Types of Constraints. Three Types

1. Tuple level integrity constraints : Primary Key & Alternate Key
2. Domain level integrity constraints: Check & Not Null
3. Referential Integrity: Foreign Key

1. **Tuple Level Integrity Constraints**: These constraints effects tuples in a table
    a. Primary Key
    b. Alternate Key

    a. ***Primary Key:*** Let PK be subset of attributes of R. PK is set to be defined as primary key if and only if it satifies the following properties:
        i. Uniqueness: No two tuples will have same primary key value (PK - Value)
        ii. Irreducibility: No proper subset of PK will satisfy uniqueness property

    Ex: student(<u>regno</u>, sname, branch, section)

    The above table student contains {regno} as primary key. This key satisfies uniqueness property. This key cannot be reduced further as it does contain only one attribute; staisfying irreducibility property.

    Ex: booking(<u>sid,bid</u>, day)

    The above table booking contains {sid, bid} as primary key. No two tuples will have same combination of sid and bid values. This key satisfies uniqueness property. Consider a proper subset to the key say {sid} and {bid}. These two subsets doesn't satisfy uniqueness property. Thus the primary key {sid, bid} cannot be reduced.

    Advantages: i. Identifies a tuple uniquely ii. Avoids duplicate entry of a tuple with existing primary key value

    Note: Primary key formed using one attribute is called simple primary key; Primary key formed using multiple attributes is called composite primary key

    SQL:

```
alter table student add constraint stu_p primary key(regno)
alter table booking add constraint book_p primary key(sid,bid)
```

    b. ***Alternate Key:*** Let AK be subset of attributes of R. AK is set to be defined as Alternate key if and only if it satifies the following properties:
        i. Uniqueness: No two tuples will have same Alternate key value (AK - Value)
        ii. Irreducibility: No proper subset of AK will satisfy uniqueness property

    Ex: student(<u>regno</u>, adharno, sname, branch, section)

    The above table student contains {adharno} as Alternate key. This key satisfies uniqueness property. This key cannot be reduced further as it does contain only one attribute; staisfying irreducibility property.

    Note: Alternate key formed using one attribute is called simple Alternate key; Alternate key formed using multiple attributes is called composite Alternate key. Proper subset of alternate key doesn't satisfies uniqueness property. Thus Alternate key cannot be reduced.

Advantages: i. Identifies a tuple uniquely ii. Avoids duplicate entry of a tuple with existing Alternate key value

SQL:
alter table student add constraint stu_ak unique(adharno)

Difference between Primary Key and Alternate Key: Primary Key doesn't accepts nulls. Alternate key accepts nulls.

2. **Domain Level Integrity Constraints**:
   a. Check Constraint
   b. Not Null Constraint

a. Check Constraint: Limits column values to a range of values or specific values

Demonstration:
Create table ab(sno integer,salary integer,gender char(1),city varchar2(15))

alter table ab add constraint ab-c1 check(salary>=1500 and salary<=10000) /* Limits salary column to accept values within the specified range */

alter table ab add constraint ab-c2 check(gender in('M','F') /* Limits the column gender to accept only specific values */

insert into ab values(1,1500,'M','Guntur') /*Insertion is Successful*/
insert into ab add values(2,1000,'M','Guntur') /*ab -C1 violated*/
insert into ab add values(3,null,'m','Tenali') /*ab-C2 violated*/
insert into ab add values(3,null,'F','Tenali') /*Successful*/
insert into ab add values(2,1600,'M','Guntur') /*Successful*/

b. Not Null Constraint: This constraint limits a column in not accepting null values

Demonstration:
Alter table ab modify(salary integer not null)
insert into ab values(5,1800,'F','Guntur')/*Successful*/
insert into ab values(5,null,'F','Hyd')/*Salary doesn't accept null*/

3. **Referential Integrity Constraint of Foreign Key Constraint**:
Let R1 and R2 be two relations. FK be subset of attributes of R1. CK be subset of attributes of R2 which is candidate key of R2. FK is said to be declared as foreign key to R1 if and only if for every FK-value there exists one matching CK-value in R2 (but not reverse).

Ex:
Consider the following schema
Emp(empno,ename,job,mrgno,deptno,salary)
Dept(deptno,dname,location)

Primary key of the table emp is {empno}. Primary key of the table dept is {deptno}

The table emp contains {deptno} as foreign key, because {deptno} is primary key (candidate key) to the table dept.

SQL Implementation.

Alter table emp add constraint emp-f foreign key(deptno) references dept

- Table containing foreign key is called as Referencing relation
- Table from where candidate key is imported as foreign key to referencing relation is called referenced relation or target relation.
- In the above example, the table emp is called as Referencing Relation and the table dept is called as Referenced relation or target relation
- Every value under the column deptno in emp table must be present as primary key value in dept but not reverse
- Every foreign key value in the table emp must be present as primary key value in the table dept but not reverse.

Defintion of Referential Intergity:

For every foreign key value in referencing relation there must exists one matching candidate key value in referenced relation.

Update operations with respect to Referential Integrity (or Referenced Relations):

(a) Insert

Initial data insertion must be done into Referenced Relations; This is because initial data insertion into Referencing relations violates Referential Integrity.

Eg: Consider the following schema
Emp(empno,ename,job,mgrno,deptno,salary)
Dept(deptno,dname,location)

Primary key of the table emp is {empno}.Primary key of the table dept is {deptno}

Foreign key of the table is {deptno}

Referenced relation is Dept ad referencing relation is emp

Initial data insertion must be done into the table dept.

If intial insertion is tried into Referencing relation i.e, emp which is not possible because foreign key value in emp will not be present as matching candidate key value in referenced relation i.e, dept. Hence initial data insertion must be always done with referenced relation.

(b) Delete

Consider the schema
emp(empno,ename,job,mgrno,deptno,salary)
dept(deptno,dname,location)

Consider the above relations with the following tuples

EMP

EMPNO	ENAME	JOB	MGRNO	DEPTNO	SALARY
1	Sri	a	2	10	2000
2	Hari	b	3	10	1500

DEPT

DEPTNO	DNAME	LOCATION
10	Sales	Guntur
20	XYZ	Tnl

Primary key for the relation EMP is {empno}. Primary key for the relation DEPT is {deptno}. Foreign key for the relations EMP is {deptno}

i. Deletion in referencing relation is possible without violating referential integrity

ii. Deletion of a tuple in referenced relation is done by following observations:

1. Deleting a tuple from referenced relations with a candidate key value is possible if and only if there is no matching foreign key value in referencing relation. This doesnt violate referential integrity.
   Ex: tuple in the relation DEPT with deptno 20 can be deleted without violating referential integrity.
2. Deleting a tuple from referenced relation with a candidate key value is not possible if and only if there is a matching foreign key value in referencing relation. If deletion is done then it violates referential integrity.

Ex: deleting a tuple from the relation DEPT with deptno 10 violates referential integrity; because deptno 10 is present as foreign key value in referencing relation EMP.

If deletion is to be done in referenced relation with a candidate key value which is having matching foreign key value in referencing relation then deletion must be done in both the relations i.e. deleting the tuple with candidate key value in referenced relation along with deletion of tuple(s) with matching foreign key value in referencing relation. This is called as cascade mode of deletion.

SQL: alter table emp add constraint emp_f foreign key(deptno) references dept on delete casade

Ex: deleting the tuple from the relation dept with deptno 10 also deletes tuples in relation emp with deptno 10

Restricting the deletion operation in both referenced and referencing relations is called restrict mode of deletion.

(c) Update

Consider the schema
emp(empno,ename,job,mgrno,deptno,salary)
dept(deptno,dname,location)

Consider the above relations with the following tuples

EMP

EMPNO	ENAME	JOB	MGRNO	DEPTNO	SALARY
1	Sri	a	2	10	2000
2	Hari	b	3	10	1500

DEPT

DEPTNO	DNAME	LOCATION
10	Sales	Guntur
20	XYZ	Tnl

Primary key for the relation EMP is {empno}. Primary key for the relation DEPT is {deptno}. Foreign key for the relation EMP is {deptno}

i. updation in referencing relation is possible without violating referential integrity if and only if the foreign key value in referencing relation gets replaced with existing candidate key value of referenced relation

ii. Updation of a tuple in referenced relation is done by following observations:

1. Updating the candidate key value in referenced relation is possible if and only if there is no matching foreign key value in referencing relation. This doesnt violate referential integrity. Ex: tuple in the relation DEPT with deptno 20 can be updated with new value say 40, never violates referential integrity.
2. Updating the candidate key value in referenced relation is not possible if and only if there is a matching foreign key value in referencing relation. If updation is done then it violates referential integrity. Ex: updating candidate key value in referenced relation DEPT with deptno 10 violates referential integrity; because deptno 10 is present as foreign key value in referencing relation EMP.

   If updation is to be done in referenced relation with a candidate key value which is having matching foreign key value in referencing relation then updation must be done in both the relations i.e. updating the tuple with candidate key value in referenced relation along with updation of tuple(s) with matching foreign key value in referencing relation. This is called as cascade mode of updation.

   Ex: updating the candidate key value in referenced relation dept with deptno 10 also updates tuples in referencing relation emp with deptno 10

   Restricting the updation operation in both referenced and referencing relations is called restrict mode of updation.

Foreign Key - Nulls?

(a) Foreign Key as subset of Primary Key

consider the following instance of supplier parts database

Supplier

Sid	Sname	City	Rating
S1	Sudhir	Gnt	4
S2	Hari	Tnl	5
S3	Vamsi	Gnt	3

Parts

Pid	Pname	Color	Weight
P1	Bolt	Red	60

P2	Bolt	Green	80
P3	Screw	Red	100
P4	Pin	Green	20

Supply

Sid	Pid	Quantity
S1	P1	15
S1	P2	20
S2	P1	5
S2	P2	25
S3	P3	10

Primary key of the table supplier is {sid}

Primary key of the table parts is {pid}

Primary key of the table supply is {sid,pid}

The table supply contains two foreign keys. They are {sid} and {pid}. These foreign keys are involved within the formation of primary key of the table supply.

Hence if foreign key is involved within the formation of primary key of the table then foreign key doesn't accept null values

(b) Foreign key not a subset to primary key

Consider the following schema

emp(empno,ename,job,hiredate,salary,comm,deptno)

dept(deptno,dname,location)

Primary of the table emp is {empno}. Primary key of the table dept is {deptno}. Foreign key for the table emp is {deptno}. Foreign key of emp is not subset to primary key of emp. Hence the foreign key of emp may accpet null values.

If Foreign key is not involved within the formation of primary key of the table then foreign key may accpet null values.