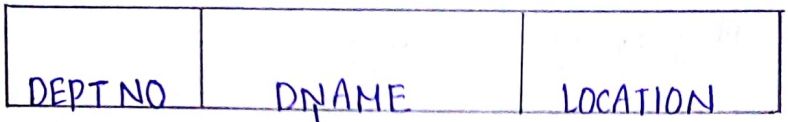
* It is typically a known value
  (i.e. it is not hidden)
* May be used to establish referential constraints.

Example: create table dept(dept_no integer, dname varchar2(20) UNIQUE,
location varchar2(20));

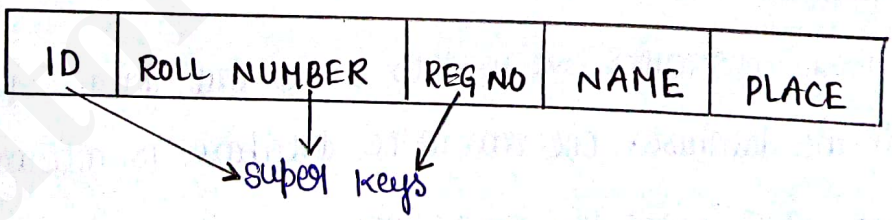


UNIQUE KEY constraint
(no row may duplicate a value in the constraint's column)

(c) Super key: An attribute or a combination of attribute that is used to identify the records uniquely is known as super key.

* A table have many super keys.
* It is also allow duplicate values also.

Examples:



(1) {ID, ROLL NUMBER, REG NO}
(2) {ID, PLACE, REG NO}
(3) {ID, NAME, REG NO}
.....

} super keys Examples.

(d) Candidate Key:

candidate key can be defined as minimal super key or irreducible super key. In other words an attribute or combination of attribute that identifies the record uniquely but none of its proper subsets can identify the records uniquely.

Example:

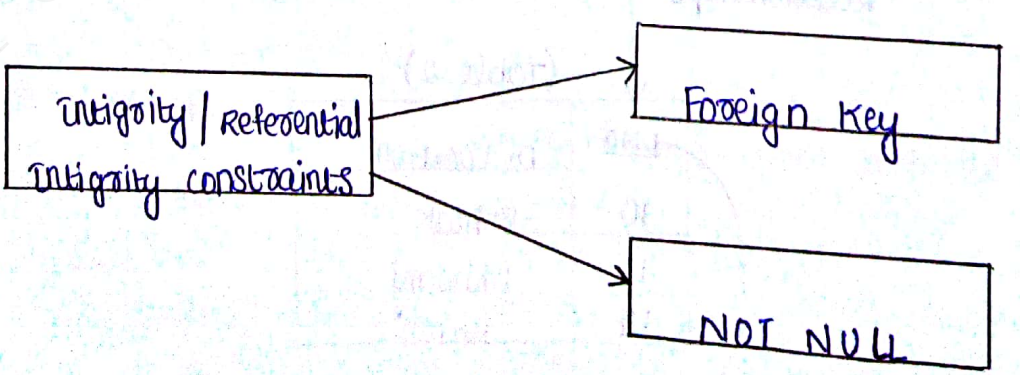
STUID	FIRSTNAME	LAST NAME	COURSE

{STUID}
{FIRSTNAME, LAST NAME} } examples of candidate key.
{STUID, COURSE}

→ primary key, UNIQUE keys are used practically but not candidate key and super key.

3. Intigrity constraints:

Intigrity constraints means depending upon constraints enter correct data or not. That means entered data as per the constraints (or) not.



(a) Foreign Key:

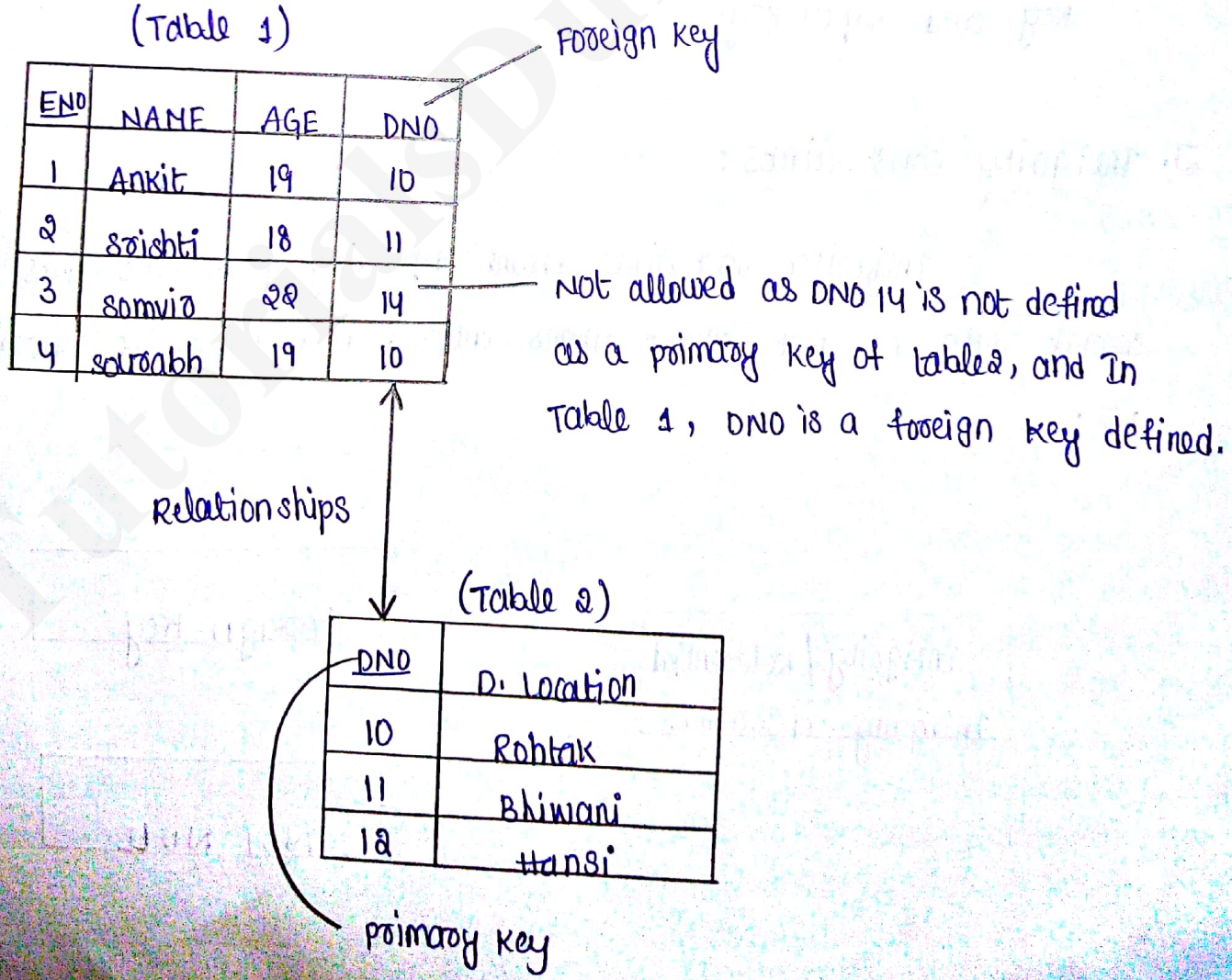
It is used to enforce referential integrity between tables in a relational database. primary key field cann't containe null values but foreign key field can contain null values.

Foreign key constraint can be defined by a CREATE TABLE statement or an ALTER TABLE statement.

Characteristics of a foreign key:

* one or more per base table
* can be null
* one or more columns corresponding to like columns in the parent table.

Example:



(6) NOT NULL:

The NOT NULL constraint enforces a column to NOT accept NULL values. This enforces a field to always contain a value, which means that you can't insert a new record, or update a record without adding a value to this field.

Example: create table student(sid integer NOT NULL, Name varchar2(20), Age integer);

The above query will declare the sid field of student table will not take NULL value.

Another key constraints:

1. Composite key:

If use multiple attributes to create a primary key then that primary key is called composite key. That means set of super keys and candidate keys.

2. Alternate key:

Alternate key can be any of the candidate keys except for the primary key.

Example: "Name, Address" as it is only other candidate key which is not a primary key.

SQL allows a table (relation) to have two or more tuples that are identical in all their attributes values. Hence, an SQL table is not a set of tuple, because a set does not allow two identical numbers; rather it multiset of tuples.

A basic query statement in SQL is the SELECT statement.

The SELECT statement used in SQL has no relationship to the SELECT operation of relational algebra.

The SELECT statement:-

The syntax of this command is:

SELECT <attribute list>
FROM <table list>
WHERE <condition>;

Query 1: Retrieve the birthday and address of employee(s) whose name is John B. Smith

Q1: SELECT BDATE, ADDRESS
FROM EMPLOYEE
WHERE FNAME='John' AND MINIT='B' AND LNAME='SMITH'

Query: Retrieve the name and address of all employee who work for the Research dept.

Q: SELET FNAME, LNAME, ADDRESS
FROM EMPLOYEE, DEPARTMENT
WHERE DNAME = 'Research' AND DNUMBER=DNO

1. Relational Model, concepts of domain, attributes, ~~tyfte~~ tuple, relation, importance of NULL values.

A) Relational Model :

$R(A_1, A_2, ----, A_n)$ is a relational schema of degree n denoting that there is a relation R having as its attributes $A_1, A_2, ---- A_n$.

By convention, Q, R and S denote relation names.

By convention, q, r and s denote relation states. For example, r(R) denotes one possible state of relation R. If R is understood from context, this could be written, more simply, as r.

By convention, t, u, and v denote tuples.

The "dot notation" R.A (Eg, STUDENT.Name) is used to qualify an attribute name, ~~used~~ usually

For the purpose of distinguishing it from a same-named attribute in a different relation

(Eg. DEPARTMENT.Name).

Domain concepts :-

A (usually named) set/universe of atomic values, where by "atomic" we mean simply that, from the point of view of the database, each value in the domain is indivisible (i-e cannot be broken down into component parts).

Examples of domains:

USA_phone_number : string of digits of length ten

SSN : string of digits of length nine

Name: String of characters beginning with an upper case letter

GPA: a real number between 0.0 and 4.0.

Sex: a member of the set {female, male}

Dept_code: a member of the set {CMPS, MATH, ENGL, PHYS, PSYC, ...}

These are all logical descriptions of domains. For implementation purposes, it is necessary to provide descriptions of domains in terms of concrete data types (or formats) that are provided by the DBMS (such as string, int, boolean), in a manner analogous to how programming languages have intrinsic data types.

Attributes: The name of the role played by some value (coming from some domain) in the context of a relational schema. The domain of attribute A is denoted dom(A).

Tuple: A tuple is a mapping from attributes to values drawn from the respective domains of those attributes. A tuple is intended to describe some entity (or relationship between entities) in the miniworld.

{ Name ----> " Rumpelstiltskin ", Sex ---> Male ,

IQ → 143 }.

Relation: A (named) set of tuples all of the same form (i.e. having the same set of attributes). The term table is a loose synonym. (some database purists would agree argue that a table is "only" a physical manifes -tation of a relation).

Importance of NULL values:

Every column in a table should contain a value, through there may be times when the value is unknown

For example, consider the following table which stores data relating to stories on CD supplies.

SUPPLIER

supplierid	Name	Address	phone	Fax

columns of the SUPPLIER table.

1) supplier id
2) supplier name
3) Address
4) supplier phone
5) supplier Fax.

To communicate with suppliers you will need their name, Address, phone number, and fax. If you do not know one or more of those pieces of data, you will not know that to enter into its corresponding column.

A zero, by constrast, is an INT or DECIMAL value. If a stores on CD supplier gave the company a thousand customers who placed an order, the Retail price column for that CD would contain a zero.

List SQL Grouping functions with examples & explain each function with example?

What is SQL single row function? By means of suitable examples illustrate the usage of SQL date, character and number functions.

Arithmetic & Logical Operations in Basic SQL :-

Arithmetic operators can perform arithmetic operations on numeric operands involved. Arithmetic operators are -

1. Addition (+)
2. subtraction (-)
3. Multiplication (*)
4. division (/).

The addition (+) and subtraction (-) operators can also be used in date arthimetic functions.

Operator	Meaning
+ (add)	Addition
- (subtract)	Subtraction
* (multiply)	Multiplication
/ (divide)	division
% (Modulo)	Modulo division.

Syntax :-

```
select   <expression1> [arithmetic operator]<expression2> ...
from     table_name
where    [expression/condition];
```