SQL constraints

1). what are constraints refers in SQL?

↳ constraints in general refers to (rules)

↳ In point of SQL constraints are (rules), constraints Are set to the columns in the table.

↳ To ensure that the data entered into the Table meets specific criteria.

2) Types of constraints in mySQL ?

Types of Constraints in MySQL:

1. **NOT NULL**: Ensures that a column cannot store `NULL` values, meaning every row must have a value in this column.
   - Example: `name VARCHAR(50) NOT NULL`
2. **UNIQUE**: Guarantees that all values in a column are distinct, preventing duplicate entries.
   - Example: `email VARCHAR(100) UNIQUE`
3. **PRIMARY KEY**: Uniquely identifies each row in a table. A `PRIMARY KEY` is a combination of `NOT NULL` and `UNIQUE`, ensuring no `NULL` values and that all records are unique.
   - Example: `employee_id INT PRIMARY KEY`
4. **FOREIGN KEY**: Enforces a relationship between columns in different tables, ensuring referential integrity. It links a column to the `PRIMARY KEY` of another table.
   - Example: `department_id INT, FOREIGN KEY (department_id) REFERENCES departments(id)`
5. **CHECK**: Validates that the data in a column meets a specific condition (available from MySQL 8.0.16 onward).
   - Example: `age INT CHECK (age >= 18)`
6. **DEFAULT**: Assigns a default value to a column if no value is provided during data insertion.
   - Example: `status VARCHAR(10) DEFAULT 'active'`
7. **AUTO_INCREMENT**: Automatically generates a unique sequential number for a column, typically used for `PRIMARY KEY` columns.
   - Example: `employee_id INT AUTO_INCREMENT`

These constraints help maintain database accuracy by controlling the types of data that can be entered into each field.

3) Create a table insert some values, And the add constraints?

```
3 • create table employees(
4       emp_no int,
5       emp_name varchar(20),
6       salary int
7   );
8 • desc employees;
```

Field	Type	Null	Key	Default	Extra
emp_no	int	YES			
emp_name	varchar(20)	YES			
salary	int	YES			

We just created table without any constraints

```
37 • insert into employees values (1,"lavanya",100000),(2,"jaanu","100000");
38 • insert into employees(emp_name,salary) values("jack",20000);
39 • select * from employees;
```

emp_no	emp_name	salary
1	lavanya	100000
2	jaanu	100000
	jack	20000

See here our (emp-no) column Allows NULL since we didn't set Any constraints to avoid NULL

→ Explain (NOT NULL) constraint in SQL?

NOT NULL: Ensures that a column cannot store `NULL` values, meaning every row must have a value in this column.

→ How to set (not null) constraint for employees table which is already created?

```
38 • select * from employees;
```

emp_no	emp_name	salary
1	lavanya	100000
2	jaanu	100000
3	jack	20000

```
48 • desc employees;
```

Field	Type	Null
emp_no	int	YES
emp_name	varchar(20)	YES
salary	int	YES

Before setting NOT NULL constraint

```
44 • alter table employees
45   modify column emp_no int  not null;
46 • desc employees;
```

Field	Type	Null
emp_no	int	NO
emp_name	varchar(20)	YES
salary	int	YES

Explanation: This modifies the `emp_no` column in the `employees` table to ensure it cannot contain `NULL` values and must always have an integer value.

→ explain unique constraint ?

↳ unique constraint does not Allow the user to enter Duplicate values.

↳ which means if we provide a column with unique constraint, All the (row - value) for that column should be unique. (no duplicates are Allowed) for The column setted with unique constraint.

→ how to set unique constraint for a column in table which is already created?

```
105 • alter table employees2
106   add constraint unique(emp_no);
107 • desc employees2;
```

```
107 • desc employees2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key
emp_no	int	YES	
emp_name	varchar(20)	YES	
salary	int	YES	

Before Setting constraint

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key
emp_no	int	YES	UNI
emp_name	varchar(20)	YES	
salary	int	YES	

After Setting constraint

Explanation: This adds a `UNIQUE` constraint to the `emp_no` column in the `employees2` table, ensuring that every value in the `emp_no` column is unique, preventing any duplicate entries.

→ what is primary key constraint?

↳ primary key is the combination of both

↳ unique & ↳ NOT NULL constraints.

↳ so the column setted as the primary key should only allows the users to insert unique & NOT NULL values for each row of that column

↳ if suppose the user try to insert (duplicate) (or) NULL values means the constrain throws error.

→ How to set primary key for the column in a table which is already created ?

```
112 • desc employees2;
```

Field	Type	Null	Key
emp_no	int	YES	
emp_name	varchar(20)	YES	
salary	int	YES	

before Setting the Primary key constraint

```
110 • alter table employees2
111   add constraint primary key(emp_no);
112 • desc employees2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key
emp_no	int	NO	PRI
emp_name	varchar(20)	YES	
salary	int	YES	

After Setting the Primary key

Explanation: This adds a `PRIMARY KEY` constraint to the `emp_no` column in the `employees2` table. It ensures that the `emp_no` values are unique and cannot be `NULL`, making `emp_no` the unique identifier for each record in the table.

→ what is check constraint in SQL?

↳ if we set check constraint for a column

↳ we need to provide some criteria (or) condition inside the `check()` → constraint.

↳ so, while inserting (row-values) → for that column

↳ the condition inside `check()` → constraint will be checked, if the value to be insert satisfies the condition, then it will be inserted in the table (or)

↳ check constraint throws error.

→ let us create new table to demonstrate check constraint?

```
120 • create table rollercoaster(
121                                 id int,
122                                 name varchar(20),
123                                 age int);
124 • desc rollercoaster;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key
id	int	YES	
name	varchar(20)	YES	
age	int	YES	

↳ i have created a table rollercoaster
↳ which has 3 columns id, name, age.
↳ i want to check the values inserted for age column should Be greaterthan (or) equal to 18 And lesserthan (or) equal to 48

→ So if user try to insert value for age between (18 to 48) it will inserted into table (or) it will throw error.

→ how to add check constraint ?

```
128 • alter table rollercoaster
129   add constraint check( age >=18 and age<=48 );
130 • insert into rollercoaster values(1,"lakshith",48);
131 • select * from rollercoaster;
```

id	name	age
1	lakshith	48

This adds a CHECK constraint to the rollercoaster table, ensuring that the age column only accepts values between 18 and 48 (inclusive).

This inserts a new record with id=1, name='Lakshith', and age=48 into the rollercoaster table, which will succeed since 48 is within the allowed range.

```
131 • insert into rollercoaster values(2,"darshith",10);
```

Error Code: 3819. Check constraint 'rollercoaster_chk_1' is violated.

This will fail because the age value of 10 does not satisfy the CHECK constraint (age must be between 18 and 48).

↳ so it will throw error

→ Explain the concept of foreign key?

↳ foreign key constraint is used to map the primary key of Another tables.

↳ so this foreign key links with primary key of Another table.

↳ So, that we can retrieve the matched Data using (joins).

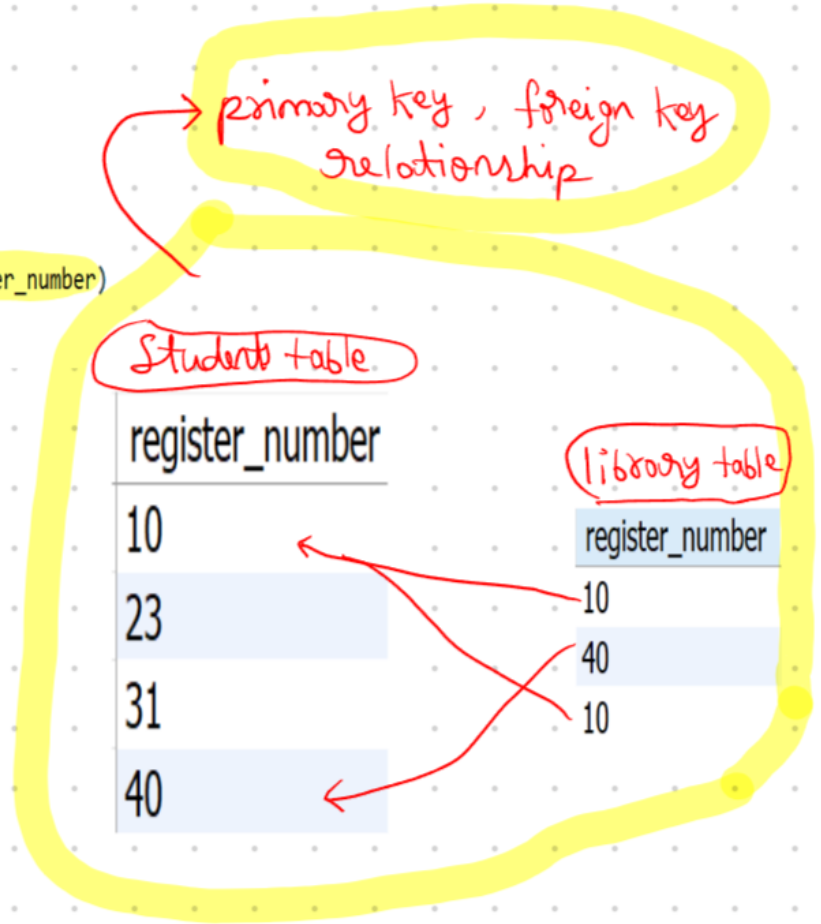
→ coding implementation for foreign key ?

```sql
create table students(
    register_number int primary key,
    student_name varchar(100)
);
```

register_number	student_name
10	jack
23	lannie
31	robin
40	bakey

```sql
create table library(
    entry int primary key,
    register_number int,
    book_name varchar(100),
    foreign key(register_number) references students(register_number)
);
```

entry	register_number	book_name
1	10	harry potter
2	40	cindrella
3	10	thor

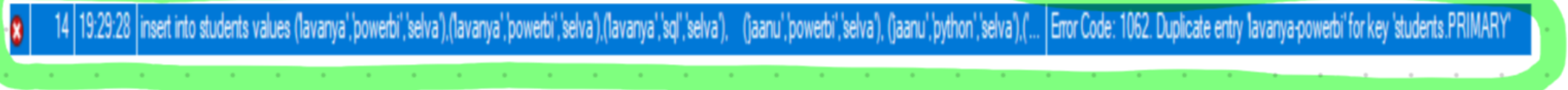


→ Explain the concept of composite key ?

⊙ when two columns in a table is assigned as {primary key} then it is called as (composite key).

⊙ where the (first column) Along with (second column) should be unique.

⊙ when you try to insert duplicate records it will (raise error)



→ practical implementation for composite key ?

```
3  create table students(
4      sname varchar(30) ,
5      course varchar(30) ,
6      trainer varchar(30),
7      primary key(sname, course));
```

```
11 insert into students
12 values ('lavanya','powerbi','selva'),('lavanya','python','selva'),('lavanya','sql','selva'),
13        ('jaanu','powerbi','selva'), ('jaanu','python','selva'),('jaanu','sql','selva'),
14        ('keerthana','powerbi','selva'),('barath','powerbi','selva');
15
16 select * from students;
17
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

sname	course	trainer
barath	powerbi	selva
jaanu	powerbi	selva
jaanu	python	selva
jaanu	sql	selva
keerthana	powerbi	selva
lavanya	powerbi	selva
lavanya	python	selva
lavanya	sql	selva