## Creating a new database

```
CREATE DATABASE SUST_AUTO_RICKSHAW_MANGEMENT_SYSTEM;
```

## Showing already created databases

```
SHOW DATABASE;
```

## Delete a database

```
DROP DATABASE SUST_AUTO_RICKSHAW_MANGEMENT_SYSTEM;
```

## Creating a new database

```
CREATE DATABASE SUST_AUTO_RICKSHAW_MANGEMENT_SYSTEM
```

## Creating a student table wit data and data types

```
CREATE TABLE Student(

    Roll int(5),
    Name varchar(20),
    Gender varchar(15),
    GPA double(3,2),
    City varchar(10),
    DateBirth Date,
    PRIMARY KEY(Roll)

);
```

## Renaming Table

```
RENAME TABLE Student TO StudentInfo ;
```

## Deleting Table

```
DROP TABLE StudentInfo ;
```

## Inserting Data in table

```
INSERT INTO studentinfo
VALUES
(102,'Mridul','Male',3.95,'Dhaka','2000-04-11'),
(103,'Mridul','Male',3.95,'Dhaka','2000-03-11');
```

```
INSERT INTO studentinfo
(Roll,Name,Gender,GPA,City,Datebirth)
VALUES
(102,'Mridul','Male',3.95,'Dhaka','2000-04-11'),
(103,'Mridul','Male',3.95,'Dhaka','2000-03-11');
```

## Seleting Data in table (Finding Data)

### Selecting Only One Column

```
SELECT Roll FROM studentinfo;
```

### Selecting Multiple Column

```
SELECT Roll,Name
FROM studentinfo;
```

### Selecting All Column

```
SELECT *
FROM studentinfo;
```

### Use Distinct to remove duplicates values from selected column

```
SELECT DISTINCT Name
FROM studentinfo;
```

### Use Order by to sort duplicates values from selected column

```
SELECT GPA
FROM studentinfo
ORDER BY GPA;
```

```
SELECT GPA
FROM studentinfo
ORDER BY GPA DESC;
```

### Use Where to Search Data from selected column with condition

```
SELECT Roll
FROM studentinfo
WHERE GPA > 3.90 ;
```

```
SELECT GPA
FROM studentinfo
ORDER BY GPA DESC;
```

## Updating Data in table

```
UPDATE studentinfo
SET Name = 'Sumonta'
WHERE Roll = 102;
```

## Deleting Data in table

```
DELETE FROM studentinfo
WHERE Roll = 103;
```

## Add Foreign Key

```
 // The "PersonID" column in the "Persons" table is the PRIMARY KEY in the "Persons" table.
 // The "PersonID" column in the "Orders" table is a FOREIGN KEY in the "Orders" table.

 CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)
);
```

## AND, OR and NOT Operators

```
SELECT * FROM Customers
WHERE Country = 'Germany' AND City = 'Berlin';
```

```
// if any of the conditions separated by OR is TRUE.

SELECT * FROM Customers
WHERE City = 'Berlin' OR City = 'Stuttgart';
```

```
// if the condition(s) is NOT TRUE.

SELECT * FROM Customers
WHERE NOT Country = 'Germany';
```

## CHECK on CREATE TABLE

```
ID int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Age int,
CHECK (Age>=18)
```

---

### Consider the following Banking Database

- branch(branch name, branch city, assets)
- customer(customer name, customer street, customer city)
- loan(loan number, brach name, amount)
- borrower(customer name, loan number)
- account(account number, branch name, balance)
- depositor(customer name, account number)

#### Find the names of all branched located in “Dhaka”

```
SELECT branch_Name
FROM branch
WHERE branch_city = “Dhaka”;
```

#### Find the names of all borrowers who have a loan in branch “Mirpur”

```
SELECT customer_name
FROM Borrower, Loan
WHERE Borrower.loan_number = loan.loan_number
and branch_name = “Mirpur”;
```

#### Find all loan numbers with a loan value greater than BDT100,000

```
SELECT loan_number
FROM loan
WHERE amount > 100000;
```

#### Find the names of all depositors who have an account with a value greater than BDT60,000

```
SELECT customer_name
FROM depositor, account
WHERE account.account_number = depositor.account_number
and balance > 60000;
```

#### Find the names of all depositors who have an account with a value greater than BDT60,000 at the “Motejheel” branch

```
SELECT customer_name
FROM depositor, account
WHERE account.account_number = depositor.account_number
and balance > 60000
and branch_city = “Motejheel”;
```