Reference Notes of Oracle 12c SQL Part 2

Oracle Trainer :- Sekhar

Constraints

****

There are 6 types of constraints in oracle.

1st one is Primary key,

2nd one is Null.

Not null

Foreign key,

check clause

Default

---

query

******

waq where you will create a table called employees which will have the following constraints, Empno primary key, ename cannot be left blank and basic salary has to be minimum Rs. 2,400/- and city default 'Mumbai'

```
SQL> create table  employees
  2  (empno number(5) primary key,
  3   ename char(20) Not Null,
  4   doj    date,
  5   basic number(9,2) check(basic >=2400),
  6   city   varchar2(19) Default 'Bengaluru');

Table created.
```

While trying to add records in following scenarios

****

SQL> insert into employees

2 values(1004, null, sysdate, 8999, Default);

values(1004, null, sysdate, 8999, Default)

*****

ERROR at line 2:

ORA-01400: cannot insert NULL into

SQL> insert into employees

2 values(1004, 'DinDayal', sysdate, 1200, Default);

insert into ggemployee

*****

ERROR at line 1:

ORA-02290: check constraint (SYSTEM.SYS_C007585) violated

SQL> insert into employees

2 values(1004, 'DinDayal', sysdate, 1450, 'London');

1 row created.

SQL> commit;

==========

IN Clause

****

In Clause works faster for fetching records and when table has huge database.

It can be used with all char, varchar2, date and number attributes.

waq where you will print all details of snum 1001, 1008, 1004, 1090

SQL> select snum, sname, city

2 from salespeople

3 where snum in (1001, 1008, 1004, 1090);

SNUM	SNAME	CITY
1001	Bill gates	Navi Mumbai
1090	Dr. Jun Jun Wala	London
1008	James Bond	Mumbai

---

waq where u will print all salesperson residing in London or newyork or chicago or mumbai.

SQL> select snum, sname, city

2 from salespeople

3 where city in ('London', 'New York', 'Chichago', 'Mumbai');

SNUM	SNAME	CITY
1090	Dr. Jun Jun Wala	London
1008	James Bond	Mumbai
1400	Dr. Rahul	London
8977	Shri amit	London

Waq where you will display snum, sname and city of salesperson not residing in

London, newyork or chicago or mumbai.

SQL> select snum, sname, city

2 from salespeople

3 where city not in ('London', 'New York', 'Chichago', 'Mumbai');

SNUM	SNAME	CITY
1001	Kalicharan	Navi Mumbai

1234 seema bihar

1456 Ranjit singh Jaipur

1040 Rana Pratap Los Angeles

---

waq to print snum, sname, and city of salespeple whose sales number should not be 1008, 1001, 1004 and 1090

```
SQL> select snum, sname, city
  2  from salespeople
  3   where snum not  in (1001, 1008, 1004, 1090);
```

****

Important string Functions

****

Uppper and lower functions in the same query.

```
SQL> select upper(sname) name, lower(city) city, comm
  2  from salespeople;
```

---

lpad( ) Function

```
SQL> select lpad('Rama was a great king  ', 72, '*') lpad
  2  from dual;
```

RPAD()

****

```
SQL> select rpad('Rama was a great king  ', 72, '*') rpad
  2  from dual;
```

InitCap() : will print every words first letter in capital

SQL> select initcap(sname) from salespeople;

SQL> select initcap(sname)sname,city from salespeople;

Ltrim() : will remove the left trailing blank spaces from the string.

****

SQL> select ltrim(' Suresh is the V.C of Bangalore University

** 2 ') from dual;**

Rtrim( ) :-

****

SQL> select rtrim(' Suresh is the V.C of Bangalore University ') from dual;

Length() : -

SQL> select length(' India wins world cup of football in 2040 ')

** 2 from dual;**

SQL> select length(' Jaipur is a nice city ') from dual;

SQL> select length(trim(' Jaipur is a nice city ')) from dual;

SQL> select trim(' Jaipur is a nice city ') from dual;

TRIM('JAIPURISANICECI

Jaipur is a nice city

SQL> select substr(' White house is a nice fort ', 5, 8) substr

** 2 from dual;**

Alter Table.

The ALTER TABLE Statement

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

sql>ALTER TABLE salespeople

ADD DateofJoin Date;

The above query will add a column to the table employee;

To drop a column in a table

sql>ALTER TABLE salespeople

DROP COLUMN DateOfjoin

---

to modify a column in a table query is

sql>alter table salespeople

modify comm number(12,2);

Delete : is used to delete all records.

****

waq to delete all records in orders table.

sql> delete from orders;

If commit has not been given deleted records can be rolled back.

sql> rollback;

sql>select * from orders;

---

waq to delete all salespeople of london city;

```
sql>  delete from salespeople
        where city = 'London';


sql> commit;
```

==========

waq to remove details of salesman no 1004;

```
sql> delete from salespeople
       where snum =1004;
```

Update

Update is use to modify the records provided you have permission.

waq to update all records where commission is increased by 200 rupees for all employees

sql>update salespeople

set comm = comm +200;

sql> update salespeople

set comm = comm -100

where city = 'London';

sql> update customers

set city ='New York', Name = 'Rama'

where cnum = 2009;

Foreign Key

****

Foreign key is a key which is a primary key in another table.

SQL> desc salespeople;

Name	Null?	Type
SNUM	NOT NULL	NUMBER(5)
SNAME		CHAR(25)
CITY		VARCHAR2(20)
COMM		NUMBER(12,2)

Creating customer table with snum as foreign key connecting to parent table salespeople;

```
SQL> create table customers
  2  (cnum number(5) primary key,
  3   cname char(28),
  4   city varchar2(20),
  5    snum number(5) references salespeople(snum));
```

Table created.

Creating Orders table with snum and cnum as foreign key connecting to respected parent tables salespeople and customers.

```
SQL> create table orders
  2   (onum number(5) primary key,
  3    odate date,
  4    oamount number(11,2),
  5    snum number(5) references salespeople (snum),
  6    cnum number(5) references customers (cnum));
```

Table created.

SQL - LIKE Clause

****

The SQL LIKE clause is used to compare a value to similar values using wildcard operators.

There are two wildcards used in conjunction with the LIKE operator:

The percent sign (%) & The underscore (_)

The percent sign represents zero, one, or multiple characters.

The underscore represents a single number or character.

==========

sql> SELECT * FROM salespepople

WHERE empname like 'D%';

The above query will display whose names begins with D

% sign represent any characters but the first character must begins with character D.

---

waq where u will display salespeople whose city name begins with A

sql>SELECT * FROM salespeople

WHERE city like 'A%';

---

w.a.q wher you will print the sname, city, comm for all people residing in

London(Use like operator)

sql>Select snum, sname, city FROM salespeople

WHERE city like 'L%';

==========

_ (underscore) in like operator represent 1 character or number or space or special

symbol.

sql>

select * from salespeople

where sname like '_____';

In the above query we will display only those names which are of 5 characters.

one _ underscore represent one character.

---

sql> select * from salespeople

where city like '_____';

The above query will print city whose name size is of 6 characters.

---

sql>select * from customers

where cname like 'a%';

The above query will display all those cnames that begins with a

Between Operator

****

The BETWEEN operator is used to select values within a range.

sql> SELECT * FROM salespeople

WHERE comm between 10000 and 20000;

The following SQL statement selects all salespeople whose commission is between

10000 and 20000

To display the employees outside the range of the previous example, use NOT BETWEEN:

Example

sql> SELECT * FROM customers

WHERE comm NOT BETWEEN 10000 AND 20000;

The above query will print only those salespeople whose salary does not fall

in the above range.

ORDER BY Clause

****

The SQL ORDER BY clause is used to sort the data in ascending or descending order, based on one or more columns.

You can use more than one column in the ORDER BY clause.

Make sure whatever column you are using to sort, that column should be in column-list.

waq where you will sort on employee name sorting based on salary in ascending order.

SQL> SELECT * FROM employee

ORDER By SALARY;

Following is the query where we sort only by name in ascending order.

SQL> SELECT * FROM EMPLOYEE

order by empname;

---

sql> select empno, city, basic from employee

order by city;

in the above query the records are sorted based on city in ascending order.

Following is an example which would sort the result in descending order by city:

SQL> SELECT * FROM employee

ORDER BY city DESC;

or

sql> select empname, city from employee

order by city desc;

****