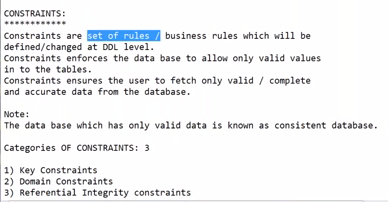
**Data integrity constraints :** set of rules to insert the data , which will be used at DDL level.

**Domain Constraints :** CHECK

SQL> --by using key constraints we can ristrict weather we can give data in that column or not

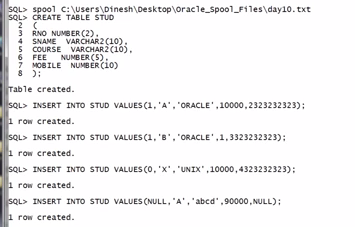
SQL> --and we can stop gtiving duplicate values

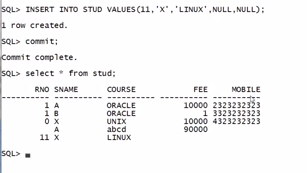
SQL> --but by using domain constraints, we can decide which type of data the table should accept by using CHECK constraint.



SQL> --first lets create a table stud without constraints.

SQL> -- without constraints we are getting invalid data as shown in fig





SQL> --rno. should not be null

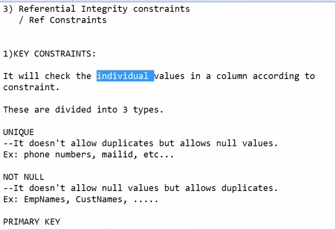
SQL> --fee can not be 1 rupee

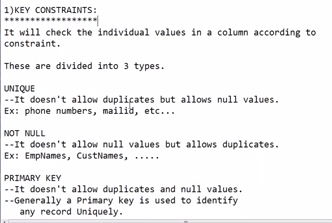
SQL> mobile numbers can not be starting with 2, 3 , 4

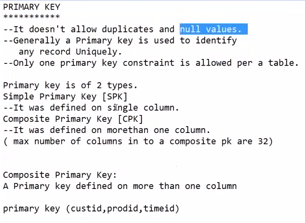
SP2-0734: unknown command beginning "mobile num..." - rest of line ignored.

SQL> --so we need to add some rules to insert data in table in the form of constraints.

SQL> --key constraint verify single value weather it is valid or not ?







SQL> --there are 3 types of key constraints.

SQL> --UNIQUE, NOT NULL , PRIMARY KEY.

SQL> -- WE CAN HAVE ONLY 1 PRIMARY KEY FOR A TABLE.

SQL> --Syntax for thekey constraint is shown in fig4

SQL> best example for primary key is serial number and roll number and employee id etv.

SP2-0734: unknown command beginning "best examp..." - rest of line ignored.

SQL> --creating student table by considering constraints.

SQL> create table student

2 (

3 sno number(5) primary key,

4 sname varchar2(10) not null,

5 course varchar2(10) not null,

6 fee number(10) not null,

7 mobno number(10) unique

8 );

Table created.

SQL> select \* from student;

no rows selected

SQL> desc student

Name Null? Type

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

SNO NOT NULL NUMBER(5)

SNAME NOT NULL VARCHAR2(10)

COURSE NOT NULL VARCHAR2(10)

FEE NOT NULL NUMBER(10)

MOBNO NUMBER(10)

SQL> insert into student values(1,;suresh', 'developer', '15000', 8790923747);

ERROR:

ORA-01756: quoted string not properly terminated

SQL> insert into student values(1,'suresh', 'developer', '15000', 8790923747);

1 row created.

SQL> insert into student values(1,'suresh', 'developer', '15000', 8790923748);

insert into student values(1,'suresh', 'developer', '15000', 8790923748)

\*

ERROR at line 1:

ORA-00001: unique constraint (SYSTEM.SYS\_C007019) violated

SQL> --not accepting duplicate sno.

SQL> insert into student values(2,'suresh', 'developer', '15000', 8790923748);

1 row created.

SQL> insert into student (sno, sname, course, fee) values(3,'sukumar', 'dba', '1500');

1 row created.

SQL> insert into student (sno, sname, course) values(3,'sukumar', 'dba');

insert into student (sno, sname, course) values(3,'sukumar', 'dba')

\*

ERROR at line 1:

ORA-01400: cannot insert NULL into ("SYSTEM"."STUDENT"."FEE")

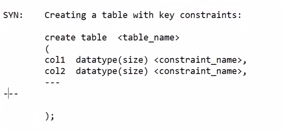
SQL> -- not null constraint violated.

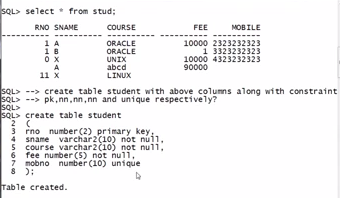
SQL> insert into student (sno, sname, course, fee) values(4,'null', 'xxx', '0');

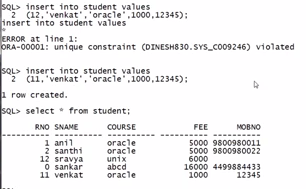
1 row created.

SQL> -- even its accepting invalid data like null, xxx, 0 as shown in the ex.

SQL> -- to avoid this in valid data we should use domain constraints.



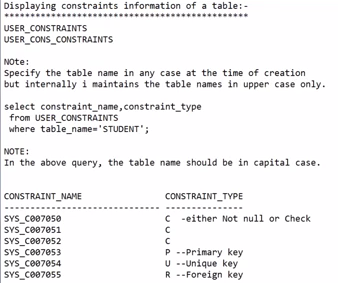


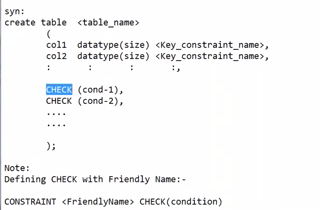
-- even its accepting invalid data like null, xxx, 0 as shown in the ex.

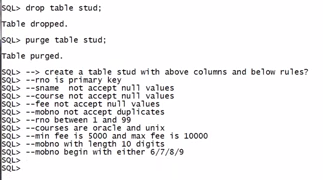
SQL> --CHECK can use any relational operator.

SQL> --domain constraints begining with CHECK and we can use condition in CHECK

SQL>







SQL> create table stud

2 (

3 sno number(5) primary key,

4 sname varchar2(10) not null,

5 course varchar2(15) not null,

6 fee number(10) not null,

7 mobno number(10) unique,

8 check rno between 1 and 99,

9 check course in ('oracle', 'unix'),

10 check fee between 5000 and 10000,

11 check len(mobno)=10,

12 check mobno=6% or mobno=7% or mobno=8% or mobno=9%

13 );

check rno between 1 and 99,

\*

ERROR at line 8:

ORA-00906: missing left parenthesis

SQL> ed

Wrote file afiedt.buf

1 create table stud

2 (

3 sno number(5) primary key,

4 sname varchar2(10) not null,

5 course varchar2(15) not null,

6 fee number(10) not null,

7 mobno number(10) unique,

8 check rno between (1 and 99),

9 check course in ('oracle', 'unix'),

10 check fee between 5000 and 10000,

11 check len(mobno)=10,

12 check mobno=6% or mobno=7% or mobno=8% or mobno=9%

13\* )

SQL> /

check rno between (1 and 99),

\*

ERROR at line 8:

ORA-00906: missing left parenthesis

SQL> ed

Wrote file afiedt.buf

1 create table stud

2 (

3 sno number(5) primary key,

4 sname varchar2(10) not null,

5 course varchar2(15) not null,

6 fee number(10) not null,

7 mobno number(10) unique,

8 check (rno between (1 and 99)),

9 check (course in ('oracle', 'unix')),

10 check (fee between 5000 and 10000),

11 check (len(mobno)=10),

12 check (mobno like 6% or mobno like 7% or mobno like 8% or mobno like 9%)

13\* )

SQL> /

check (rno between (1 and 99)),

\*

ERROR at line 8:

ORA-00907: missing right parenthesis

SQL> ed

Wrote file afiedt.buf

1 create table stud

2 (

3 sno number(5) primary key,

4 sname varchar2(10) not null,

5 course varchar2(15) not null,

6 fee number(10) not null,

7 mobno number(10) unique,

8 check (rno between 1 and 99),

9 check (course in ('oracle', 'unix')),

10 check (fee between 5000 and 10000),

11 check (len(mobno)=10),

12 check (mobno like 6% or mobno like 7% or mobno like 8% or mobno like 9%)

13\* )

SQL> /

check (mobno like 6% or mobno like 7% or mobno like 8% or mobno like 9%)

\*

ERROR at line 12:

ORA-00911: invalid character

SQL> ed

Wrote file afiedt.buf

1 create table stud

2 (

3 sno number(5) primary key,

4 sname varchar2(10) not null,

5 course varchar2(15) not null,

6 fee number(10) not null,

7 mobno number(10) unique,

8 check (rno between 1 and 99),

9 check (course in ('oracle', 'unix')),

10 check (fee between 5000 and 10000),

11 check (len(mobno)=10),

12 check (mobno like '6%' or mobno like '7%' or mobno like '8%' or mobno like '9%')

13\* )

SQL> /

check (rno between 1 and 99),

\*

ERROR at line 8:

ORA-00904: "RNO": invalid identifier

SQL> ed

Wrote file afiedt.buf

1 create table stud

2 (

3 sno number(5) primary key,

4 sname varchar2(10) not null,

5 course varchar2(15) not null,

6 fee number(10) not null,

7 mobno number(10) unique,

8 check (sno between 1 and 99),

9 check (course in ('oracle', 'unix')),

10 check (fee between 5000 and 10000),

11 check (len(mobno)=10),

12 check (mobno like '6%' or mobno like '7%' or mobno like '8%' or mobno like '9%')

13\* )

SQL> /

check (len(mobno)=10),

\*

ERROR at line 11:

ORA-00904: "LEN": invalid identifier

SQL> ed

Wrote file afiedt.buf

1 create table stud

2 (

3 sno number(5) primary key,

4 sname varchar2(10) not null,

5 course varchar2(15) not null,

6 fee number(10) not null,

7 mobno number(10) unique,

8 check (sno between 1 and 99),

9 check (course in ('oracle', 'unix')),

10 check (fee between 5000 and 10000),

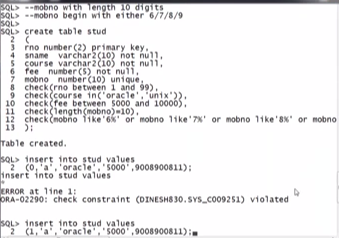
11 check (length(mobno)=10),

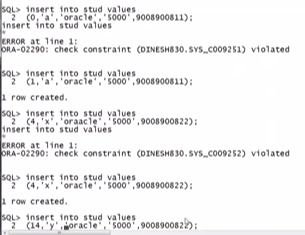
12 check (mobno like '6%' or mobno like '7%' or mobno like '8%' or mobno like '9%')

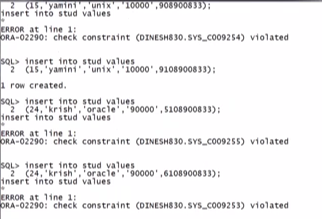
13\* )

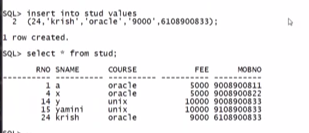
SQL> /

Table created.









Here all values are valid , because we have used domain constraints also.

SQL> --by using key constraints we can ristrict weather we can give data in that column or not

SQL> --and we can stop gtiving duplicate values

SQL> --but by using domain constraints, we can decide which type of data the table should accept by using CHECK constraint.