Reference Notes of Oracle 12c SQL Part 4

Oracle Trainer :- Sekhar

Set Operators

****

Union

*****

union is a set operator which is used for combining 2 queries.

The SQL UNION operator is used to combine the result sets of 2 or more SELECT statements. It removes duplicate rows between the various SELECT statements.

```
SQL> select snum from salespeople
  2  union
  3  select snum from customers;
```

```
 SNUM
------
  444
 1001
 1008
 1013
 1040

 SNUM
------
 1577
 1666
 1777
 1899
 1982
 3453
```

==========

Union all

****

This operator is used for combining both the queries and both queries will get executed.

rules while using any set operator is it should have a common attribute name and

data type and data type size

in the following query both the table output would come

```
SQL> select snum from salespeople
  2  union all
  3  select snum from customers;
```

Minus operator

****

Minus operator is used for removing the common values from both the tables.

Query

write a query where u will print all salespeople who have still not been able to bring a single customer

```
SQL> select snum from salespeople
  2  minus
  3  select snum from customers;
```

==========

Intersect operator

****

This operator which is a set operator will get printed if their is common records in both the tables.

Query

******

Write a query where u will print all salespeople who have booked at least 1 customer

```
SQL> select snum from salespeople
  2  intersect
  3  select snum from customers;
```

SNUM
1001
1040
1456

Joins

*****

Joins is a facility in oracle sql to combine 2 or more tables in to a single query as per the logical requirement of the project.

You can also use 4 to 7 tables also in the same query if required as per the business requirements of the project while writing joins queries.

The are many types of joins.

1) Equi Join : In equi join you need a common values in 2 or more tables.

Then those common values may be printed or not as per the logical requirement of the query.

query

*****

write a query where you will print the salesman no, name and who are his customer along with their cname and no

```
SQL> select salespeople.snum, sname, customers.cnum, cname
  2  from salespeople, customers
  3  where salespeople.snum = customers.snum;
```

SNUM SNAME CNUM CNAME

---------- ---------- ---------- ----------

1001 Kalia 2019 Haynes

1456 Ranjit singh 2007 Grass

1040 Rana Pratap 2044 Diana

==========

query

****

Write a query where you will print the snum, sname, and his cnum, and cname and also print the salesman no from sales table. (Hint use allias table names)

SQL> select s.snum, sname, cnum, cname, c.snum

2 from salespeople s, customers c

3 where s.snum = c.snum;

SNUM SNAME CNUM CNAME SNUM

---------- ---------- ---------- ---------- ----------

1456 Ranjit singh 2007 Grass 1456

1040 Rana Pratap 2044 Diana 1040

1013 Dr. Batli Wala 2891 Janaki R 1013

9001 James Singh 2828 Suganya Gowda 9001

==========

Query

******

Write a query where you will print snum, name and cnum and cname and print only

those salesperson where customer and salespersons reside in the same city.

```
SQL> select s.snum, sname, s.city, c.cnum, c.cname, c.city
   from salespeople s, customers c
   where s.snum = c.snum
   and
    rtrim(s.city)=rtrim(c.city);
```

==========

Inner Joins

****

Inner joins are also known as equi joins.

The Inner Join keyword selects all rows from both tables as long as there is a match

between the columns. If there are rows in the "Customers" table that do not have matches in "Orders", these customers will NOT be listed or displayed.

query

******

Write a query where using inner join print cnum from order table , cname from its master table, onum, amount and print only those customers who placed a orders.

```
sql>select o.cnum, c.cname, onum, oamount
  2  from customers c
  3    inner join orders o
  4     on c.cnum = o.cnum;
```

---

query

******

write a query where using inner join print snum, sname, and which customer they are giving service along with customer name and their customer number and print only those salespeople who are servicing any customers.

```
SQL> select s.snum, s.sname, c.cnum, c.cname
  2  from salespeople s
  3   inner join customers c
  4   on s.snum = c.snum;

      SNUM SNAME                     CNUM CNAME
---------- ------------------------- ---------- --------------------
      1456 Ranjit singh              2007 Grass
      1456 Ranjit singh              2001 Kalia
      1040 Rana Pratap               2044 Diana
      1013 Dr. Batli Wala            2891 Janaki R
      9001 James Singh               2828 Suganya Gowda
```

==========

LEFT JOIN

****

The LEFT JOIN keyword returns all rows from the left table (table1), with the matching rows in the right table

The result is NULL in the right side when there is no match.

query

******

write a query using left join print all cname, their cnum from orders table if they have placed orders and also the onum and oamount.

```
SQL> select c.cname, o.cnum, o.onum, o.oamount
  2  from customers c
  3  left join orders o
  4   on c.cnum = o.cnum
  5   order by c.cname desc;

CNAME                CNUM      ONUM    OAMOUNT
-------------------- ---------- ---------- ----------
Suganya Gowda
Lucy Singh           2014      3067    6543.34
Lucy Singh           2014      3029    9494.33
Kalia
Janaki R
Haynes
Grass                2007      3002     87366
Grass                2007      3004  234564.45
Grass                2007      3024   44425.44
Diana                2044      3007     425425
Diana                2044      3078   87345.33

11 rows selected.
```

The above query will also print customers who have not placed orders by giving null value their.

==========

RIGHT JOIN

****

The RIGHT JOIN keyword returns all rows from the right table (table2), with the matching rows in the left table (table1).

The result is NULL in the left side table when there is no match.

query

****

write a query where u will print all customers name and the orders number they have placed use right join.

```
SQL> select customers.cname, orders.onum
  2  from customers
  3    right join orders
  4     on customers.cnum = orders.cnum;

CNAME                ONUM
-------------------- ----------
Grass                3002
Grass                3004
Lucy Singh           3029
Lucy Singh           3067
Diana                3078
Diana                3007
```

==========

FULL OUTER JOIN

The FULL OUTER JOIN keyword returns all rows from the left table (table1) and from the right table (table2).

The FULL OUTER JOIN keyword combines the result of both LEFT and RIGHT joins.

Query

****

Write a query where you will print the cname, cnum from customer table and onum, cnum and order amount from order tables

using full outer joins

```
SQL> select c.cname, c.cnum, o.cnum, o.onum, oamount
  2  from customers c
  3   full outer join orders
  4    o on c.cnum = o.cnum;

CNAME                CNUM      CNUM      ONUM    OAMOUNT
-------------------- ---------- ---------- ---------- ----------
Grass                2007      2007      3004  234564.45
Diana                2044      2044      3007    425425
Grass                2007      2007      3002     87366
Lucy Singh           2014      2014      3067    6543.34
Lucy Singh           2014      2014      3029    9494.33
Grass                2007      2007      3024   44425.44
Diana                2044      2044      3078   87345.33
Janaki R             2891
Haynes               2019
Kalia                2001
Suganya Gowda        2828
```

Sub Query

****

A Subquery or Inner query or Nested query is a query within another SQL query, and embedded within the WHERE clause.

A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

Subqueries can be used with the SELECT, INSERT, UPDATE, and DELETE statements along with the operators like =, <, >, >=, <=, IN, BETWEEN etc.

There are a few rules that subqueries must follow:

---

a)Subqueries must be enclosed within parentheses.

b)A subquery can have only one column in the SELECT clause, unless multiple columns are in the main query for the subquery to compare its selected columns.

c)An ORDER BY cannot be used in a subquery, although the main query can use an ORDER BY.

d)Subqueries that return more than one row can only be used with multiple value operators, such as the IN operator.

e) The BETWEEN operator cannot be used with a subquery; however, the BETWEEN can be used within the subquery.

Query

******

Write a sub query where you will print all order details for customer name called grass.

```
SQL> select * from orders
  2  where cnum in
  3   (select cnum from customers
  4    where cname = 'Grass');

ONUM ODATE       OAMOUNT      CNUM      SNUM
---------- --------- ---------- ---------- ---------------------------------
    3004 09-JAN-15  234564.45      2007      1456
    3002 16-FEB-16     87366      2007      1456
    3024 04-JAN-16   44425.44      2007      1456
```

---

query

*****

Write a sub query where you will print all customers details of salesman name is Ranjit singh.

```
SQL> select * from customers
  2  where snum in
  3   (select snum from salespeople
  4    where rtrim(sname) = 'Ranjit singh');

    CNUM CNAME              CITY                    SNUM
---------- -------------------- -------------------- ----------
    2001 Kalia              Patna                 1456
    2007 Grass              New York              1456
```

Query

******

Write a sub query where you will print all orders details of customers who reside in Los Angeles.

```
SQL> select * from orders
  2    where cnum in
  3      (select cnum from customers
  4        where city = 'Los Angeles');
```

```
ONUM ODATE       OAMOUNT      CNUM      SNUM
---------- --------- ---------- ---------- ----------
     3007 16-FEB-16     425425      2044      1040
     3078 05-MAY-16   87345.33      2044      1040
```

Query

******

Write a sub query where you will print all the customer number, name and city who have not placed any orders.

```
SQL> select cnum, cname, city
  2    from customers
  3     where cnum not in
  4         (select cnum from orders);
```

CNUM	CNAME	CITY
2891	Janaki R	Mumbai
2019	Haynes	Cariro
2001	Kalia	Patna
2828	Suganya Gowda	Mumbai