**Types of Sql Statements**

SQL commands are mainly categorized into four categories as:

1. DDL – Data Definition Language(create,drop,alter,truncate,rename,comment)
2. DQl – Data Query Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language



**TRUNCATE**

TRUNCATE: This is used to remove all records from a table, including all spaces allocated for the records are removed.

truncate table GraduatedStudents

Now all the records are removed. Only the structure of the tables is preserved.

**Creating a table**

Create database databasename;

create table students(

student\_id int primary key,

name varchar(20),

major varchar(20)

)

Here we gotta set primary keys manually which is not right:

insert into students values(1,'Vako',20);

insert into students values(2,'Vako',20);

But with identity keyword it sets that automatically and starts from zero.

create table people(

person\_id int primary key identity,

name varchar(20),

age int

)

So we would just do:

insert into people values('Vako',20);

insert into people values('Lala',25);

That’s it!

**Comments**

--our comment

**Alter a table(adding a new column)**

if we want to change a table we use alter keyword!

If we forgot to add a column then we can use ‘alter’ keyword:

Alter table tableName add columnName varchar(16)

alter table students add gpa int

then we use ‘update’ keyword to set the values because they are null.

Update tableName set column1=value1, column2=value2

where condition

update students set gpa=7 where student\_id=3

**SQL constraints (NOT NULL, UNIQUE)**

If we wanna add a required column meanign that it cannot be null then we use **not null** attribute:

create table people(

id int primary key identity,

[name] varchar(16) not null,

age int unique

)

So now there can’t be equal ages, and name cannot be omitted however it can be an empty string.

insert into people values('Vahid',20)

insert into people values('Hey',20)

Violation of UNIQUE KEY constraint

We can also add contsraints as shown below

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    City varchar(255),  
    CONSTRAINT CHK\_Person CHECK (Age>=18 AND City='Sandnes')  
);

**Naming Convetion for Constraints**

PK\_TableName\_ColumnName -- primary key constraint FK\_TableName\_ColumnName - foreign key constraint CK\_TableName\_ColumnName - check constraint

UQ - for unique constraint

DF - for default constraint

**Deleting a record(row) from a table**

We can use just **delete** keyword here:

delete from people where [name]=''

**Foreign key**

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.

create table pets(

pet\_id int primary key identity,

name varchar(16) not null

)

alter table people add pet\_id int foreign key references pets(pet\_id)

**Joins**

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

Inner join -  selects records that have matching values in both tables

LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table (if left table has more fields then right table’s fields will be set to null)

RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table

FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table



select \*

from people

inner join pets

on people.pet\_id=pets.pet\_id

**Deleting a field (column)**

In order to delete a column, we use “drop” keyword.

alter table people add [name] nvarchar(20)

**Group by (clause)**

For instance, we have got people from the same country and now we can use group by clause to select all the people from the same country.

If there is a cloumn and aggregate function up in our select statement then we can’t select like this->

select name,count(\*) from Students

Column 'Students.name' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.

So here we gotta use group by clause.

select country,count(\*) from Students group by country

select country,name,count(\*) from Students group by country,name

Here it first groups them by country and when there are two identical country names it groups them by their names

America Jeck 1

Cacusus Lala 1

Azerbaijan Nihad 1

Azerbaijan Vahid 1

After group by clause, if we wanna filter it then we have to use having clause.

select country,name,count(\*) from Students group by country,name having country='Azerbaijan'

We can use where clause only before group by clause!

select country,name,count(\*) from Students where country='Azerbaijan' group by country,name

select CustomerName, SUM(Amount) as Expenditure from Orders group by CustomerName order by CustomerName

**Copying values from another table**

insert into GraduatedStudents select name,surname,phoneNumber,score,comment from Students where isGraduated=1

**Check constraint**

The CHECK constraint is used to limit the value range that can be placed in a column.

create table Students(

Id int primary key identity,

Name varchar(16) check(name like '%a%')

)

insert into Students values('Hey')

The INSERT statement conflicted with the CHECK constraint "CK\_\_Students\_\_Name\_\_3A81B327".

It has to contain an a in it!

create table Students(

Id int primary key identity,

Name varchar(16) check(name like '%a%'),

Age int check(Age>1 and Age<10)

)

insert into Students values('Hay',9)

**Union,Union All,Intersect,Except**

1. **UNION**: Combine two or more result sets into a single set, without duplicates.
2. **UNION ALL**: Combine two or more result sets into a single set, including all duplicates.
3. **INTERSECT**: Takes the data that is in both tables
4. **EXCEPT**: Takes the data that is in the first table but not in the second

select \* from UnGraduatedStudents

select \* from GraduatedStudents



select \* from UnGraduatedStudents

union

select \* from GraduatedStudents



select \* from UnGraduatedStudents

union all

select \* from GraduatedStudents



select \* from UnGraduatedStudents

intersect

select \* from GraduatedStudents



select \* from UnGraduatedStudents

Except

select \* from GraduatedStudents

