Agenda

```
- Indexes
- Constraints
- Alter Table
```

Indexes

```
- For Faster Searching
```

```
SELECT * FROM books;
DESC books;

-- display all books of c programming
EXPLAIN FORMAT = JSON SELECT * FROM books WHERE subject = "C Programming";
--1.55

CREATE INDEX idx_books_subject ON books(subject);

EXPLAIN FORMAT = JSON SELECT * FROM books WHERE subject = "C Programming";
--0.90

SHOW INDEXES FROM books;

-- display all books of c of author Yashwant Kanetkar
EXPLAIN FORMAT = JSON SELECT * FROM books WHERE author = "Yashwant Kanetkar";
--1.55

CREATE INDEX idx_books_author ON books(author DESC);

EXPLAIN FORMAT = JSON SELECT * FROM books WHERE author = "Yashwant Kanetkar";
--0.70

--display ename,deptname from emps;
EXPLAIN FORMAT = JSON SELECT e.ename,d.dname FROM emps e INNER JOIN depts d ON
e.deptno=d.deptno;
--2.90

CREATE INDEX idx_emps_deptno ON emps(deptno);
CREATE INDEX idx_depts_deptno ON depts(deptno);

EXPLAIN FORMAT = JSON SELECT e.ename,d.dname FROM emps e INNER JOIN depts d ON
e.deptno=d.deptno;
-- 2.50
```

UNIQUE INDEX

```
SELECT * FROM emps;

DESC emps;

CREATE UNIQUE INDEX idx_emps_ename ON emps(ename);

DESC emps;

SHOW INDEXES FROM emps;

INSERT INTO emps VALUE(6,"Rahul","20",4);
--Duplicate entry 'Rahul' for key 'emps.idx_emps_ename'

INSERT INTO emps VALUE(6,"Vishal","20",4);

INSERT INTO emps(empno,deptno) VALUES(7,30);
-- It will insert NULL in ename

INSERT INTO emps(empno,deptno) VALUES(8,40);
-- Multiple NULL Values are allowed for unique Index

CREATE UNIQUE INDEX idx_emps_mgr ON emps(mgr);
-- Unique index cannot be created on cols having duplicate data
```

COMPOSITE INDEX

```
SELECT * FROM emp;

-- display empno,ename,sal where job = analyst and deptno=20;
EXPLAIN FORMAT = JSON SELECT empno,ename,sal FROM emp WHERE deptno=20  AND job =
"ANALYST";

-- display empno,ename,sal havig job = "analyst" OR deptno = 30;
EXPLAIN FORMAT = JSON SELECT empno,ename,sal FROM emp WHERE deptno=30  OR job =
"ANALYST";

CREATE INDEX idx_emp_dj ON emp(deptno,job);
CREATE INDEX idx_emp_dj ON emp(job,deptno);
--CREATE INDEX idx_emp_dj ON emp(deptno,job DESC);
--CREATE INDEX idx_emp_dj ON emp(deptno DESC,job DESC);

DESC emp;

SHOW INDEXES FROM emp;
EXPLAIN FORMAT = JSON SELECT empno,ename,sal FROM emp WHERE deptno=30;
--1.10

EXPLAIN FORMAT = JSON SELECT empno,ename,sal FROM emp WHERE job = "ANALYST";
```

```
--1.65

EXPLAIN FORMAT = JSON SELECT empno,ename,sal FROM emp WHERE sal = 3000;
--1.65

CREATE INDEX idx_emp_job ON emp(job);
CREATE INDEX idx_emp_deptno ON emp(deptno);

EXPLAIN FORMAT = JSON SELECT empno,ename,sal FROM emp WHERE job = "ANALYST";
--0.70

EXPLAIN FORMAT = JSON SELECT empno,ename,sal FROM emp WHERE deptno=30;
--1.10

CREATE TABLE students(
    std INT,
    rollno INT,
    name VARCHAR(20),
    marks DECIMAL(10,2));

INSERT INTO students VALUES(1,1,"SOHAM",35);
INSERT INTO students VALUES(1,2,"MOHAN",45);
INSERT INTO students VALUES(2,1,"ROHAN",55);
INSERT INTO students VALUES(2,2,"MAHESH",65);

SELECT * FROM students;

CREATE UNIQUE INDEX idx_st_sr ON students(std,rollno);

INSERT INTO students VALUES(2,2,"SURESH",75);
--error
INSERT INTO students VALUES(1,2,"SURESH",75);
--error

INSERT INTO students VALUES(1,3,"SURESH",75);
```

CLUSTERED INDEX

```
- It is automatically crated on Primary Key.
```

DROP INDEX

```
DESC books;

SHOW INDEXES FROM books;

DROP INDEX idx_books_subject ON books;

SHOW INDEXES FROM books;
```

```
DROP INDEX idx_books_author ON books;
```

Constraints

```
1. NOT NULL
2. UNIQUE
3. PRIMARY KEY
4. FOREIGN KEY
5. CHECK
```

```
--Column Level Constraint
CREATE TABLE temp(
    col1 INT PRIMARY KEY,
    col2 INT NOT NULL,
    col3 VARCHAR(20) UNIQUE,
    col4 VARCHAR(30)
);

--Table Level Constraint
CREATE TABLE temp(
    col1 INT,
    col2 INT NOT NULL,
    col3 VARCHAR(20),
    col4 VARCHAR(30),
    UNIQUE(col2),
    PRIMARY KEY(col1)
);
```

NOT NULL

```
- Null values cannot be given in these coloumns
- It can be given only on column level
```

```
    CREATE TABLE temp1 (c1 INT, c2 INT, c3 INT NOT NULL);
    DESC temp1;

    INSERT INTO temp1 VALUES(1,1,1);
    INSERT INTO temp1 VALUES(NULL,2,2);
    INSERT INTO temp1 VALUES(3,NULL,2);

    INSERT INTO temp1 VALUES(4,4,NULL);
    --error
```

```
    INSERT INTO temp1(c1,c2) VALUES(4,4);

    SHOW INDEXES FROM temp1;
```

UNIQUE

```
- Duplicate values are not allowed in the column
- However Multiple NULL values are allowed
- It internally creates a UNIQUE INDEX
```

```
CREATE TABLE temp2(c1 INT, c2 INT UNIQUE);

DESC temp2;

SHOW INDEXES FROM temp2;

INSERT INTO temp2 VALUES(1,1);
INSERT INTO temp2 VALUES(2,2);
INSERT INTO temp2 VALUES(3,3);
INSERT INTO temp2 VALUES(4,4);

INSERT INTO temp2 VALUES(3,5);

INSERT INTO temp2 VALUES(4,5);
--error Duplicate entry '5' for key 'temp2.c2'

INSERT INTO temp2 VALUES(5,NULL);
INSERT INTO temp2 VALUES(6,NULL);

DROP TABLE students;

CREATE TABLE students(
    std INT,
    rollno INT,
    name VARCHAR(20),
    marks DECIMAL(10,2),
    UNIQUE (std,rollno));

INSERT INTO students VALUES(1,1,"SOHAM",35);
INSERT INTO students VALUES(1,2,"MOHAN",45);
INSERT INTO students VALUES(2,1,"ROHAN",55);
INSERT INTO students VALUES(2,2,"MAHESH",65);

SELECT * FROM students;

DESC students;

SHOW INDEXES FROM students;
```

```
INSERT INTO students VALUES(2,2,"SURESH",75);
--error
INSERT INTO students VALUES(1,2,"SURESH",75);
--error

INSERT INTO students VALUES(1,3,"SURESH",75);

SELECT * FROM students;
```

Primary Key

```
primary key = unique + not null
```

```
CREATE TABLE cdac_student(
    prn CHAR(12) PRIMARY KEY,
    name VARCHAR(30)  NOT NULL,
    email VARCHAR(50) UNIQUE NOT NULL,
    mobile VARCHAR(12) UNIQUE NOT NULL,
    addr VARCHAR(80)
);

DESC cdac_student;

SHOW INDEXES FROM cdac_student;

INSERT INTO cdac_student
VALUES(1001,"rohan","rohan@gmail.com","8983049388","pune");

INSERT INTO cdac_student
VALUES(1001,"pradnya","pradnya@gmail.com","1234567890","pune");
--error
INSERT INTO cdac_student
VALUES(NULL,"pradnya","pradnya@gmail.com","1234567890","pune");
--error
INSERT INTO cdac_student
VALUES(1002,NULL,"pradnya@gmail.com","1234567890","pune");
--error
INSERT INTO cdac_student
VALUES(1002,NULL,"pradnya@gmail.com","8983049388","pune");
--error

INSERT INTO cdac_student
VALUES(1002,"pradnya","pradnya@gmail.com","1234567890","pune");
```

```
DROP TABLE students;

CREATE TABLE students(
```

```
    std INT,
    rollno INT ,
    name VARCHAR(20),
    marks DECIMAL(10,2),
    PRIMARY KEY (std,rollno));

DESC students;

SHOW INDEXES FROM students;

INSERT INTO students VALUES(1,1,"SOHAM",35);
INSERT INTO students VALUES(1,2,"MOHAN",45);
INSERT INTO students VALUES(2,1,"ROHAN",55);
INSERT INTO students VALUES(2,2,"MAHESH",65);

INSERT INTO students VALUES(2,2,"SURESH",75);
--error
INSERT INTO students VALUES(1,2,"SURESH",75);
--error

INSERT INTO students VALUES(NULL,NULL,"SURESH",75);
--error

INSERT INTO students VALUES(1,NULL,"SURESH",75);
--error

INSERT INTO students VALUES(1,3,"SURESH",75);

SELECT * FROM students;
```

Surrogate Primary Key

```
CREATE TABLE products(
    pid INT AUTO_INCREMENT,
    pname VARCHAR(20),
    price DECIMAL(10,2),
    PRIMARY KEY(pid) );

INSERT INTO products(pname,price) VALUES("Redmi",100000);
INSERT INTO products(pname,price) VALUES("samsung",200000);

INSERT INTO products(pname,price) VALUES("Bread",35);
INSERT INTO products(pname,price) VALUES("Milk",70);

INSERT INTO products(pname,price) VALUES("Odomus",85);

INSERT INTO products VALUES(22,"Maggie",14);

INSERT INTO products(pname,price) VALUES("Lays",10);
```

Foriegn Key

```
DROP TABLE emps;
DROP TABLE depts;

CREATE TABLE depts(
    deptno INT PRIMARY KEY,
    dname CHAR(5)
);

INSERT INTO depts VALUES(10,"DEV");
INSERT INTO depts VALUES(20,"QA");
INSERT INTO depts VALUES(30,"OPS");
INSERT INTO depts VALUES(40,"ACC");

CREATE TABLE emps(
    empno INT PRIMARY KEY,
    ename CHAR(10),
    mgr INT,
    deptno INT,
    FOREIGN KEY (deptno) REFERENCES depts(deptno)
);

DESC emps;
SHOW INDEXES FROM emps;

INSERT INTO emps VALUES (1,"Amit",4,10);
INSERT INTO emps VALUES (2,"Rahul",3,10);
INSERT INTO emps VALUES (3,"Nilesh",4,20);

INSERT INTO emps VALUES (4,"Nitin",5,50);
--error FK constraint Failed

INSERT INTO emps VALUES (4,"Nitin",5,40);

INSERT INTO emps VALUES (5,"Sarang",NULL,NULL);

DELETE FROM depts WHERE deptno = 30;
DELETE FROM depts WHERE deptno = 40;
--error FK constraint Failed

DELETE FROM emps WHERE deptno = 40;
DELETE FROM depts WHERE deptno = 40;
```

```
DROP TABLE emps;
DROP TABLE depts;

CREATE TABLE depts(
    deptno INT PRIMARY KEY,
    dname CHAR(5)
```

```
);

INSERT INTO depts VALUES(10,"DEV");
INSERT INTO depts VALUES(20,"QA");
INSERT INTO depts VALUES(30,"OPS");
INSERT INTO depts VALUES(40,"ACC");

CREATE TABLE emps(
    empno INT PRIMARY KEY,
    ename CHAR(10),
    mgr INT,
    deptno INT,
    FOREIGN KEY (deptno) REFERENCES depts(deptno) ON DELETE CASCADE ON UPDATE
CASCADE
);


INSERT INTO emps VALUES (1,"Amit",4,10);
INSERT INTO emps VALUES (2,"Rahul",3,10);
INSERT INTO emps VALUES (3,"Nilesh",4,20);
INSERT INTO emps VALUES (4,"Nitin",5,40);
INSERT INTO emps VALUES (5,"Sarang",NULL,30);

UPDATE depts SET deptno = 50 WHERE deptno = 40;
SELECT * FROM depts;
SELECT * FROM emps;

DELETE FROM depts WHERE deptno = 30;
SELECT * FROM depts;
SELECT * FROM emps;
```