

```

# Session 1
/*
1.  SQL Create Databases
    - show the databases
2.  SQL Drop Databases
3.  SQL Create Table
    - Show the table
4.  SQL insert into table
    - single values
    - Multiple values
5.  Retrive the data from the tables
6.  SQL Drop Table
*/

#check the databases
show databases;

create database emp;

show databases;

drop database emp;

create table emp_info(
emp_ID integer,
emp_name Varchar(250),
emp_surname varchar(250));

show tables;

insert into emp_info(emp_ID,emp_name,emp_surname)
values (1,'Dinesh','top');

insert into emp_info(emp_ID,emp_name,emp_surname)
values
(1,'Dinesh','top'),
(2,'Vamsi','Bottom'),
(3,'Reddy','Right');

select * from emp_info;

drop table emp_info;

show tables;

#Assessment

```

```
show databases;
use persons;
```

```
create table person_info(
id integer,
first_name varchar(250),
Last_name varchar(250));
```

```
show tables;
```

```
select * from person_info;
```

```
# inserting the multiple values into tables
insert into person_info(id,first_name,last_name)
values
(1,'vamsi','reddy'),
(2, 'Sam' , 'reddy'),
(3, 'Vijay', 'Kumar');
```

```
select * from person_info;
```

```
drop table person_info;
```

```
show tables;
```

```
# Session 2
/*
SQL Null Values
SQL update Statement
SQL Delete Statement
SQL ALter TAbLe
---- Add Column in Existing Column
---- Alter Table - Drop Column

*/
```

```
create database school;
```

```
use school;

create table student_info(
id int,
fname varchar(25),
lname varchar(25)
);

show tables;

insert into student_info(id,fname,lname)
values
(1,'vamsi','reddy'),
(2, 'sai','kumar');

select * from student_info;

create table teacher_info(
tech_id int,
tech_fname varchar(25),
tech_lname varchar(25),
salary int

select * from teacher_info;

#SQL update Statement
update teacher_info set salary = 70000;

create table teacher_data(
id int,
fname varchar(25),
lname varchar(25),
salary int,
primary key(id)
);

insert into teacher_data(id,fname,lname,salary)
values
(1,'vamsi','reddy',68900),
(2,'sai','kumar',50000),
(3,'kumar','sai',null);

select * from teacher_data;

update teacher_data set salary = 40000 where id=3;

#SQL Delete Statement

delete from teacher_data where id=2;
```

```

#

/*
SQL ALter Table
---- Add Column in Existing Column
---- Alter Table - Drop Column
*/

alter table teacher_data add location varchar(25);

alter table teacher_data drop location;

select * from teacher_data;


#create Database
Create database customer;

use customer;
# create customer_info Table
create table customer_info(
id integer auto_increment,
first_name varchar(25),
last_name varchar(25),
salary integer,
primary key(id)
);

show tables;
select * from customer_info;
#insert records into tables
insert into customer_info(first_name,last_name,salary)
values
('Vamsi','Reddy','150000'),
('Siva','Sai','67000'),
('Krishna','Kumar',60000),
('Vicky','Sai',Null); #it means that value is missing here

#Gettinh the all null values in table
select * from customer_info where salary is null;

#sql update statement it used to Replace the null values

update customer_info set salary=70000 where id=4;
select * from customer_info;

# delete statement - used to delete the data
delete from customer_info where id=4;

```

```

select * from customer_info;

# Alter Table main
# Add the column using Alter to existing table

alter table customer_info add phonenummer integer;
select * from customer_info;

alter table customer_info add dob Date;
select * from customer_info;

#to view schema
desc customer_info;

#drop the column using alter
alter table customer_info drop column dob;
select * from customer_info;

# Session 3
/*
What is SQL Constraints?
- SQL constraints are used to specify rules for the data in a table.
- Constraints are used to limit the type of data that can go into a
  table.

1) Not Null - A column cannot have a NULL value
2) Unique - All values in a column are different
3) Primary Key - A combination of a NOT NULL and UNIQUE. Uniquely
  identifies each row in a table
4) Foreign Key - Prevents actions that would destroy links between tables
5) Default - Sets a default value for a column if no value is specified
6) Check - the values in a column satisfies a specific condition
7) Index - Used to create and retrieve data from the database very
  quickly

*/

create database person;
show databases;
use person;

#create a not null Constraint
create table person_info(
id int not null,
First_name varchar(35) not null,
Last_name varchar(35) not null,
age int
);

select * from person_info;

# to view the schema

```

```

desc person_info;

#use the modify method to do changes in table
alter table person_info modify age int not null;

# unique constraint
use person;
create table student(
id int not null,
first_name varchar(34),
last_name varchar(23),
age int not null,
unique(id)
);

desc student;

#insert the values
insert into student values(1,'vamsi','reddy',24);
#repeating the same id
select * from student;
insert into student values(3,'vamsi1','tr ',24);
select * from student;

#make a unique for another column
alter table student add unique (first_name);

#adding two feature as unique
alter table student
add constraint uqi_student unique(age,first_name);

desc student;

#can we drop the constraint

alter table student
drop index uqi_student;

## primary key constraint

create table student2(
id int not null,
fname varchar(25) not null,
lname varchar(25),
age int,
primary key(id)
);

desc student2;

#drop the primary key
alter table student2 drop primary key;

```

# Foreign Key - it is used to link 2 tables together and it sometime knows as referenc key

```
create table student3(  
id int not null,  
first_name varchar(34) not null,  
last_name varchar(23) not null,  
age int not null,  
salary int,  
primary key(id)  
);
```

```
desc student3;
```

#creating another table

```
create table department(  
id int not null,  
department_id int not null,  
department_name varchar(35),  
primary key(department_id),  
constraint fk_studentdepart foreign key(id) references student3(id)  
);  
desc department;
```

# check constraints  
# it is used

```
create table student4(  
id int not null,  
first_name varchar(34) not null,  
last_name varchar(23) not null,  
age int not null,  
salary int,  
primary key(id),  
check(salary<50000)  
);
```

```
desc student4;
```

```
insert into student4 values(1,'Vamsi','Reddy',23,30000);  
insert into student4 values(2,'Vamsai','Reafddy',22,79000);
```

```
select * from student4;
```

```
## Default Constraint  
create table student5(  
id int not null,  
first_name varchar(34) not null,  
last_name varchar(23) not null,  
location varchar(34) default 'Hyderabad'  
);
```

```
desc student5;
```

#----- Index Constraint-----#

/\*

In MySQL, an index is a data structure that improves the speed of data retrieval operations on a database table.

It is created on one or more columns of a table, and it provides a fast way to look up and retrieve rows based on the values in those columns.

An index constraint, on the other hand, is not a standard term in MySQL.

It seems there might be some confusion in the terminology.

If you are referring to constraints related to indexes, there are certain constraints that can be associated with indexes in MySQL, such as UNIQUE and PRIMARY KEY constraints.

\*/

```
create database indexty;
```

```
use indexty;
```

```
-- Creating a table with a UNIQUE constraint
```

```
CREATE TABLE products (  
    product_id INT PRIMARY KEY,  
    product_name VARCHAR(50) UNIQUE,  
    price DECIMAL(10, 2)  
);
```

```
-- Inserting data into the table
```

```
INSERT INTO products (product_id, product_name, price) VALUES  
(1, 'Laptop', 999.99),  
(2, 'Smartphone', 499.99),  
(3, 'Tablet', 299.99);
```

```
-- Attempting to insert a duplicate product_name (violating the UNIQUE  
constraint)
```

```
-- This will result in an error
```

```
INSERT INTO products (product_id, product_name, price) VALUES  
(4, 'Laptop', 899.99);
```