**DATA INTEGRITY CONSTRAINTS**

**CONSTRAINTS**

Constraints are known as set of rules / business rules. Constraints, force the data base to accept only valid values in to the tables.

Categories of CONSTRAINTS: 3

🡪1) Key Constraints

🡪 2) Domain Constraints

🡪 3) Referential Integrity constraints / Ref Constraints

**1) KEY CONSTRAINTS**

These constraints will verify individual values. Key constraints are divided into 3 types.

**UNIQUE**

It doesn't allow duplicates, but allow null values.

Ex: phone numbers, mailid, etc...

email varchar(30) UNIQUE,

aadhar numeric(12) UNIQUE,

**NOT NULL**

It doesn't allow null values, but allow duplicate values.

Ex: empNames, CityNames, .....

sname varchar(20) NOT NULL,

**PRIMARY KEY**

It doesn't allow duplicates and doesn't allow null values. Generally, a Primary key is used to identify any record Uniquely. Only one primary key constraint is allowed per a table.

Ex: bank account numbers, empid

EXAMPLE:

Table creation without constraints:

CREATE TABLE STUD

(

RNO NUMERIC(2),

SNAME VARCHAR(10),

COURSE VARCHAR(10),

FEE NUMERIC(5),

MOBILE NUMERIC(10)

);

INSERT INTO STUD VALUES(1,'A','ORACLE',10000,2323232323);

INSERT INTO STUD VALUES(1,'B','ORACLE',1,3323232323);

INSERT INTO STUD VALUES(0,'X','UNIX',10000,4323232323);

INSERT INTO STUD VALUES(NULL,'A','abcd',90000,NULL);

INSERT INTO STUD VALUES(11,'X','LINUX',NULL,NULL);

INSERT INTO STUD VALUES(20,'AJAY','ORACLE',10000,9866987700);

INSERT INTO STUD VALUES(21,'ASHOK','ORACLE',10000,986698772);

SELECT \* FROM STUD;

Note:

In the above table the we can find invalid values.

How to avoid / restrict invalid values into database table?

BY defining constraints on the tables before data to be inserted into the table.

**SYNTAX**

Creating table with key constraints:

create table <table\_name>

(

col1 datatype(size) <constraint\_name>,

col2 datatype(size) <constraint\_name>,

---- ----- ---- );

Ex:

create a table student with columns rno, sname, course, fee and mobile along with constraints pk, nn, nn, nn and unique respectively?

create table stud1

(

rno numeric(2) primary key,

sname varchar(10) not null,

course varchar(10) not null,

fee numeric(5) not null,

mobile numeric(10) unique

);

**Note**

Even after the key constraints on the table, still we have invalid values.

We can eliminate them by using DOMAIN constraints.

**2) DOMAIN constraint**

We can define conditions on the columns by using domain constraint. Before inserting a value, the value is verified based on condition. The name of domain constraint is CHECK.

**syntax**

col datatype(size) <key constraint>,

check (col <condition>)

**Ex**

rno number(3) primary key,

check (rno between 1 and 999)

**Ex**

create a table student with columns rno, sname, course, fee and mobile along with constraints pk, nn, nn, nn and unique respectively?

create table stud2

(

rno numeric(2) primary key,

sname varchar(10) not null,

course varchar(10) not null,

fee numeric(5) not null,

mobile numeric(10) unique,

check(rno between 1 and 99),

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

check(fee between 5000 and 10000),

check(len(mobile)=10),

check(mobile like'6%' or mobile like'7%' or mobile like'8%' or mobile like'9%')

);

How to get constraints Information?

select CONSTRAINT\_NAME, CONSTRAINT\_TYPE

from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

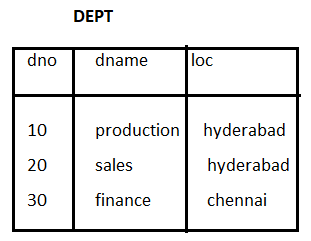
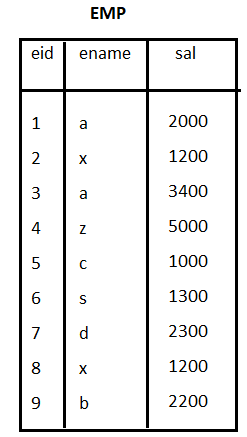
where TABLE\_NAME='stud2';

How to get not null constraints information?

sp\_help stud2;

**NORMALIZATION AND DENORMALIZATION CONCEPTS**

To eliminate communication gap, we can use DENORMALIZATION and NORMALIZATION methods.



NOTE

By using above tables, we are unable fetch the complete data of an object like, department name of any employee, number of emps in a dept and etc. This problem is known as communication gap between end user and data base.

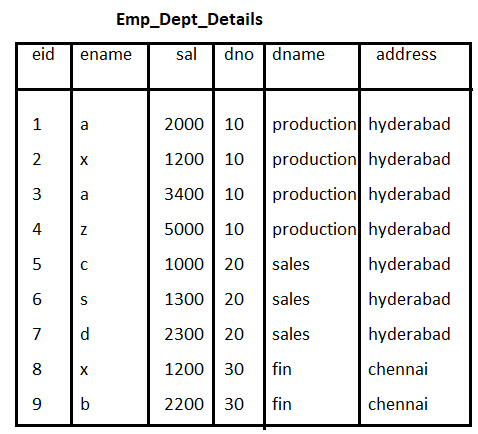
**To eliminate communication gap, we have 2 methods.**

🡪DENORMALIZATION

🡪NORMALIZATION

**DENORMALIZATION**

Maintaining all necessary information in one big table is known as Denormalized method.



Advantage :- Communication gap is eliminated.

Disadvantage:

data duplicity, Occupy more disk space, Data retrieval time is very long

**NORMALIZATION**

The Process of Dividing big table in to sub tables, until Max. data duplicity is reduced is known as normalization process.

i) Ist NF(normal form):

Dividing big table into sub tables based on repeated groups of data.

emp dept

----- ------

eid ename sal dno dname loc

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

1 a 2000 10 production hyderabad

2 x 1200 10 production hyderabad

3 a 3400 10 production hyderabad

4 z 5000 10 production hyderabad

5 c 1000 20 sales hyderabad

6 s 1300 20 sales hyderabad

7 d 2300 20 sales hyderabad

8 x 1200 30 fin chennai

9 b 2200 30 fin chennai

Ex:

Emp Table:

create table emp

as

select eid,ename,sal from emp\_dept\_details;

DEPT table:

create table dept

as

select dno,dname,loc from emp\_dept\_details;

ii) IInd NF:

\*\*\*\*\*\*\*\*\*\*\*\*

Eleminate duplicate records and defining

primary keys in the above tables

emp dept

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

eid ename sal dno dname loc

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

1 a 2000 10 production hyderabad

2 x 1200 20 sales hyderabad

3 a 3400 30 fin chennai

4 z 5000 PK

5 c 1000

6 s 1300

7 d 2300

8 x 1200

9 b 2200

PK

Ex: Eleminating Duplicate records

create table dept1

as

select distinct dno,dname,loc from dept;

iii) IIIrd NF: [ Boyce Codd Normal Form ] (BCNF)

\*\*\*\*\*\*\*\*\*\*\*\*\*

We can define relation between the tables.

Consider PK of parent table and define Fk column

in the child table.

We can define FK column , by using Referential Integrity Constraint.

emp dept

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

eid ename sal dno dno dname loc

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

1 a 2000 10 10 production hyderabad

2 x 1200 10 20 sales hyderabad

3 a 3400 10 30 fin chennai

4 z 5000 10 PK

5 c 1000 20

6 s 1300 20

7 d 2300 20

8 x 1200 30

9 b 2200 30

PK FK

Adv: No communication gap, Duplicity is reduced, Occupy less disk space, Data search time is reduced.

**3) REFERENTIAL INTEGRITY CONSTRAINT / REF CONSTRAINT**

By using this we can define relation between tables.

Foreign key column contains only values from primary key. Foreign key contains duplicates and null values also.

A table which has primary key is considered as parent/Master/Base table.

A table which has foreign key is known as Child/Detailed/Derived table.

**REFERENCES**

It is used to define a FK column in the child table, by using parent table PK column.

Defining FK with default name:

syntax:

<col\_name> datatype(size) REFERENCES <parent\_table>(PK\_column);

Examples:

consider the tables comp\_dtls and prod\_dtls.

And we are maintaining set of products from each company.

Based on this create the tables and maintain relation between

the tables?

Ex: create comp\_dtls as parent table

Ex: create prod\_dtls as child table

create table comp\_dtls

(

cmpid char(5) primary key,

cmpname varchar2(20) not null,

Country varchar2(20) not null

);

create table prod\_dtls

(

pid char(4) primary key,

pname varchar2(20) not null,

cost number(7,2),

mfg date,

warrenty varchar2(10),

cmpid char(5) references comp\_dtls(cmpid)

);