**Constraints**

**Day-01**

**26-03-2025**

**============**

Why Constraints?

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

show user;

create table clients(

clientdId number(10),

firstName varchar2(50),

lastName varchar2(20)

);

desc clients;

insert into clients values(102031,'Ravi','Kumar');

insert into clients values(103021,'Kishor', 'Rao');

select \* from clients;

insert into clients values(102031,'Ganesh','Kumar');

select \* from clients;

insert into clients values(null,'Bharath','Krishna');

insert into clients values(121212,null,null);

select \* from clients;

-> Till now, while creation of the tables we have not used any constraints. That's why can create the table with inaccurate data (data with duplication and data with null).

-> When we want to create the tables based on the business logic we must be use "constraints".

What is Constraints?

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

-> constraint is a rule to define the database object according to the business logic (to get the data accurately).

->Constraints also called as "SQL Constraints" or also called as "Integrity Constraints".

-> in two levels:

1) Column level constraints

2) Table level constraints

-> while implementing the column if we can define the constraints those are called as "column-level constraints".

Ex:

table create <Table-name>(

col1 datatype constraint-type,

col2 datatype

);

-> When we can apply the constraint at the end of the table or after the columns implementations called as "table level constraints".

Ex:

create table <table-name>(

col1 datatype,

col2 datatype,

....,

constraint <constraint-name> constraint-type(col-name)

);

Here:

the table level constraints can be defined with "constraint class".

-> the constraint class can be represented with "constraint keyword".

Types of Constraints:

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

-> Different types of constraints:

1) Unique Constraints:

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

-> When we want to create a table without any duplication then we can use "unique constraints".

-> With the unique constraints we can't avoid to define null values for the table.

-> We can define the unique constraint more than one column at a time.

i) Unique constraints at column level:

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

Syntax:

create table <table-name>(

col1 datatype unique,

col2 datatype,

col3 datatype unique,

......

);

create table clients(

client\_id number(10) unique,

first\_name varchar2(50),

last\_name varchar2(50)

);

insert into clients values(null,null,null);

commit;

select \* from clients;

insert into clients values(102030,'Ravi','Kumar');

insert into clients values(112233, 'Uma','Mahesh');

commit;

select \* from clients;

insert into clients values(102030,'Ganesh','Kumar');

ii) Unique Constraints at table level:

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

Syntax:

create table <table-name>(

col1 datatype,

col2 datatype,

.....

constraint <constraint-name> unique(col1)

);

create table clients(

clientId number(10),

firstName varchar2(50),

lastName varchar2(50),

gender varchar2(10),

company varchar2(30),

constraint unique\_clients unique(company)

);

insert into clients values(1,'Ravi','Kumar','Male','Ashok IT');

commit;

insert into clients values(2,'Ganesh','Kumar','Male','Ashok IT');

iii) Unique Constraint for more than one column:

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

Syntax:

create table <table-name>(

col1 datatype,

col2 datatype,

.....

constraint <constraint-name> unique(col1,col2)

);

create table clients(

cid number(10),

firstName varchar2(50),

lastName varchar2(50),

constraint clients\_unique unique(firstName,lastName)

);

insert into clients values(112233,'ravi','kumar');

commit;

insert into clients values(112223,'ganesh','kumar');

Note:

-----

-> While defining the unique constraints at column level, the system can provide the constraint name.

-> While defining the unique constraints at table level, developer should provide constraint name using constraint class.

-> constraints can always define with DDL commands only.