Foreign Key

- Foreign key is used to establish relationship between tables.
- Foreign key is a referential integrity constraint.

Case#1

Master Table

```
create table course(
        course_name varchar(30) primary key,
        course_fee int not null
);
```

Child Table

```
create table student(
        regdno int primary key,
        name varchar(30)  not null,
        course_name varchar(30),
        foreign key(course_name) references course(course_name)
);
```

Conclusion

- One table foreign key must belong to another table primary key.

Case#2

```
create table course(
        course_name varchar(30) primary key,
        course_fee int
);
create table student(
        regdno int primary key,
        name varchar(30)  not null,
        course_fee int,
        foreign key(course_fee) references course2(course_fee)
);
```

Conclusion

- One table foreign key must belong to another table primary key or unique key.

```
        create table course2(
        course_name varchar(30) primary key,
        course_fee int unique
    );
```

Case#3

```
create table course(
    course_name int primary key,
    course_fee int
);

create table student(
    regdno int primary key,
    name varchar(30)  not null,
    course_name varchar(20),
    foreign key(course_name) references course3(course_name)
);
```

Conclusion

- Primary key and foreign key must belongs to same data type.

Case#4

```
insert into student values(101,'Alok','alok','CTC',9090,'java');
```

Conclusion

- We are not allowed to insert values in child table other than primary key value in master table.

Case#5

```
insert into student values(101,'Ram','java');

insert into student values(102,'Raj','java');
```

Conclusion

- Foreign accept duplicate value.

Case#6

```
insert into student values(103,'Raj',null);
```

Conclusion

- Foreign accept null value.

Deleting Records from Parent Table

```
DELETE FROM course WHERE course_name='mysql';
```

Conclusion

- We can’t able delete record from parent table directly, without deleting child record.

Solutions

Approach1

- First delete record from child table and after that delete record from parent table.

Approach2

- Set related foreign key value to null and then delete record from parent table.

Approach3 - on delete cascade

- Whenever we are using this clause in child table, if you are deleting a master table record automatically related child table records are deleted.

```
create table course(
        course_name varchar(30) primary key,
        course_fee int not null
);

create table student(
        regdno int primary key,
        name varchar(30)  not null,
        course_name varchar(30),
  foreign key(course_name) references course(course_name) on delete cascade
);

DELETE FROM course WHERE course_name='mysql';
```

Approach4 – on delete set null

- Whenever we are using this clause in child table, if you are deleting a master table record automatically related child table records are set to null.

```
    create table course(
        course_name varchar(30) primary key,
        course_fee int not null
    );

    create table student(
        regdno int primary key,
        name varchar(30)  not null,
        course_name varchar(30),
      foreign key(course_name) references course(course_name) on delete set null
    );
    DELETE FROM course WHERE course_name='mysql';
```

Working with Composite Primary Key

```
create table course(
    course_id int,
    course_name varchar(30),
    course_fee int not null,
    primary key(course_id, course_name)
);

create table student(
    regdno int primary key,
    name varchar(30)  not null,
    course_id int,
    course_name varchar(30),
    foreign key(course_id, course_name) references course(course_id, course_name)
);
```

```
INSERT into course VALUES(1,'mysql',5000);

INSERT into student values(101,'Ram',1,'mysql');
```

Add Foreign Key Constraint on Existing Column

```
ALTER TABLE student ADD CONSTRAINT foreign key(course_name) references
course(course_name);
```

```
SELECT constraint_name FROM information_schema.KEY_COLUMN_USAGE WHERE
TABLE_SCHEMA = 'testdb' AND TABLE_NAME = 'student';
```

Drop Foreign Constraint

```
ALTER TABLE student DROP FOREIGN KEY student_ibfk_1;
```

Change Constraint Name

```
create table student(
        regdno int primary key,
        name varchar(30)  not null,
        course_name varchar(30),
        CONSTRAINT fk_new_course_id foreign key(course_name) references
course(course_name)
);
```

Check Constraints

- This constraint is used to define logical conditions according to our business rule.
- Syntax:

  ```
  create table tablename(
    column1 datatype(size) check logical condition,
    column2 datatype(size)
  );
  ```

Example#1

```
create table student(

        age int check (age between 10 and 20)

);

INSERT INTO student VALUES(5);

INSERT INTO student VALUES(15);

INSERT INTO student VALUES(25);
```

Example#2

```
create table student(
	mobno bigint check(length(mobno) = 10)
);
INSERT INTO student VALUES(123456789);
INSERT INTO student VALUES(12345678910);
INSERT INTO student VALUES(1234567891);
```

Example#3

```
create table student(
	course varchar(30) check(course in('Data Engineering','DevOps'))
);
INSERT INTO student VALUES('Data Science');
INSERT INTO student VALUES('devops');
INSERT INTO student VALUES('Data Engineering');
```

Example#4

```
create table student(
	regd_date date check(regd_date between '2024-01-31' and '2024-02-20')
);
INSERT INTO student VALUES('2024-01-30');
INSERT INTO student VALUES('2024-01-31');
INSERT INTO student VALUES('2024-02-10');
INSERT INTO student VALUES('2024-02-20');
INSERT INTO student VALUES('2024-02-21');
```

Add Constraint on Existing Column

Syntax

```
ALTER TABLE table_name ADD CONSTRAINT constraint_name CHECK (condition);
```

Example

```
create table student(

        regd_date date

);

SELECT constraint_name FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA = 'testdb' AND TABLE_NAME = 'student';

SHOW CREATE TABLE student;

ALTER TABLE student ADD CONSTRAINT regd_date_check_ct CHECK (regd_date between '2024-01-31' and '2024-02-25');
```

Drop Constraint

```
ALTER TABLE student DROP CONSTRAINT regd_date_check_ct;
```

Default value for table column

- The DEFAULT clause is used to insert a default value into a column.

Example

```
create table student(

        name varchar(30),

        address varchar(30) default 'bbsr'

);


INSERT INTO student values('Ram');

INSERT INTO student(name) values('Ram');

INSERT INTO  student values('Alok','CTC');
INSERT INTO  student values('Raj',null);
```

Questions

1. What is constraint?
2. What are the different categories of constraint?
3. What is not null constraint?
4. What is unique constraint?
5. What is primary key constraint?
6. What is check constraint?
7. What is foreign key constraint?
8. What is the difference between not null and unique key?
9. What is the difference between primary key and unique key?
10. Can you add constraints to a table that already has data?
11. How many primary keys can exists on a table?
12. Write query to create composite primary key?
13. Write query to create primary key without using primary key clause.
14. What is default clause? How it works?
15. How to add or drop constraint?

Assignments

1. Create customers table with customer_id, username, email, phone, address. Declare customer_id as primary key.
2. Create orders table with order_id, order_date, amount_paid, customer_id. Declare order_id as primary key and customer_id as foreign key. Here customers is master table and orders is child table.
3. Create products table with product_id, product_name, price, description columns. Declare product_id as primary key.
4. Create orders_product table with id, order_id, product_id, quanity columns. Declare order_id and product_id as foreign key and id as primary key.