check :-

=> use check constraint when rule based on conditions

syn :- CHECK(condition)

ex :- sal must be min 3000

```
CREATE TABLE emp11
(
  empno INT PRIMARY KEY,
  ename VARCHAR(10) NOT NULL,
  sal     MONEY  CHECK(sal>=3000)
)
```

INSERT INTO emp11 VALUES(100,'A',1000) => ERROR
INSERT INTO emp11 VALUES(101,'B',5000)
INSERT INTO emp11 VALUES(102,'C',NULL) => ACCEPTED

NOTE :- check constraint allows null values

ex 2 :- gender must be 'm','f'

```
gender  char(1)  check(gender IN ('m','f'))
```

ex 3 :- amt must be multiple of 100

```
amt  money  check(amt%100=0)
```

ex 4 :- pwd must be min 6 chars

```
pwd  varchar(12) check(len(pwd)>=6)
```

ex 5 :- emailid must contain '@'
must end with '.com' or '.co' or '.in'

```
emailid varchar(30)
      check(emailid like '%@%'
          and
          (
           emailid like '%.com'
           OR
```

```
                emailid like '%.co'
                OR
                emailid like '%.in'
                ))
```

foreign key :-

=> foreign key is used to establish relationship
between two tables.

=> to establish relationship take pk of one table
and add it to another table as fk and declare
with references constraint.

example :-

dept

dno(pk)	dname	loc
10	hr	hyd
20	it	blr

emp

empno	ename	sal	dno references dept(dno)
1	a	4000	10
2	b	3000	20
3	c	2000	90 => not accepted
4	d	3000	10
5	e	2000	null

=> values entered in fk column should match with
values entered in pk column.

=> fk allows duplicates and nulls.

=> after declaring fk a relationship is created
between two tables called parent/child relationship.

=> pk table is parent and fk table is child.

```
create table dept55
(
  dno     int primary key,
```

```
  dname  varchar(10) unique not null
)

insert into dept55 values(10,'hr'),(20,'it')

create table emp55
(
 empno int primary key,
 ename varchar(10) not null,
 sal   money check(sal between 5000 and 10000),
 dno   int references dept55(dno)
)

 insert into emp55 values(1,'A',5000,10)
 insert into emp55 values(2,'B',6000,90) => error
 insert into emp55 values(3,'C',6000,10)
 insert into emp55 values(4,'D',7000,NULL)
```

25-jul-23

Relationship Types :-
---

1 one to one (1:1)
2 one to many (1:m)
3 many to one (m:1)
4 many to many (m:n)

=> by default sql server creates one to many (1:m) relationship
   between two tables

how to establish 1:1 relationship :-

```
DEPT                          MGR
DNO   DNAME                         MGRNO       MNAME        DNO REFERENCES
DEPT(DNO)
10      HR                    100     A       10  UNIQUE
20      IT                    101     B       20
```

=> in the above example relationship between DEPT & MGR is one to one

=> to establish 1:1 relationship declare foreign key with unique

constraint.

```
CREATE TABLE dept88
(
 dno  int PRIMARY KEY,
 dname VARCHAR(10) UNIQUE NOT NULL
)

INSERT INTO dept88 VALUES(10,'HR'),(20,'IT')

CREATE TABLE mgr
(
  mgrno INT PRIMARY KEY,
  mname VARCHAR(10) NOT NULL,
  dno   INT REFERENCES dept88(dno)  UNIQUE
 )

 INSERT INTO mgr VALUES(1,'A',10)
 INSERT INTO mgr VALUES(2,'B',10) => ERROR
 INSERT INTO mgr VALUES(3,'C',90) => ERROR
```

How to establish many to many relationship :-

STUDENT		COURSE	
SID	SNAME	CID	CNAME
1	A	10	.NET
2	B	11	SQL

=> in the above example one student can register for many courses and one course can be taken by many students so relationship between student and course is many to many.

=> to establish many to many relationship create 3rd table and add primary keys of both tables as foreign keys.

REGISTRATIONS

SID	CID	DOR	FEE
1	10	?	?
1	11	?	?
2	10	?	?

```
 CREATE TABLE STUDENT
```

```
(
 SID INT PRIMARY KEY,
 SNAME VARCHAR(10) NOT NULL
 )

INSERT INTO STUDENT VALUES(1,'A'),(2,'B')

CREATE TABLE COURSE
(
  CID  INT  PRIMARY KEY,
  CNAME VARCHAR(10) NOT NULL
 )

INSERT INTO COURSE VALUES(10,'.NET'),(11,'SQL')

CREATE TABLE REGISTRATIONS
(
   SID  INT  REFERENCES STUDENT(SID) ,
   CID  INT  REFERENCES COURSE(CID),
   DOR  DATE,
   FEE  MONEY
 )

INSERT INTO REGISTRATIONS VALUES(1,10,GETDATE(),3000)
INSERT INTO REGISTRATIONS VALUES(1,11,GETDATE(),3000)
INSERT INTO REGISTRATIONS VALUES(2,10,GETDATE(),3000)
```

TABLE LEVEL :-

=> if constraints are declared after declaring all columns then it is called table level.

=> use table level to declare constraints for multiple or combination of columns .

```
 CREATE TABLE <tabname>
 (
   col1  datatype(size) ,
   col2  datatype(size),
   ------------------- ,
       constraint(col1,col2,---)
  )
```

Declaring check constraint at table level :-

PRODUCTS

PRODID	PNAME	PRICE	MFD_DT	EXP_DT	
100	A	100	2023-07-01	2023-01-01	INVALID

RULE :- EXP_DT > MFD_DT

```
CREATE TABLE PRODUCTS
(
  PRODID INT PRIMARY KEY,
  PNAME  VARCHAR(10) NOT NULL,
  MFD_DT DATE,
  EXP_DT DATE,
      CHECK(EXP_DT > MFD_DT)
 )
```

INSERT INTO PRODUCTS VALUES(100,'A','2023-07-01','2023-01-01') => ERROR

INSERT INTO PRODUCTS VALUES(100,'A','2023-01-01',GETDATE())