• • •
<pre> Creating table in database CREATE TABLE student (    id INT PRIMARY KEY,    name VARCHAR(50),    age INT NOT NULL</pre>
Inserting values in a table/schema INSERT INTO student VALUES(1, "Kanad", 23); INSERT INTO student VALUES(2, "Sudip", 25);
<pre> Show all data from a table SELECT * FROM student; /* Types of SQL commands:</pre>
DDL (Data Definition Language): create, alter, rename, truncate and drop  DQL (Data Query Language): select  DML (Data Manipulation Language): insert, update, delete
<pre>DCL (Data Control Language): grant and revoke permissions to users TCL (Transaction Control Language): start transaction, commit, rollback (advanced) */ /* Database Related Queries:</pre>
CREATE DATABASE db_name; CREATE DATABASE IF NOT EXISTS db_name;  DROP DATABASE db_name;
DROP DATABASE IF EXISTS db_name;  SHOW DATABASES; SHOW TABLES; */
CREATE DATABASE IF NOT EXISTS college; DROP DATABASE IF EXISTS company; SHOW DATABASES; SHOW TABLES;
/* Table Related Queries:
<pre>CREATE TABLE table_name (     column_name1 dataype constraint,     column_name2 dataype constraint, ); */</pre>
<pre>CREATE TABLE student(    id INT PRIMARY KEY, not null and unique    name VARCHAR(50) );</pre>
2. Select and view all columns SELECT * FROM table_name;  SELECT * FROM student;  /*
3. Insert INSERT INTO table_name(column1, column2) VALUES (col1_val1, col2_val1), (col1_val2, col2_val2);
*/ For multiple value we run this query INSERT INTO student (id, name) VALUES (1, "Kanad"),
<pre>(2, "Sudip"), (3, "Koushik");  For single value INSERT INTO student VALUES(4, "Soumya");</pre>
/*
employee info(id, name, salary) -> Add following information into the DB: 1, "adam", 25000 2, "bob", 30000 3, "casey", 40000
-> Select and view all your table data
CREATE TABLE employees_info(    id INT PRIMARY KEY,    name VARCHAR(100),    salary INT );
Ans 2: inerting data into that table INSERT INTO employees_info (id, name, salary) VALUES (1, "adam", 25000), (2, "bob", 30000),
(3, "casey", 40000);  Ans 3: showing data from a table SELECT * FROM employees_info;
/* KEYS: 1. Primary Key:
CONSTRAINTS  1. NOT NULL (columns cannot have a null value) col1 int NOT NULL  2. UNIQUE (all values in column are different) col2 int UNIQUE
3. PRIMARY KEY ()  */  USE college;  CREATE TABLE temp1(
<pre>id INT UNIQUE );  INSERT INTO temp1 VALUES(101); this will run fine INSERT INTO temp1 VALUES(101); but here this will give error as duplicate entry 101.</pre>
<pre>SELECT * FROM temp1;  CREATE TABLE temp2(    id INT UNIQUE,    name VARCHAR(50) NOT NULL,</pre>
age INT );  INSERT INTO temp2 VALUES(1, "Kanad", 23); INSERT INTO temp2(id, age) VALUES(2, 25); error: field 'name' doesn't have a default value.
<pre>INSERT INTO temp2(id, name) VALUES(2, "Sudip"); this will run fine as age is not necessary. /* -&gt; Ways to define Primary Key: */</pre>
First syntax: CREATE TABLE temp3(    id INT PRIMARY KEY,    name VARCHAR(50),    age INT,
city VARCHAR(30) );  Second syntax: CREATE TABLE temp3(
<pre>id INT, name VARCHAR(50), age INT, city VARCHAR(30), PRIMARY KEY (id) );</pre>
Third syntax: (combination as primary key)  CREATE TABLE temp3(    id INT,
<pre>name VARCHAR(50), age INT, city VARCHAR(30), PRIMARY KEY (id, name) id or name can be duplciate but (id, name) combination can't );</pre>
/* FOREIGN KEY> Prevent action that would detroy links between tables */
<pre>CREATE TABLE customer (    id INT PRIMARY KEY,    name VARCHAR(50), ); CREATE TABLE buying_items(</pre>
<pre>cust_id int, FOREIGN KEY (cust_id) REFERENCES customer(id) );  /* DEFAULT</pre>
-> sets the default value of a column = salary INT DEFAULT 20000 */  CREATE TABLE emp (
<pre>id INT,     salary INT DEFAULT 25000 );  INSERT INTO emp(id) VALUES(2154); here salary is not mandatory SELECT ** FROM emp: this will show one row with 1 2154 1 25000 1</pre>
SELECT * FROM emp; this will show one row with   2154   25000    /* CHECK> It can limit the values allowed in a column */
<pre>CREATE TABLE newTab (    age INT CHECK (age &gt;= 18) );</pre>
INSERT INTO newtab VALUES(17); this shows check constraint is violated INSERT INTO newtab VALUES(18); this one will run fine  SELECT * FROM newtab;
<pre> One more way to use CHECK: CREATE TABLE city (    id INT PRIMARY KEY,    city VARCHAR(50),    age INT,    CONSTRAINT age_check CHECK (age &gt;= 18 AND city="Delhi")</pre>
<pre>INSERT INTO city VALUES(1, "Mumbai", 21); age_check violated INSERT INTO city VALUES(1, "Delhi", 17); age_check violated INSERT INTO city VALUES(1, "delhi", 27); runs fine</pre>
INSERT INTO city VALUES(3, "DELHI", 23); runs fine INSERT INTO city VALUES(4, "dElHI", 23); runs fine INSERT INTO city VALUES(4, "dElHI", 23); runs fine
CREATE TABLE student (    rollNo INT PRIMARY KEY,    name VARCHAR(50),    marks INT NOT NULL,
<pre>grade VARCHAR(1),   city VARCHAR(20) );  INSERT INTO student (rollNo, name, marks, grade, city)</pre>
VALUES (101, "anil", 78, "C", "Pune"), (102, "bhumika", 93, "A", "Mumbai"), (103, "chetan", 85, "B", "Mumbai"), (104, "dhruv", 96, "A", "Delhi"),
<pre>(105, "emanuel", 12, "F", "Delhi"), (106, "farah", 82, "B", "Delhi"); SELECT * FROM student;</pre>
/* SELECT in detail */  Basic: SELECT col1, col2 FROM table_name; SELECT name, marks FROM student; this will show name and marks in table
SELECT city FROM student; show all cities SELECT DISTINCT city FROM student; shows all distinct city (for this: Pune, Mumbai, Delhi)  /*
-> to define some conditions SYNTAX: SELECT col1, col2 FROM table_name WHERE condition; */
<pre>SELECT * FROM student WHERE marks &gt; 80; showing 4 rows SELECT * FROM student WHERE city = "Mumbai"; showing 2 results SELECT * FROM student WHERE (city = "Delhi" AND grade = "A"); showing one result</pre>