# Agenda

```
Foreign Key
CHECK
Alter Table
Stored Procedure
Triggers
```

## Foreign Key

```
CREATE TABLE students(
    std INT,
    rollno INT,
    name CHAR(10),
    PRIMARY KEY(std,rollno));

CREATE TABLE marks(
    subject CHAR(15),
    marks DECIMAL(5,2),
    std INT,
    rollno INT,
    Foreign Key (std,rollno) REFERENCES students(std,rollno)
);

INSERT INTO  students VALUES(1,1,"abc");
INSERT INTO  students VALUES(1,2,"lmn");
INSERT INTO  students VALUES(2,1,"pqr");


INSERT INTO marks VALUES("Maths",70,1,1);
INSERT INTO marks VALUES("Maths",60,1,2);
INSERT INTO marks VALUES("Maths",40,2,1);

INSERT INTO marks VALUES("History",50,1,1);
INSERT INTO marks VALUES("History",90,1,2);
INSERT INTO marks VALUES("History",60,2,1);

--display report card of 1st standard students
SELECT s.std,s.rollno,s.name,m.marks FROM students s INNER JOIN marks m ON
s.rollno=m.rollno;

SELECT s.std,s.rollno,s.name,m.marks FROM students s INNER JOIN marks m ON
s.rollno=m.rollno AND s.std=m.std WHERE m.std=1;

CREATE TABLE emps(
    empno INT PRIMARY KEY,
    ename CHAR(10),
```

```
    mgr INT,
    deptno INT,
    FOREIGN KEY (deptno) REFERENCES depts(deptno),
    FOREIGN KEY (mgr) REFERENCES emps(empno)
);

INSERT INTO emps VALUES(1,"Amit",4,10);
--error

SELECT @@foreign_key_checks;

SET @@foreign_key_checks=0;

SELECT @@foreign_key_checks;

INSERT INTO emps VALUES(1,"Amit",4,10); -- OK

SET @@foreign_key_checks=1;

INSERT INTO emps VALUES(2,"Rahul",3,10);
--error
```

## CHECK

```
CREATE TABLE employee(
    empid INT PRIMARY KEY,
    ename CHAR(10) CHECK (LENGTH(ename)>1),
    age INT CHECK (age>18),
    sal DECIMAL(10,2) CHECK (sal>1000)
);

INSERT INTO employee VALUES(1,"a",16,800);
--error

INSERT INTO employee VALUES(1,"rohan",16,800);
--error

INSERT INTO employee VALUES(1,"rohan",20,800);
--error

INSERT INTO employee VALUES(1,"rohan",20,1800);
```

```
SHOW CREATE TABLE emps;

DROP TABLE emps;

CREATE TABLE emps(
    empno INT,
    ename CHAR(10),
```

```
    mgr INT,
    deptno INT,
    CONSTRAINT `pk_empno` PRIMARY KEY(empno),
    CONSTRAINT `fk_deptno` FOREIGN KEY (deptno) REFERENCES depts(deptno),
    CONSTRAINT `fk_mgr` FOREIGN KEY (mgr) REFERENCES emps(empno)
);
```

## Alter

```
CREATE TABLE mobiles(
    mid INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(20),
    price DECIMAL (10,2)
);

INSERT INTO mobiles(name,price) VALUES("Xiomi",10000);
INSERT INTO mobiles(name,price) VALUES("Samsung",20000);

SELECT * FROM mobiles;

ALTER TABLE mobiles AUTO_INCREMENT = 100;

INSERT INTO mobiles(name,price) VALUES("Vivo",15000);

-- ADD a column description in mobiles table
ALTER TABLE mobiles ADD COLUMN description CHAR(20);

UPDATE mobiles SET description = "Best" WHERE mid = 1;
UPDATE mobiles SET description = "Better" WHERE mid = 2;
UPDATE mobiles SET description = "Good" WHERE mid = 100;

-- Change the datatype from char to varchar of description
ALTER TABLE mobiles MODIFY COLUMN description VARCHAR(20);

-- Change the col name of  description to desci
ALTER TABLE mobiles CHANGE COLUMN description desci VARCHAR(20);

-- DELETE the coloumn desci
ALTER TABLE mobiles DROP COLUMN desci;

-- ADD a unique constraint on name
ALTER TABLE mobiles ADD CONSTRAINT UNIQUE(name);

SHOW CREATE TABLE mobiles;

-- REMOVE THE CONSTARINTS
ALTER TABLE mobiles DROP PRIMARY KEY;
ALTER TABLE mobiles DROP CONSTRAINT name;
```

## PSM / PL SQL

```
- Persistant Stored modules
- Procedural Language
```

```
CREATE TABLE result (id INT, value CHAR(100));

SHOW PROCEDURE STATUS WHERE db="classwork_db";

SELECT @res;

SET @res = 10;

SELECT @res;
```

## Trigger

```
- BEFORE INSERT
- AFTER INSERT
- BEFORE UPDATE
- AFTER UPDATE
- BEFORE DELETE
- AFTER DELETE

INSERT - NEW
UPDATE - NEW & OLD
DELETE - OLD
```

```
CREATE TABLE accounts(accno INT, acc_type CHAR(10),balance DECIMAL(10,2));

INSERT INTO accounts VALUES(1,"SAVINGS",10000);
INSERT INTO accounts VALUES(2,"SAVINGS",20000);
INSERT INTO accounts VALUES(3,"CURRENT",30000);
INSERT INTO accounts VALUES(4,"SAVINGS",40000);


CREATE TABLE transactions (tid INT PRIMARY KEY AUTO_INCREMENT, tx_type CHAR(10),
amount DECIMAL(10,2),accno INT);

INSERT INTO transactions(tx_type,amount,accno) VALUES("credit",5000,1);
INSERT INTO transactions(tx_type,amount,accno) VALUES("debit",18000,1);
INSERT INTO transactions(tx_type,amount,accno) VALUES("debit",15000,2);
INSERT INTO transactions(tx_type,amount,accno) VALUES("debit",20000,3);
INSERT INTO transactions(tx_type,amount,accno) VALUES("credit",5000,1);
```