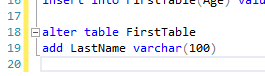
**Structed Query Language**

DDL- data defination language

**Create (**misol create table\_name(column\_name datatype, column\_name2 datatype, …..**)**

**Alter –** table ni arxitekturasini ozgartirishda kerak bo’ladi

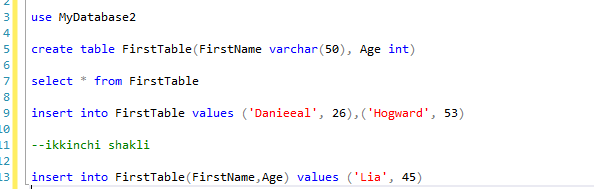
**Drop –** drop butunlay table ni ochirib yuboradi

**Truncate –** table ni ichidagi ma’lumotlarni ochrish uchun ishlatilinadi, Faqat column name lar qoladi Ichidagi malumotlar yoqoladi

**Varchar-** so’zlarni kiritiladigaan column bosa varchar beriladi

**Int –** butun son

Insert into bilan table ga value qo’shsa bo’ladi:

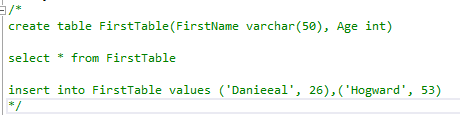
  
--varchar ni yonidagi son bu column name ga kiritiladiga max harflar

Comment qilish va olib tashlash

CTRL+K+C **comment**

CTRL+K+U **uncomment**

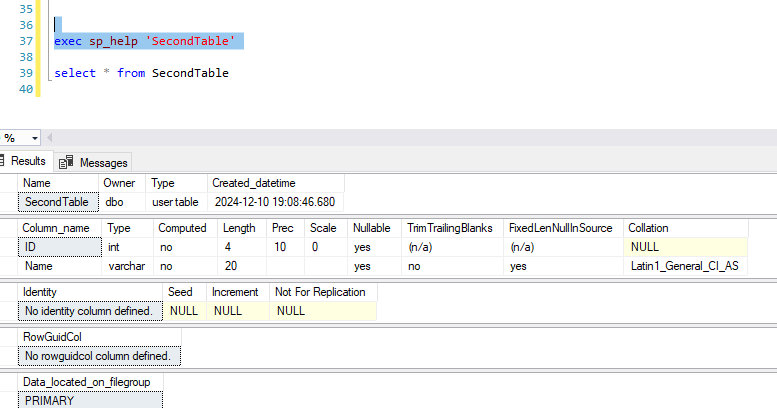
---keyin bir necha rowlarni birdan comment qilish uchun /\* dan foydalanamiz ya’ni qayerda boshlansa /\* tugash qatoriga esa \*/ ni qo’yamiz. Masalan:



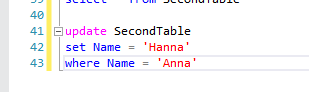
DDL- yani data definition language bilan table ni o’zini ozgartirish uchun ishlatilinadi

**DML yani Data manipulation Language bilan table ni ichidagi ma’lumotlarni boshqaramiz.**

exec sp\_help bu table ni arxitekturasini ko’rsatib beradi. Masalan:



**Update** bilan ishlash: bu bilan malum bir table ichidagi malumotlarni o’zgartira olamiz. Masalan:

Bu yerda data update qilinyapti demak birinchi nimani ozgartirmoqchi bolsak oshani **set** bilan yozib olamiz va **where** bilan qaysi datani ozgartirish kerakligini kiritamiz.

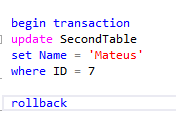
**Delete from** SecondTable = **Truncate** SecondTable

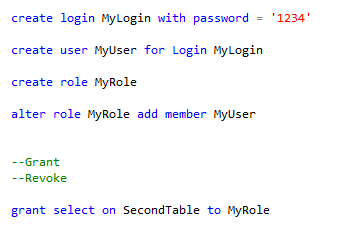
Bu code bilan esa qaysidir aniq bir rowni o’chirib tashlasa bo’ladi:

**DQL – Data Query Language**

--Select

**TCL Transaction Control Language**

**Begin transaction** bu bilan yozgan codimizni check qilishimiz mumkin yani biror code qatori oldiddan shuni yozib kyn run qilsak nma bolishini korib oolishimiz mumkin agar togri bolsa yoki code ni run qilish ni davom ettirmoqchi bo’lsak **commit**  deb kiritamiz yoki o’z holiga qaytarmoqchi bo’lsak **rollback** deb yozamiz. Masalan:



DCL Data Control Language – bu bilan table larga login password qoyish va shunga oxshash operatsiyalar bajariladi.

**DATA TYPES**

url : https://www.w3schools.com/sql/sql\_datatypes.asp

**Wildcards(bular like bilan ishlaydi)**

**%** - bu qaysidir harfdan kyn bir yoki bir nechta harf kelgan taqdirda uni qidirish uchun ishlatilinadi. Masalan: *movies* tabledan ‘King kong’ filmini qidirishim kerak shu holatda man ‘King%’ bersam, har qanday nomi King deb boshlangan kinolarni topib beradi.

**\_** - bu esa bitta harf ya’ni characterni bildiradi. Masalan ‘King Kong’ ni qidirmoqchi bo’lsam, ‘King Kon\_’ desam chiqadi. '\_r%' bu misolda 2-harfi r bolgan so’z qidirilyapti

**[] –** bu bilan ma’lum bir harfdan harfgacha bo’lgan harflarni kiritsak bo’ladi. Masalan:

select \* from soccer.player\_mast

where playing\_club not like '[A-L]%' bu yerda playing\_clubida bosh harfi A dan L gacha bo’lmagan club lar chiqadi.

**distinct –** duplicatlar ni yoqotin bitta qilib chiqaradi value larni

select top 10 percent \* from employee.employees

order by salary

**Bu yerda** top 10% ga kiradiganlarni chiqarish uchun query yozilgan

select \* from employee.employees

order by salary desc

offset 3 rows fetch next 5 rows only

Offsetni SQL da ishlatilinishi, bu yerda 3 ta row tashlab 5 tasini olishi ko’zda tutilgan. ***OFFSET*** va ***FETCH***

Ikki column ni qo’shib bitta column qilish uchun :

select emp\_name + ' ' + job\_name as fullname from employee.employees

yoziladi.

**Lesson 6**

**Union all –** bu ikkita jadvalni qo’shish uchun ishlatilinadi.(duplicatlarni olib tashlamaydi)

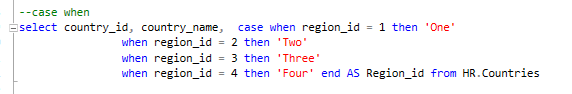
**Union –** bunda esa by default duplicatlarni olib tashlab, asc order qilib chiqarib beradi.

**Except** – esa birinchi ko’rsatilgan jadvalda bor lkn ikkinchisida yo’qlarini chiqarib beradi ya’ni difference qiladigan value larni chiqaradi.

**Intersect** – Bu esa except ni teskarisi ya’ni ikkala jadvalda borlarini chiqarib beradi. Ikkalasida bir xil bolgan data ni chiqaradi.

**If lar SQL da**

**Case when … then …. End**

* *Case when condition then true result else false result* -- shu structura bo’yicha ishlaydi lekin har doim ham else shartmas. Ammo **end** ni har doim qo’yish kerak.
* 

**IIF**

Bu esa xuddi case when ni ikkinchi xolati faqat excel ga oxshab, () va , lar bilan bo’ladi syntax lar. Misol:

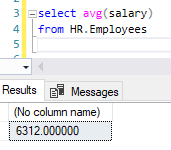
select country\_id, country\_name, IIF(region\_id = 1, 'One',

IIF(region\_id=2,'Two',

IIF(region\_id=3,'Three',

'Four'))) as region\_id from HR.Countries

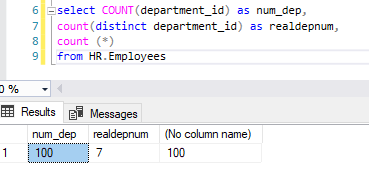
**aggregate functions**

Avg(column\_name)

select cast(avg(salary) as decimal(8,2)) as average\_salary

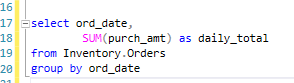
from HR.Employees

bu yerda esa type decimal da va column ga name berib chiroyli chiqarish(cast bilan)



**Count**

Count sanaydi. Misolda distinctsiz ishlatilganida column dagi barcha rowlarni sanab chiqqan, agar distinct qo’shsak duplicatlarni olib tashlab, aniq nechtaligini sanaydi, **count(\*)** ham barcha rowlarni sanab chiqadi, **null** larni ham lekin **count(department\_id)** esa **null** ni olmaydi.



Sum ga misol:

Sum, max ya’ni agregation lar bn group by ni ishlatish kk chunki bular bittadan olrtiq bo’lgan valuelarni hisoblash uchun ishlatilinadi. Misolda ham group\_date bilan har birk un ajratilib o’sha kunlarning Sumi hisoblangan.