-- To insert into Identity column

SET IDENTITY\_INSERT Employee ON

INSERT INTO Employee (Eid, Ename, Salary, DeptName) Values(1009,'Kiran', 3000, 'HR')

SET IDENTITY\_INSERT Employee OFF

INSERT INTO Employee ( Ename, Salary, DeptName) Values('Samyak', 3000, 'HR')

SELECT \* FROM Employee

DELETE Employee WHERE Eid=1010

OPERATORS

Arithemtic Opr + , -, / , \*

Relational <, <=, >,>=, <>

Logical AND, OR, NOT

Bitwise &, |, ^

Assignment/ Equality =

SET Opr UNION , UNION ALL, INTERSECT

SELECT 25 + 20

SELECT 25 - 20

SELECT 25 / 20

SELECT 25 % 20

SELECT \* FROM Employee WHERE (Eid <= 1001 AND Eid>=1009)

SELECT \* FROM Employee WHERE (Eid > 1001 AND Eid<1009)

SELECT \* FROM Employee WHERE (Eid <= 1001 OR Eid>=1009 OR Salary = 3000)

Bitwise Opr - Converts values into binary number format and perform Operation

Ex: SELECT 25&20

25 converted to binary , 20 converted to binary , both binary values multiplication result

Constraints

--------------------------------------

Constraints will enforce the business rules on the data

Helps to implement different integrity level

NOT NULL - will not allow null values into a column

DROP TABLE Employee

CREATE TABLE Employee(

Eid INT NOT NULL,

Ename varchar(100),

Salary INT ,

DeptId INT)

insert into Employee(Eid,Ename, Salary, DeptId) values(1002,'Rama', 6000,60)

insert into Employee(Eid,Ename, Salary, DeptId) values(1003,'Rakesh', 5000,60)

insert into Employee(Eid,Ename, Salary, DeptId) values(1002,'Rita', 2400,60)

-- Cannot insert the value NULL into column 'Eid', table 'HanuDB.dbo.Employee'; column does not allow nulls. INSERT fails.

Select \* from Employee

UNIQUE - Will not allows duplicate values into the column

DROP TABLE Employee

CREATE TABLE Employee(

Eid INT UNIQUE,

Ename varchar(100),

Salary INT ,

DeptId INT)

insert into Employee(Eid,Ename, Salary, DeptId) values(1002,'Rama', 6000,60)

insert into Employee(Eid,Ename, Salary, DeptId) values(1003,'Rakesh', 5000,60)

insert into Employee(Eid,Ename, Salary, DeptId) values(1002,'Rita', 2400,60)

-- Violation of UNIQUE KEY constraint 'UQ\_\_Employee\_\_C1971B5201E622DE'. Cannot insert duplicate key in object 'dbo.Employee'. The duplicate key value is (1002).

insert into Employee(Ename, Salary, DeptId) values('Rita', 2400,60)

insert into Employee(Ename, Salary, DeptId) values('Riya', 2200,60)

-- WIll accepst one null value into the columns

Select \* from employee

PRIMARY KEY

\* Will not allows null values into the column

\* will not allows duplicates into the columns

\* NOT NULL + UNIQUE + Some Extra

\* Only one primary key is allowed in a table

CREATE TABLE Department (

DeptId INT PRIMARY KEY,

DeptName VARCHAR(50)

)

CREATE TABLE Department (

DeptId INT CONSTRAINT PK\_Eid PRIMARY KEY,

DeptName VARCHAR(50),

)

DROP TABLE Department

CREATE TABLE Department (

DeptId INT,

DeptName VARCHAR(50),

CONSTRAINT PK\_DeptId PRIMARY KEY(DeptId)

)

FOREIGN KEY

Referential integrity

\* Will refer to the primary key column of a parent table

\* there can be n Foreign keys in a table

CREATE TABLE Employee(

Eid INT,

Ename VARCHAR(100),

Salary INT,

MngrId INT,

DeptId INT,

Gender char(5),

CONSTRAINT PK\_Eid PRIMARY KEY(Eid),

CONSTRAINT FK\_DeptID FOREIGN KEY(DeptId) REFERENCES Department(DeptId),

CONSTRAINT CHK\_Salary CHECK (Salary>=1000)

)

INSERT INTO Employee VALUES(1004,'Jack',6000, 1004,10,'M')

--The INSERT statement conflicted with the FOREIGN KEY constraint "FK\_DeptID". The conflict occurred in database "HanuDB", table "dbo.Department", column 'DeptId'.

SELECT \* FROM Department

INSERT INTO Department VALUES(10,'IT')

INSERT INTO Department VALUES(20,'HR')

INSERT INTO Department VALUES(30,'SALES')

CHECK

Implements Domain level integrity

validates the value with check constraint condition , if statisfied , allowed into the column

INSERT INTO Employee VALUES(1005,'Thompson',2000, 1004,10,'M')

DELETE FROM Employee WHERE Eid= 1006

SELECT \* FROM Employee

ALTER TABLE Employee

ADD CONSTRAINT CHK\_Salary CHECK (Salary>=1000)

INSERT INTO Employee VALUES(1006,'Welbaum', -2000, 1004,10,'F')

-- The INSERT statement conflicted with the CHECK constraint "CHK\_Salary". The conflict occurred in database "HanuDB", table "dbo.Employee", column 'Salary'.

ALTER TABLE Employee

DROP CONSTRAINT CHK\_Salary

Specifc Columns / Rows

-----------------------

select \* from Employee

Select Ename, Gender from Employee

Select Ename, Gender, Salary from Employee where Eid >=1008

Select MngrId from Employee

DISTINCT - Will eleimnates the duplicates in the result

---------------------------------------------------------------

Select DISTINCT mngrID from Employee

SELECT \* FROM Employee

SELECT \* FROM Employee WHERE DeptId= 10 AND Gender = 'M'

SELECT \* FROM Employee WHERE DeptId= 10 AND Gender = 'M' AND Salary >=3000

SELECT Ename, Salary FROM Employee WHERE DeptId= 10 AND Gender = 'M' AND Salary >=3000

INSERT into Employee(Eid,DeptId,Gender ) values(1012,10,'M')

IS - this "is" is specially designed to work with NULL VALUES

-----------------------------------------------------------------

SELECT \* FROM Employee WHERE Ename = NULL

SELECT \* FROM Employee WHere Ename IS NULL

SELECT \* FROM Employee WHere Ename IS NOT NULL

LIKE - Used to check the string patterns

Uses Wild Chars %, \_, ^

% - indicates any string

R% - Strat with R, Can be any thing later

%R - Any string in the begening , SHould with R

%R% - Any thing in the start , ANy thing at ENd , in the middle there should R

\_ indicates a single char

\_a% - Second char should be

\_ \_ a % -

Select \* from Employee WHERE Ename LIKE 'R%'

Select \* from Employee WHERE Ename NOT LIKE '\_\_t%'

SELECT \* FROM Employee WHERE Ename LIKE '%a'

IN - Specially used to compare in a list of values

------------------------------------------------------------------

SELECT \* FROM EMPLOYEE WHERE Gender IN('M','F')

Select \* from Employee WHere Eid IN (1010,1084,1093,1006,1002)

Select \* from Employee WHere Eid = 1010 OR Eid=1084 OR Eid=1093 OR Eid=1006 or Eid= 1002

BETWEEN - Used to Compare values in a range

-----------------------------------------------

SELECT \* FROM Employee Where Eid BETWEEN 1006 AND 1009

SELECT \* FROM Employee Where Salary BETWEEN 2000 AND 4000

ORDER BY - Used to sort Data

-----------------------------------

Select \* from Employee Order by Salary

\* By deafult it sorts in Ascending

Select \* from Employee Order by Salary desc

Select \* from Employee Order by Salary asc