netlink
digital solutions

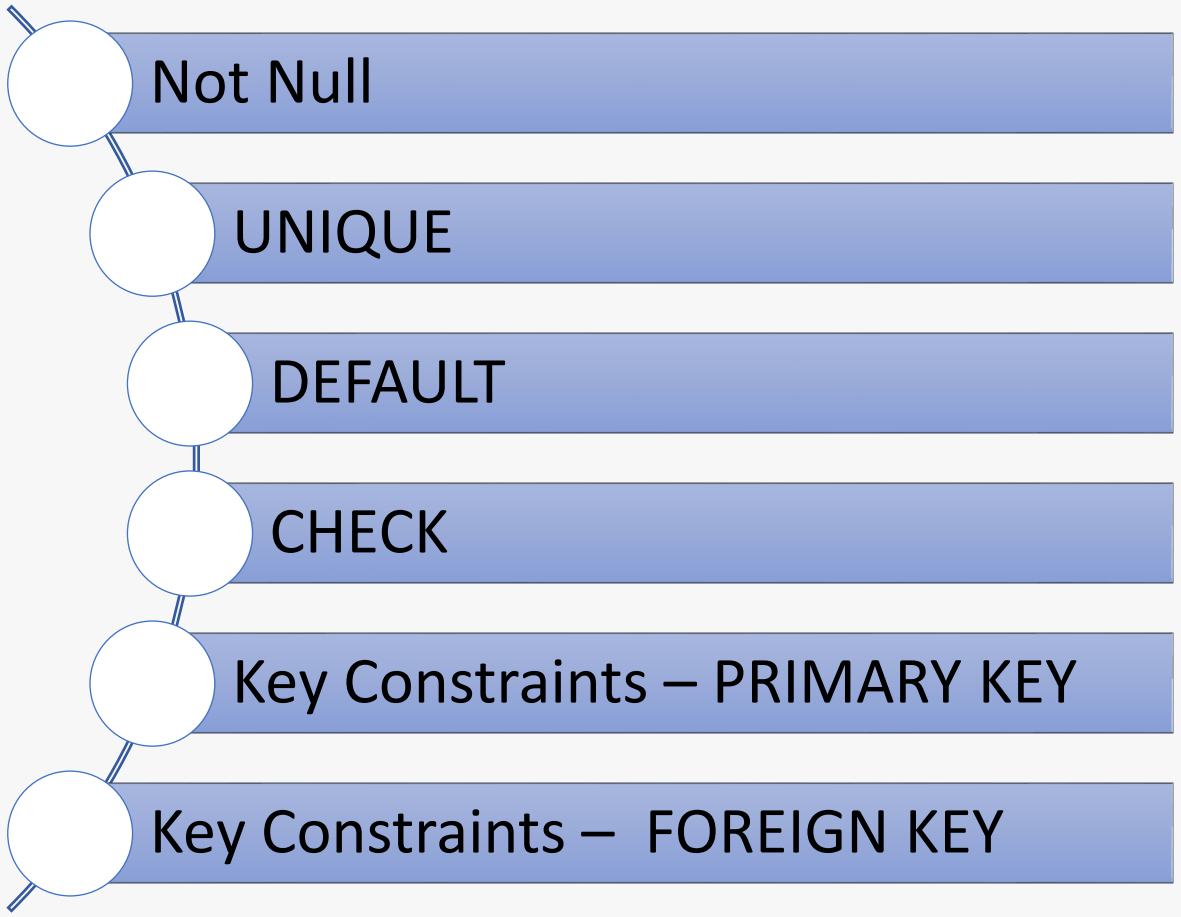
DATABASE

www.netlink.com

Constraint

Constraints enforce limits to the data or type of data that can be inserted/updated/deleted from a table. The whole purpose of constraints is to maintain the **data integrity** during an update/delete/insert into a table.

Types of constraints



NOT NULL

- NOT NULL constraint makes sure that a column does not hold NULL value.
- When we don't provide value for a particular column while inserting a record into a table, it takes NULL value by default.
- By specifying NOT NULL constraint, we can be sure that a particular column(s) cannot have NULL values.

```
CREATE TABLE STUDENTS(

ROLL_NO INT NOT NULL,

STU_NAME VARCHAR (35) NOT NULL,

STU_AGE INT NOT NULL,

STU_ADDRESS VARCHAR (235) ,

PRIMARY KEY (ROLL_NO) );
```

Adding NOT NULL Constraint to existing columns

ALTER TABLE table_name
ALTER COLUMN column_name SET NOT NULL;

To add set multiple NOT NULL constraint to multiple columns,

ALTER TABLE table_name
ALTER COLUMN column_name_1 SET NOT NULL,
ALTER COLUMN column_name_2 SET NOT NULL,
…;

UNIQUE

- UNIQUE Constraint enforces a column or set of columns to have unique values

★★★ If a column has a unique constraint, it means that column cannot have duplicate values in a table.

```
CREATE TABLE STUDENT(
ROLL_NO INT NOT NULL,
STU_NAME VARCHAR (35) NOT NULL UNIQUE,
STU_AGE INT NOT NULL,
STU_ADDRESS VARCHAR (35) UNIQUE,
PRIMARY KEY (ROLL_NO) );
```

To add a constraint to a table, you use ALTER TABLE ADD CONSTRAINT statement:

ALTER TABLE table_name
ADD CONSTRAINT constraint_name constraint_definition;

DEFAULT

The DEFAULT constraint provides a default value to a column when there is no value provided while inserting a record into a table.

```
CREATE TABLE STUDENT(
ROLL_NO    INT   NOT NULL,
STU_NAME VARCHAR (35) NOT NULL,
STU_AGE INT NOT NULL,
EXAM_FEE INT   DEFAULT 10000,
STU_ADDRESS VARCHAR (35) ,
PRIMARY KEY (ROLL_NO) );
```

To change a default value of the column, you use ALTER TABLE ALTER COLUMN SET DEFAULT or DROP DEFAULT

alter table table_name
alter column column_name [SET DEFAULT value | DROP DEFAULT];

CHECK

This constraint is used for specifying range of values for a particular column of a table

When this constraint is being set on a column, it ensures that the specified column must have the value falling in the specified range.

```
CREATE TABLE STUDENT(
 ROLL_NO   INT  NOT NULL CHECK(ROLL_NO
>1000) ,

 STU_NAME VARCHAR (35)  NOT NULL,

 STU_AGE INT  NOT NULL, EXAM_FEE
INT DEFAULT 10000,
 STU_ADDRESS VARCHAR (35) ,
 PRIMARY KEY (ROLL_NO) );
```

Example

In the above example we have set the check constraint on ROLL_NO column of STUDENT table. Now, the ROLL_NO field must have the value greater than 1000

The basic syntax of ALTER TABLE to **ADD CHECK CONSTRAINT** to a table is as follows −

```
ALTER TABLE table_name
ADD CONSTRAINT MyUniqueConstraint CHECK
(CONDITION);
```

PRIMARY KEY

Uniquely identifies each record in a table. It must have unique values and cannot contain nulls.

If there is a two column with combination of unique and not null constraint then both are eligible to become primary key

In this case we need to select any one column for primary key

Characteristics of primary key

- It's a combination of unique and not null constraint
- A table can contain only one primary key
- Primary key should not null

To add a constraint to a table, you use ALTER TABLE ADD Primary key CONSTRAINT statement:

```
ALTER TABLE table_name ADD PRIMARY KEY (column);
```

Student table

Roll_NO	Stu_Name	Stu_Age	Stu_Address
101	Alex	20	UK
201	Max	19	US
301	jack	20	India

Example:

```
CREATE TABLE STUDENT(
ROLL_NO    INT   NOT NULL,
STU_NAME VARCHAR (35)   NOT NULL UNIQUE,
STU_AGE INT NOT NULL,
STU_ADDRESS VARCHAR (35) UNIQUE,
PRIMARY KEY (ROLL_NO) );
```

FOREIGN KEY

Foreign keys are the columns of a table that points to the primary key of another table. They act as a cross-reference between tables.

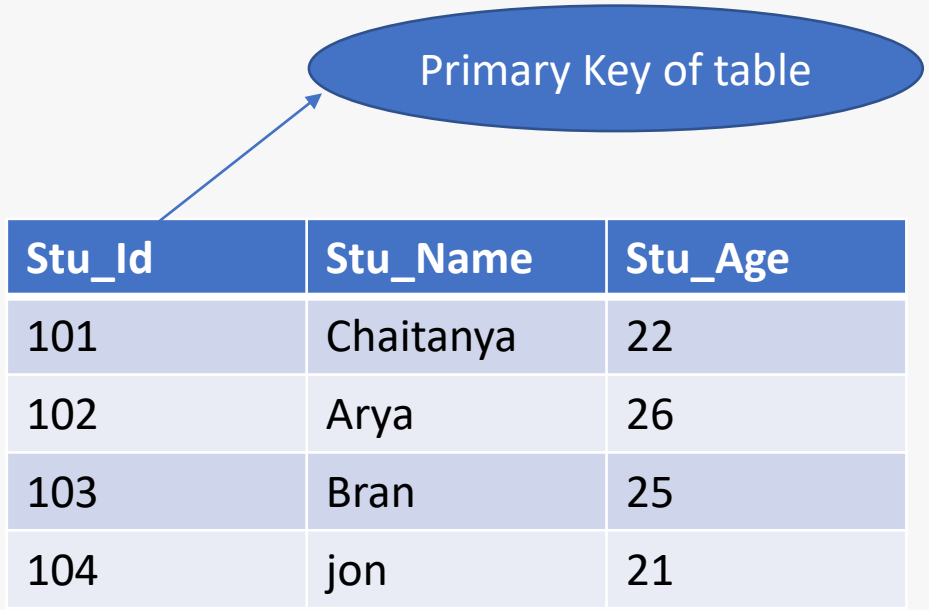
Add a foreign key constraint to an existing table

```
ALTER TABLE child_table
ADD CONSTRAINT constraint_name
FOREIGN KEY (fk_columns)
REFERENCES parent_table (parent_key_columns);
```

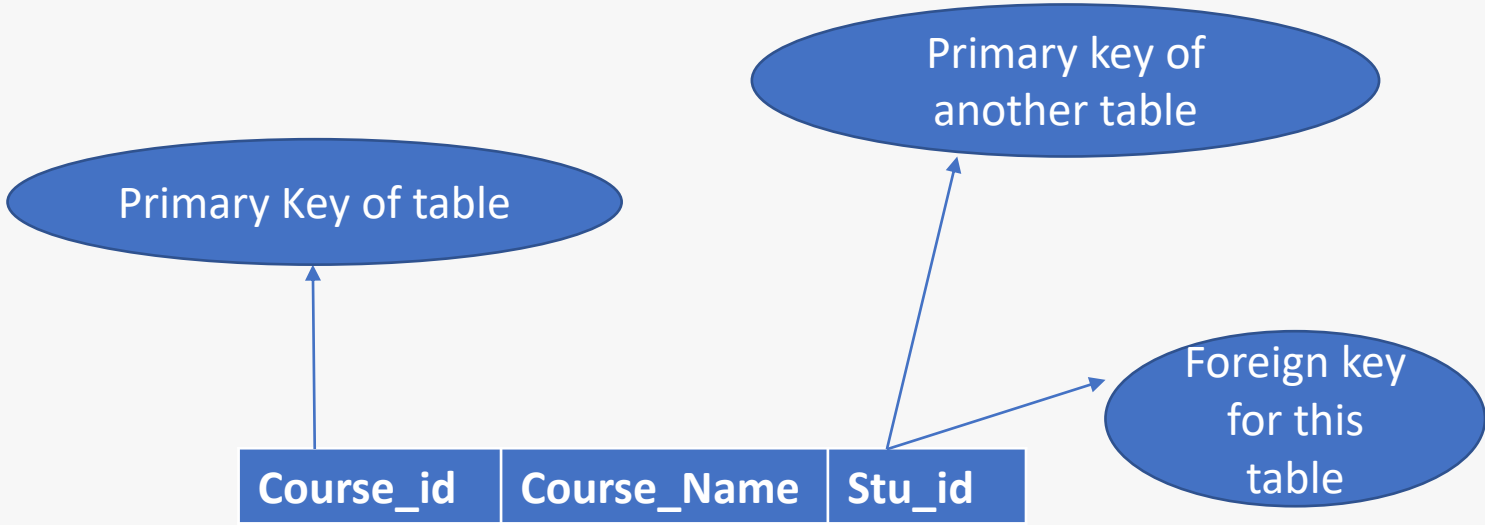
SYNTEX:

```
[CONSTRAINT fk_name]   FOREIGN KEY(fk_columns)  REFERENCES parent_table(parent_key_columns)
[ON DELETE delete_action]
[ON UPDATE update_action]
```

FOREIGN KEY



Stu_Id	Stu_Name	Stu_Age
101	Chaitanya	22
102	Arya	26
103	Bran	25
104	jon	21



Course_id	Course_Name	Stu_id
C01	xyz	101
C02	pqr	102
C03	abc	101
C04	stu	103


netlink
digital solutions
Thank you
www.netlink.com