SQL NOTES CHAALAK BROOOO

create table employee(id int,name varchar(20),dept varchar(20),salary int,city varchar (15)); -- varchar la length specify karavich lagte

select \* from employee --to view table content

insert into employee values (02,'Baala','BE\_Elec',25000,'BLR'); -- for insert single entry in a row

insert into employee values(03,'ABhi','BE\_Mech',35000,'ASN'); insert into employee values(04,'Prats','BE\_Agri',45000,'RATN'); insert into employee (id,name,dept,salary,city) values

(05,'Sayali','BE\_Elect',55000,'Pune'),(06,'Vedu','BE\_civil',65000,'NSk'); --For multiple entries at a one time

exec sp\_rename 'employee.id','ID','Column'; -- to rename col name sp\_rename 'employee.[name]','NAME','column' -- to rename multiple col names GO

sp\_rename 'employee.[dept]','DEPT','column' GO

sp\_rename 'employee.[salary]','SALARY','column' GO

sp\_rename 'employee.[city]','CITY','column' GO

insert into employee values (02,'Baala','BE\_Elec'); -- IT WILL SHOW ERROR AS INSERTED VALUES ARE LESS THAN SPECIFIED COLS

insert into employee(ID,NAME,DEPT) values (0,'Baala','BE\_Elec');--IT WILL INSERT VALUES IN PARTICULAR COLS AND REST WILL NE NULL

insert into employee values (07,'Baala','BE\_Elec','','');-- INT CHYA JAGI EMPTY ASEL TAR '0' YEYIL ANI STRING CHYA JAGI 'EMPTY FIELD' YENAR

insert into employee values (02,'Baala','BE\_Elec',NULL,NULL);-- BEST PARCTICE IS TO WRITE NULL IF VALUES ARE NOT AVAILABLE INSTEAD OF LEAVING IT EMPTY

--CLAUSES

-- 1) WHERE CLAUSE (a) COMPARATIVE OPERATOR (=,!=,<,>,<=,>=)

select \* from employee where DEPT = 'BE\_Elec'; -- Comparative made fakt be\_elec hava asel tr EQUAL TO KARAYCH

select \* from employee where DEPT != 'BE\_Elec'; -- jar be\_elec sodun hava asel tr NOT EQUAL TO

select \* from employee where SALARY < 50000; -- LESS THAN

select \* from employee where SALARY <= 55000; -- LESS THAN EQUAL TO select \* from employee where SALARY > 50000;-- GREATER THAN

select \* from employee where SALARY > 55000;-- SOBAT 2 CMD SELECT KELA TAR 2 WINDOW MADE TE EXECUTE HONAR

select \* from employee where SALARY >=55000;--GREATER THAN EQUAL TO

--(b) LOGICAL OPERATOR (and,or)

select \* from employee where DEPT= 'BE\_elec' and CITY='blr';

select \* from employee where DEPT= 'BE\_elec' and CITY='kyn'; -- FOR "AND" OPERATOR BOTH THE CONDITION MUST BE SATISFIED

select \* from employee where DEPT= 'BE\_elec' OR CITY='KYN'; -- FOR "OR" ANY CONDITION MUST BE SATISFIED BUT BOTH THE VALUES WOULD BE SHOWN

--(c) IN/NOT IN OPERATOR

select \* from employee where CITY in ('kyn','ratn') -- SHOWS DATA WHERE CITY WOULD BE KYN AND RATN

select \* from employee where CITY not in ('kyn','ratn') -- SHOWS DATA LEAVING KYN AND RATN

select \* from employee where CITY in ('kyn','ratn') AND SALARY=15000; -- USING ALL 3 OPERATOR SIMULTANEOUSLY

select \* from employee where CITY in ('kyn','ratn') AND SALARY>=15000;

--(d) BETWEEN/NOT BETWEEN (Used mostly for INT operation) select \* from employee where SALARY between 10000 and 30000; select \* from employee where SALARY not between 10000 and 30000;

select \* from employee where NAME between 'A' and 'S'; -- A pasun suru hoil pan S nahi yenar karan te S-1 honar

insert into employee values (09,'zakir','BE\_Elec',25000,'BLR');

select \* from employee where NAME between 'A' and 'zz';-- jar aplya Z cha pn data hava asel tr ZZ takycha

select \* from employee where NAME between 'A' and 'z';-- nusta Z takla tr te Z-1 honar mnje data fkt Y paryntacha yenr

--(e) Like(exact data not known only description known)

select \* from employee where NAME like 'a%'; -- starting with 'a' pude kahi yeyil select \* from employee where NAME like '%i'; -- ending with 'i' starting la kahi asel select \* from employee where NAME like '%a%';-- staring n ending nahi mahit pan madhe

'a' aslela disel pan kute hi asu shakto 'a' select \* from employee where NAME like '%i%' ;

select \* from employee where NAME like 's%i'; -- first and last alphabet is known select \* from employee where NAME like '\_a%'; -- only 2nd alphabet is known

select \* from employee where NAME like '\_a%' and SALARY in (25000,55000); -- LIKE-IN

--> ya madhe exact value chi range daychi (1st range,last range)

select \* from employee where NAME like '[ab]%'; -- ya madhe ektar 'a' kiva 'b' ne start honare data yeyil

select \* from employee where NAME like '[^ab]%'; -- leaving 'a' and 'b' sarva yetil select \* from employee where NAME like '[a-s]%'; -- 'a' to 's' chya range madhe sarva

show hoil

select \* from employee where NAME like '%[s]%';

select \* from employee where NAME like '%[rit]%'; -- kute hi 3 paiki aslela alphabet dakhvtil

--(f) ARITHMETIC OPERATOR --> +,-,\*,/

select \*,new\_salary = salary\*0.2 from employee;

select \*,new\_salary = salary + salary\*0.2 from employee;

select \*,new\_salary = salary + salary\*0.4 from employee where name='sayali'; select \*,new\_salary= (((salary + salary \* 0.3)/30)\*2) from employee where

name='sayali';

-- 2) ORDER-BY CLAUSE (USED FOR ASC/DESC OPERATION)

select \* from employee

select \* from employee order by SALARY desc;

select \* from employee order by SALARY;-- fakt asa lihila tr asc order madhech yenar ani lihaych aselch tr 'asc' lihaych

select \* from employee where name in ('Ajay','Sayali') order by CITY desc; select \* from employee where name in ('Ajay','Sayali') order by salary;

--\*\*DML CMDS\*\* -->(UPDATE,DELETE,INSERT)

--\*UPDATE\*

update employee set name='Appu' where id=5; -- updates data specifically UPDATE employee set salary=75000 where salary is null;

update employee set salary=85000,city ='KP' where salary=0; -- can update multiple entries of single row

--\*\*DELETE\*\* --> IT CAN UNDO OR ROLL BACK THE CONDTN

delete employee where name='Appu'; -- deletes single row

select \* from INFORMATION\_SCHEMA.TABLES; -- CHECKS ALL TABLES IN DB/

select \* from INFORMATION\_SCHEMA.TABLES;DELETE employee -- FAKT DATA DELETE HONAR TABLE STRCTURE TASACH RAHNAR KARAN DELETE IS DML CMD

--\*\*DDL\*\* (create,drop,alter,rename,truncate)--> it is related to structure|WHERE CLAUSE IS NOT USED IN DDL CMDS,ONLY FOR DATA RELATED OPNS WE USE WHERE CLAUSE

--\*\*DROP\*\* --> WE CANT ROLL BACK OR UNDO AS WHOLE TABLE GETS DELETED

drop table employee;

--\*\*TRUNCATE\*\* (command is used to delete all the rows from a table|is stucture related cmd| preserves the structure of the table for future use

create table student (roll\_no int,name varchar(20),class int,div varchar(5),age int) insert into student (roll\_no ,name ,class ,div ,age ) values (101,'Shreyas',1,'A',6),

(102,'Shreya',5,'B',7),(103,'Shree',4,'C',8),(104,'Reea',1,'A',5),

(105,'Sonal',3,'B',8),(106,'Abhiyas',2,'D',7),(107,'Abhi',3,'B',8), (108,'Shankar',2,'A',6)

select \* from student truncate table student

--\*\*ALTER\*\* (TO CHANGE IN STRUCTURE)

alter table student add marks int; -->(to add single col)

alter table student add place varchar(10),contact\_no int; -->to add multiple cols at a time

alter table student drop column marks; --> deletes single col

alter table student drop column place,contact\_no; --> deletes multiple col

alter table student alter column contact\_no varchar(10); --> changes the datatype of contact from int to string (string to int hot nahi)

update student set contact\_no='abcd' where name='shree';

update student set contact\_no='abcd' where name=sonal; --> Invalid column name 'sonal' error is shown

update student set contact\_no='1234' where name='sonal';

--(new data ka datatype change hoga,previous wala data int hi rahega)

select \* from student where age between 7 and 8; select \* from student where name like '%[s]%'; select \* from student where name like '%s%';

--\*\*AGG FUNCTIONS\*\* -->(SUM,AVG,MIN,MAX,TOP,DISTINCT) --> SUM AND AVG ARE USED FOR INT ONLY REST ARE FOR INT & VARCHAR

select \* from student select min(age) from student

select min(age) as min\_age from student select max(age) from student

select max(age) as max\_age from student select avg(age) as avg\_age from student select sum(age) as sum\_age from student

select count(age) from student -- SHOWS NO OF RECORDS

SELECT distinct(age) from student -- shows u unique and single records and not the repeated ones

select top 2 \* from student select top 2 name from student

select count (distinct(age)) as nwcount from student

--\*\*GROUP BY CLAUSE\*\* --> W/O AGG FN GBC WONT WORK

select \* from student

select div,avg(age) from student group by div select div,min(age) from student group by div

select contact\_no,avg(age) from student where contact\_no is not null group by contact\_no order by avg(age)

select div,avg(age) from student group by div order by avg(age)

--\*\*HAVING CLAUSE\*\*--> USED TO FILTER GROUP BY CLAUSE

select div,avg(age) as avg\_age from student group by div having avg(age)<7 select div,avg(age) as avg\_age from student group by div having avg(age)<5

--\*\*COMBINATION OF CLAUSES (W-G-H-O)

--YOU CAN USE ALL CLAUSES IN ONE QUERY,BUT IT HAS SEQUENCE-->WHERE--> GROUP BY -->

HAVING --> ORDER BY

select \* from student

select div,avg(age) from student group by div

select div,min(age) as min\_age from student where div='B' group by div having min(age)

>6 order by min(age)

select div,avg(age) as new\_age from student group by div having avg(age)>5 order by avg(age) desc

select div,count(div) from student group by div having count(div)>2

select name,div,count(div) from student group by div having count(div)>2 --Column 'student.name' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.

--\*\*CONSTRAINTS\*\*

--WHILE CREATING TABLE CONSTRAINTS APPLY OR SET RULES AS FOLLOWS:-

--1)NOT TO ENTER DUPLICATE VALUES

--2)NOT TO ENTER NULL

--\*\*TYPES OF CONSTRAINTS

--1)PRIMARY KEY- dont enter duplicate and null values|One table can hv only 1 PK

--2)UNIQUE KEY- dont enter duplicate values

--3)NOT NULL- dont enter null

--4)CHECK-checks condn ie entered while creating table

--5)Default-Set default key while creating table

--6)IDENTITY-By default col no starts with 1, only pply in int for auto increment. can also specift range.

--7)Foreign Key (FK)--> to create relation bw 2 tables| w\o using PK in 1st table u cant use FK in 2nd table

--8)Date- to add date col

--1)Primary Key (PK)

create table Q1 (id int Primary key,name varchar(10)); select \* from Q1

insert into Q1 values(01,'Amit');

insert into Q1 values(01,'Amt'); --Violation of PRIMARY KEY constraint

'PK Q1 3213E83FA6D0AA84'. Cannot insert duplicate key in object 'dbo.Q1'. The duplicate key value is (1)

insert into Q1 values(02,'Amt');

insert into Q1 values(null,'Abc'); --Cannot insert the value NULL into column 'id', table 'master.dbo.Q1'; column does not allow nulls. INSERT fails.

--2)Unique Key (UK)

create table Q2 (id int unique,name varchar(10)); select \* from Q2

insert into Q2 values(01,'Amit');

insert into Q2 values(01,'Amu');--Violation of UNIQUE KEY constraint

'UQ Q2 3213E83E40114747'. Cannot insert duplicate key in object 'dbo.Q2'. The duplicate key value is (1).

insert into Q2 values(null,'Abc')-- 2 veles jar execute kela tr same above error yeyil

create table Q3(id int primary key,name varchar(10) unique); select \* from Q3

insert into Q3 values(01,'Amit');

insert into Q3 values(02,'Amit');-- ikde name vegla hava ahe karan name "unique ahe" mhnanun varti alela same error yenar

insert into Q3 values(02,'Akiy');

insert into Q3 values(01,'Amt');-- ikde id PK aslya mule error yenar same as above

--3)Not Null

create table Q4(id int primary key,name varchar(10) not null); select \* from Q4

insert into Q4 values(01,'Amit');

insert into Q4 values(02,null);-- Cannot insert the value NULL into column 'name', table 'master.dbo.Q4'; column does not allow nulls. INSERT fails

insert into Q4 values(02,'Amit');-- duplicate name will execute except null value

--\*Combn of UK & not null = PK

create table Q5(id int primary key,name varchar(10) unique not null); select \* from Q5

insert into Q5 values(01,'Amit');

insert into Q5 values(02,null); --behaves like pk

insert into Q5 values(02,'Amit');-- name la pn pk ahe mhnun error same as above

--4)Check (checks condn ie entered while creating table) create table Q6(id int primary key,marks int check(marks>35)); insert into Q6 values(01,60);

select \* from Q6

insert into Q6 values(02,35);--condn check honar,ikde 35 ahe pn condn madhe >35 ahe mhnun error yenar-->The INSERT statement conflicted with the CHECK constraint "CK Q6 marks 1FEDB87C". The conflict occurred in database "master", table "dbo.Q6", column 'marks'.

insert into Q6 values(04,45);

insert into Q6 values(03,25);--same as 35 error

create table Q7(id int primary key,marks int check(marks>60 and marks <=80)); select \* from Q7

insert into Q7 values(01,55); --range madhe nahiye mhnun condn satify nahi hot insert into Q7 values(06,80);

insert into Q7 values(03,65); insert into Q7 values(04,60); insert into Q7 values(05,77); insert into Q6 values(03,25);

--5)Default (Set default key while creating table)

create table Q8(id int,city varchar(10) default 'Mumbai')-- default value create kartanach dila tr te nantr values insert kartana city dila nahi kiva default dila tr hi value default chi yeyil mhanje Mumbai

insert into Q8 values (01,default) insert into Q8 values (01,'Pune');

insert into Q8(id) values (02);-- ikde fakt id mahitey pn city nahi dila tr default city yenar

select \* from Q8

--6)Identity(By default col no starts with 1)

create table Q9(id int identity ,city varchar(10));-- id la "identity constraint dila" mhnun jari values insert kartana fakt city dila tr te by default 1 pasunch suru hoil

insert into Q9 values('Pune'); insert into Q9 values('Mum'); insert into Q9 values('Nsk'); insert into Q9 values('blr'); insert into Q9 values('Pen');

insert into Q9 values(6,'dmb');-- id apun deu nahi shakt karan identity dila ahe create kartana..An explicit value for the identity column in table 'Q9' can only be specified when a column list is used and IDENTITY\_INSERT is ON.

select \* from Q9

create table P9(id int identity(10,10) ,city varchar(10));-- apun id la range pn deu shakto (start range,range chi step)

insert into P9 values('Pune'); insert into P9 values('Mum'); insert into P9 values('Nsk'); insert into P9 values('blr'); insert into P9 values('Pen'); select \* from P9

--7)Foreign Key (FK)--> to create relation bw 2 tables| w\o using PK in 1st table u cant use FK in 2nd table

create table dept1(did int primary key identity,dept varchar(10)); insert into dept1 values('mech');

insert into dept1 values('civl'); insert into dept1 values('comp'); insert into dept1 values('it'); select \* from dept1

create table stu(id int primary key identity,name varchar(10),did int foreign key references dept1(did));

insert into stu values ('Abhi',11); insert into stu values ('Ajay',1); insert into stu values ('bala',2); insert into stu values ('appu',3); insert into stu values ('prats',4); select \* from stu

select \* from dept1

--8)Date(to add date col)

create table acct\_details(acct\_no int primary key identity,name varchar (20),acct\_open\_date date);

insert into acct\_details values ('Abhi','2022-12-22'); insert into acct\_details values ('Ajay','2022-12-21'); insert into acct\_details values ('balu','2022-12-23'); select \* from acct\_details

--\*\*DATE-TIME\*\*

select getdate();-- gives current date n time

select getdate()+1; -- gives next day date with current time select getdate()-1; --gives previous day date with current time

--\*\*DATE FUNCTION\*\*

--a) Date Diff- to get diff btw 2 dates |eg:- to find age

--b) Date Add- to add specific date

--c) Date Part- if you need particular part from the date

--A) Date Diff (yyyy-mm-dd)--> format

select DATEDIFF(yyyy,'1997-07-22',GETDATE());

select DATEDIFF(yy,acct\_open\_date,getdate()) as age from acct\_details; select DATEDIFF(dd,acct\_open\_date,getdate()) as age from acct\_details; select \*, DATEDIFF(ss,acct\_open\_date,getdate()) as age from acct\_details;

--year- yy/yyyy

--month-m/mm

--quater- q/qq

--day- y/dy

--date- d/dd

--week-wk

--milsec-ms

--hour-h

--min-n

--sec- s/ss

--B)Date Add

select DATEADD(yy,20,getdate());-- 20 yrs nantrch ala select DATEADD(yy,-20,getdate());

select \*, DATEADD(mm,28,acct\_open\_date) as age from acct\_details where acct\_no=2;-- fakt 2 no chya acct lach 28 months add honar

--C)Date Part

select DATEPART(hh,getdate());-- atach time che hours milnar,tasach mins-sec-msec milnar

select \*, DATEpart(yy,acct\_open\_date) as yy from acct\_details --particular year milala select \* from acct\_details where acct\_open\_date like '%2022%'-- fakt 2022 che details

ale