Constraints

Types of Constraints

- **NOT NULL** - Ensures that a column cannot have a NULL value
- **UNIQUE** - Ensures that all values in a column are different
- **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- **FOREIGN KEY** - Uniquely identifies a row/record in another table
- **CHECK** - Ensures that all values in a column satisfies a specific condition
- **DEFAULT** - Sets a default value for a column when no value is specified
- **INDEX** - Used to create and retrieve data from the database very quickly

- Syntax:

```
CREATE TABLE table_name (
    column1 datatype constraint,
    column2 datatype constraint,
    column3 datatype constraint,
    ....
);
```

Unique and NOT NULL Constraint

```
CREATE TABLE Persons (
    ID integer(3)  NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255) NOT NULL,
    Age integer(3),
      UNIQUE (ID)
);
```

```
CREATE TABLE Persons (
    ID integer(3)  NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255) NOT NULL,
    Age integer(3),
    CONSTRAINT UC_Person UNIQUE (ID,LastName)
);
```

Primary Key

```
CREATE TABLE Persons (


ID integer(3),
LastName varchar(255) NOT NULL,
FirstName varchar(255) NOT NULL,
Age integer(3),

PRIMARY KEY (ID)

);
```

```
CREATE TABLE Persons (
    ID integer(3),
    LastName varchar(255) NOT NULL,
    FirstName varchar(255) NOT NULL,
    Age integer(3),

    CONSTRAINT PK_Person PRIMARY KEY (ID
       ,LastName)

);
```

Foreign Key

```
CREATE TABLE Orders (
    OrderID integer(3)  NOT NULL,
    OrderNumber integer(3)  NOT NULL,
    PersonID integer(3),
    PRIMARY KEY (OrderID),
    CONSTRAINT FK_PersonOrder FOREIGN KEY (P
ersonID)
    REFERENCES Persons(PersonID)
);
```

CHECK Constraint

```
CREATE TABLE Persons (
    ID integer(3) NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age integer(3),
    CHECK (Age>=18)
);
```

```
CREATE TABLE Persons (
    ID integer(3)  NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age integer(3),
    City varchar(255),
    CONSTRAINT CHK_Person CHECK (Age>=18 AND
      City='Sandnes')
);
```

Example

```
create table CityMaster (
      CityID integer(4) primary key,
      cityName varchar(255) default 'toronto'
);
```

```
create table persons(
      ID integer(3),
      LasteName varchar(255) NOT NULL,
      FirstName Varchar(255),
      Age integer(3),
      CityID integer(4),
      constraint pk_persons primary key (ID),
      constraint ck_persons check (age>=18),
      constraint fk_persons foreign key (CityID)
      references CityMaster (CityID)
);
```

List of indexes

```
Show index from customers;
Show index from persons;
```

Create table from existing table

Syntax

- **CREATE TABLE new_table_name AS**
  SELECT column1, column2,...
  FROM existing_table_name
  WHERE ....;

- **CREATE TABLE newTable LIKE existingTable;**

Example

- **create table cust as select * from customers;**

- **create table c like customers;**

Insert from existing table

INSERT INTO

- Syntax:
- **INSERT INTO TableName SELECT columns FROM TableName WHERE criteria**

- Example:
- **insert into c select * from cust where customerID>5;**

Changing table properties

ALTER

- ALTER TABLE can
  - Add a new column
  - Remove an existing column
  - Add a new constraint
  - Remove an existing constraint

Add Column

- Syntax

```
ALTER TABLE <table>
 ADD COLUMN <col>
```

- Example:
- **alter table cust add column country varchar(100);**

Modify Column

- Syntax

```
ALTER TABLE <table>
 MODIFY COLUMN <col>
```

- Example:
- **alter table cust modify column postalCode varchar(6);**

Rename Column

- Syntax

```
ALTER TABLE <table>
 CHANGE COLUMN <old_name>
     <new_name> <col specification>
```

- Example:
- **alter table cust change column postalCode zipcode varchar(6);**

Drop Column

- Syntax

```
ALTER TABLE <table>
 DROP COLUMN <col>
```

- Example:
- **alter table cust drop column city;**

Add constraint

- Example1:
- **alter table cust add primary key (customerID);**
- Example2:
- **alter table cust add constraint unique (country);**
- Example3:
- **alter table cust modify name varchar(40) not null;**

Drop constraint

- Example1:
- **`alter table cust modify name varchar(40);`**

- Example2:
- **`alter table cust drop index customerID;`**

Drop index

- Display index and keyName:
- **`show index from TableName;`**

- Syntax:
- **`drop index keyName on tableName;`**

- Example:
- **`drop index country on cust;`**

Update

Update

- Syntax:

```
UPDATE table_name
    SET column1 = value1,
        column2 = value2, ...
    WHERE condition;
```

Update

- Example:
- **update cust set country="canada" where customerID>5**;

- Example:
- **update cust set postalCode="M1M2M3" where customerID=1;**

Delete Records

Delete

- Syntax:

```
DELETE FROM table_name
WHERE condition;
```

Delete

- Example:
- **`delete from cust where nickName="DJ";`**

- Example:
- **`delete from cust;`**

OR

- **`delete * from cust;`**

Deleting table

Drop table

- Syntax:
- **`DROP TABLE table_name;`**

- Example:
- **`Drop table c;`**