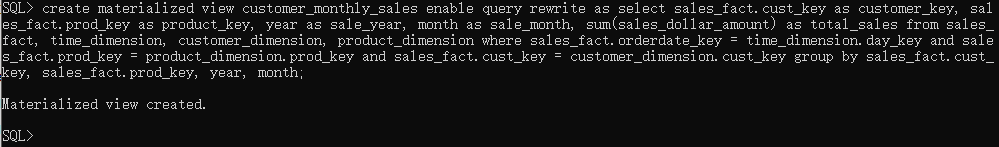
(4-d)

Select name, year, sum(sales\_dollar\_amount) as total\_sales from sales\_fact, customer\_dimension, time\_dimension where sales\_fact.orderdate\_key = time\_dimension.day\_key and sales\_fact.cust\_key = customer\_dimension.cust\_key group by cube(name, year);



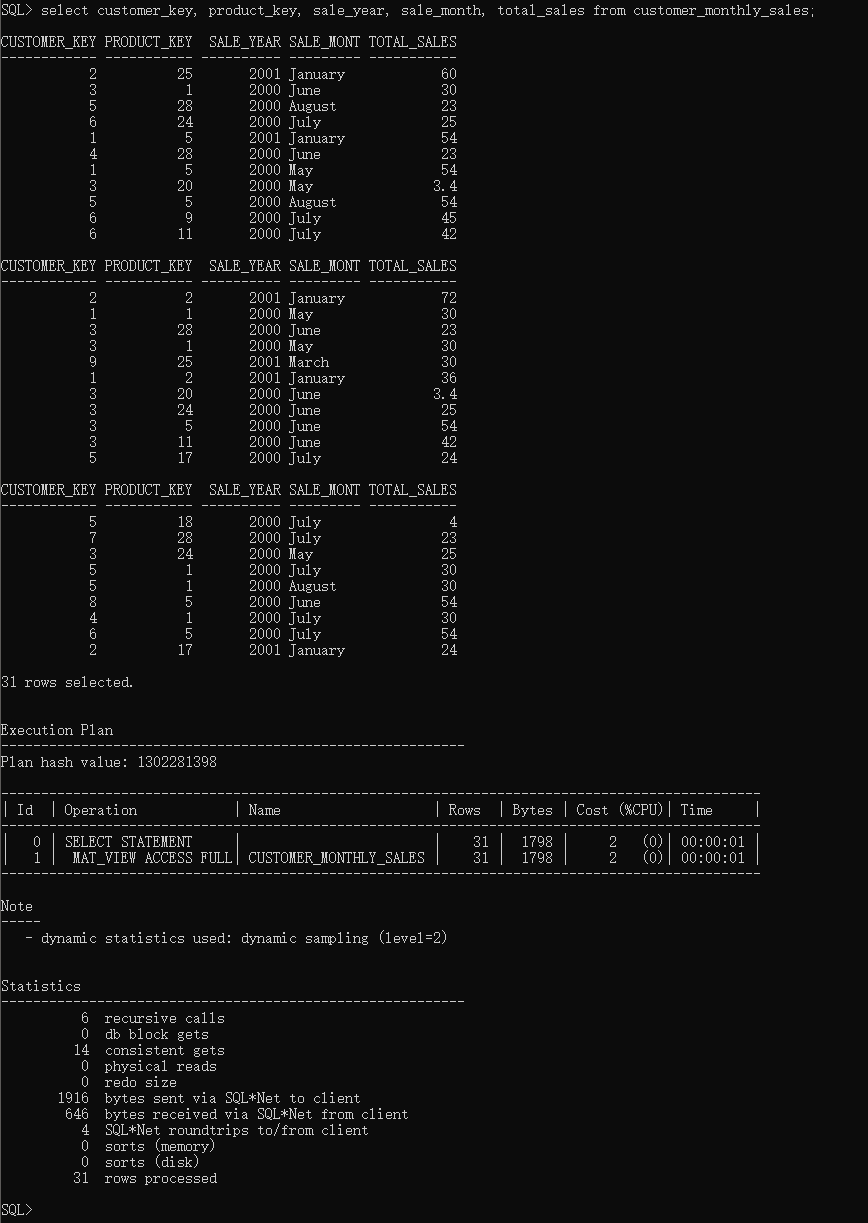
(5-a)

create materialized view customer\_monthly\_sales enable query rewrite as select sales\_fact.cust\_key as customer\_key, sales\_fact.prod\_key as product\_key, year as sale\_year, month as sale\_month, sum(sales\_dollar\_amount) as total\_sales from sales\_fact, time\_dimension, customer\_dimension, product\_dimension where sales\_fact.orderdate\_key = time\_dimension.day\_key and sales\_fact.prod\_key = product\_dimension.prod\_key and sales\_fact.cust\_key = customer\_dimension.cust\_key group by sales\_fact.cust\_key, sales\_fact.prod\_key, year, month;



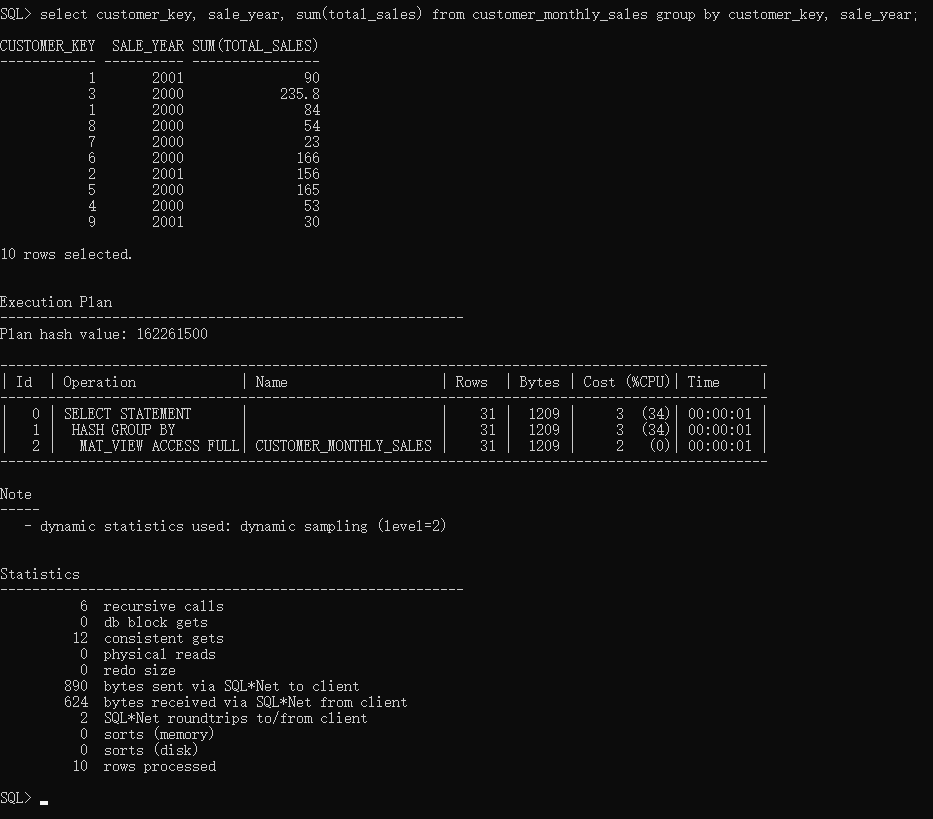
(5-b)

select customer\_key, product\_key, sale\_year, sale\_month, total\_sales from customer\_monthly\_sales;



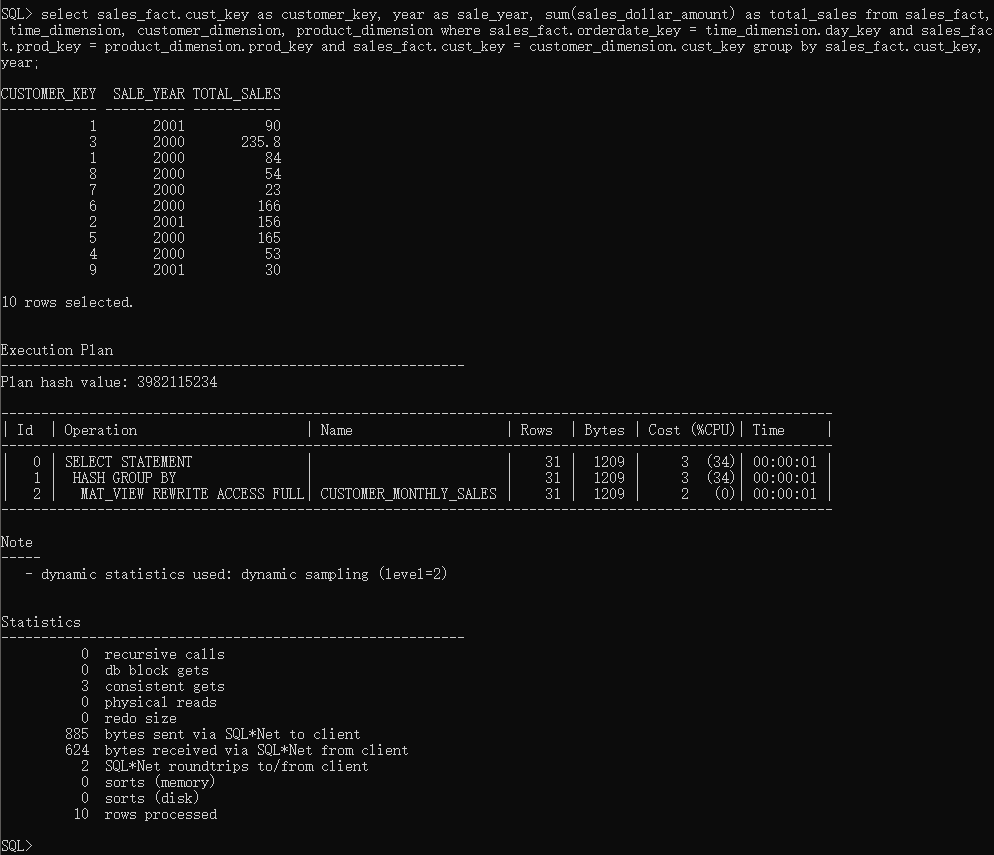
(5-c)

select customer\_key, sale\_year, sum(total\_sales) from customer\_monthly\_sales group by customer\_key, sale\_year;



(5-d)

select sales\_fact.cust\_key as customer\_key, year as sale\_year, sum(sales\_dollar\_amount) as total\_sales from sales\_fact, time\_dimension, customer\_dimension, product\_dimension where sales\_fact.orderdate\_key = time\_dimension.day\_key and sales\_fact.prod\_key = product\_dimension.prod\_key and sales\_fact.cust\_key = customer\_dimension.cust\_key group by sales\_fact.cust\_key, year;



Yes. From the execution plan, we can see the MV Rewrite feature and customer\_monthly\_sales is used in this query.