-- Constraints are restrictions that we put on the column values

-- PK ( unique & not null)

-- Check ( is used to define some range)

-- default

-- not null (mandatory)

-- unique (no duplicacy)

-- FK ( is used when we have more than 1 table)

use CTSPracticeDB

create table employee

(id int primary key,

name char(20) not null,

managername char(30) not null,

age int check (age between 20 and 40),

dept char(10) check (dept In ('HR','Accts','Sales')),

salary int check(salary > 10000) default 19000 ,

address char(50) unique)

insert into employee(id, name , address, managername , age, dept , salary)

values(1, 'Ajay', 'Delhi', 'Deepak', 34,'HR', 13000)

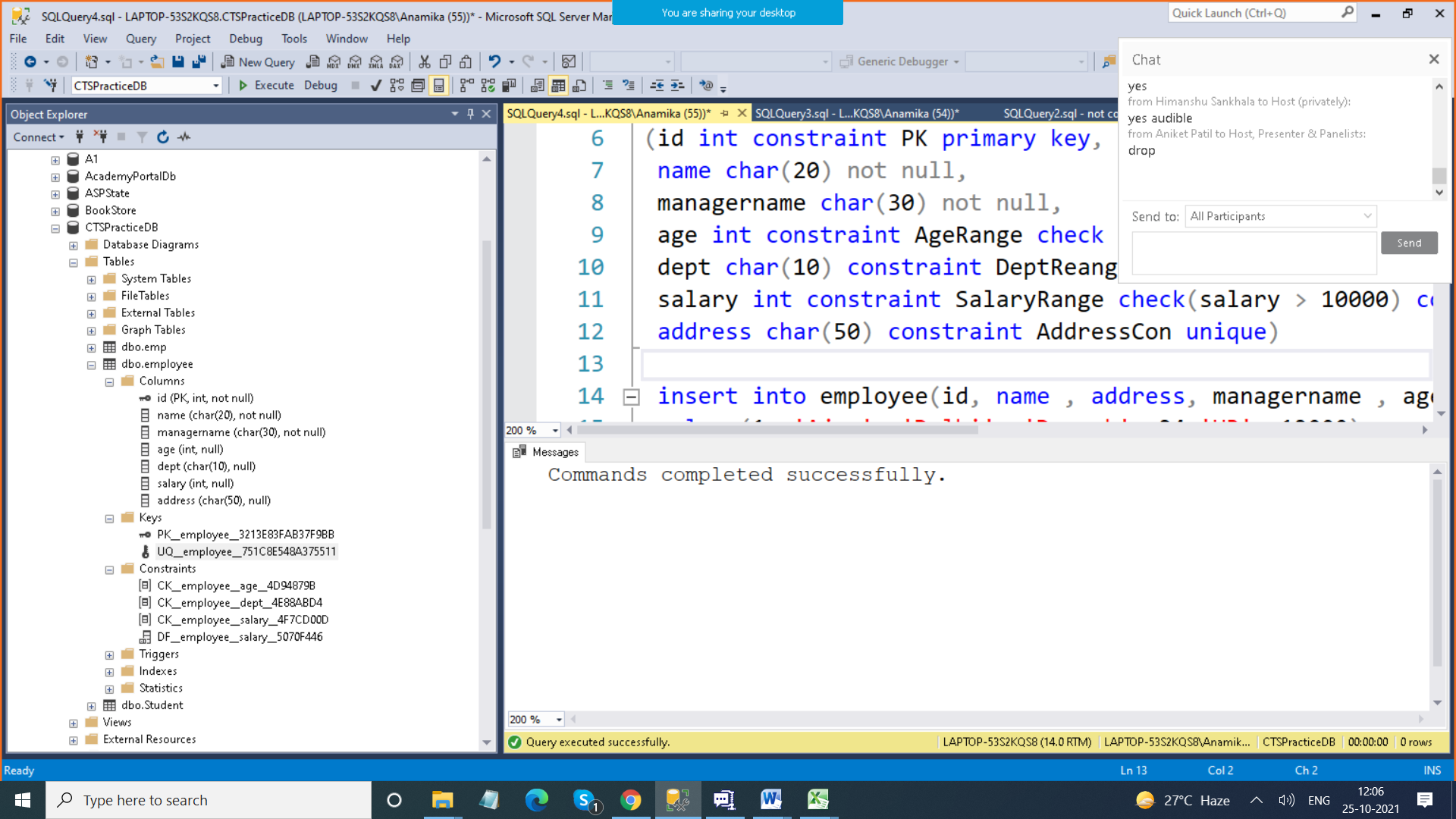
select \* from employee

insert into employee(id, name , address, managername , age, dept , salary)

values(3 , 'Sagar', 'A 90, New Delhi', 'Pradeep', 23,'HR', 13000)

insert into employee(id, name , address, managername , age, dept)

values(4 , 'Sagar', 'A 91, New Delhi', 'Pradeep', 23,'HR')



drop table employee

use CTSPracticeDB

create table employee

(id int constraint PK primary key,

name char(20) not null,

managername char(30) not null,

age int constraint AgeRange check (age between 20 and 40),

dept char(10) constraint DeptRange check (dept In ('HR','Accts','Sales')),

salary int constraint SalaryRange check(salary > 10000) constraint SalaryDef default 19000 ,

address char(50) constraint AddressCon unique)

alter table employee drop constraint DeptRange

select \* from STUDENT

delete from STUDENT where name is null

alter table student alter column name char(20) not null

alter table student add constraint MarksRange

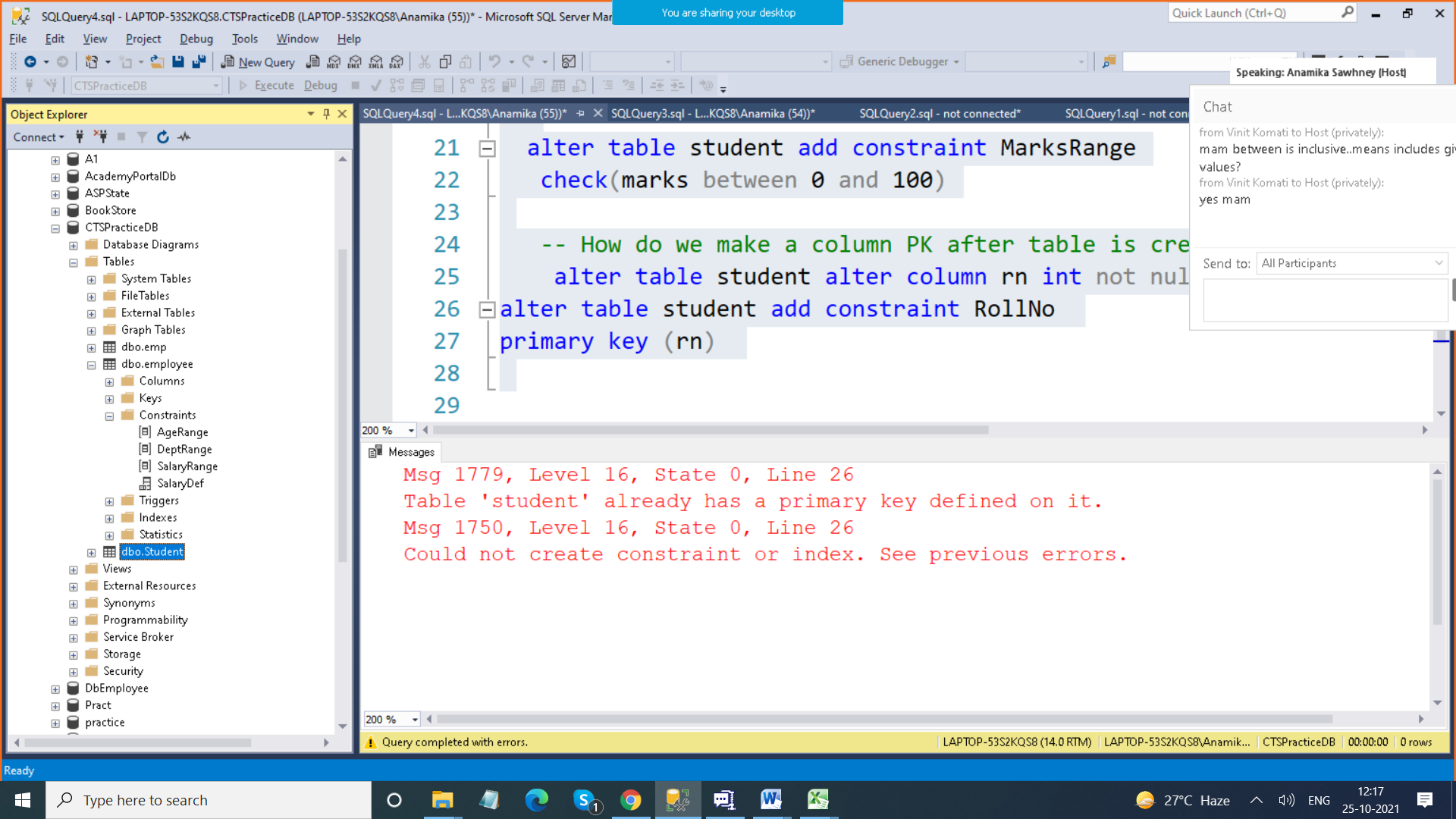
check(marks between 0 and 100)

-- How do we make a column PK after table is created

alter table student alter column rn int not null

alter table student add constraint RollNo

primary key (rn)



**JOINS**

Inner Join > It gives you matching Records

Outer Join > > It gives you matching as well as Non matching Records

1. Left Outer Join
2. Right Outer Join
3. Full Outer Join

Self Join > Table joined to itself

Cross Join > When you join two tables (THERE IS NO NEED TO HAVE A COLMMON COLUMN)

alter table employee drop column managername, dept

select \* from employee

create table department (id int primary key identity(100,2),

name varchar(20),

managername varchar(40)

)

select \* from department

insert into department(name, managername) values

( 'HR', 'Ajay'),

( 'Accts', 'Deepak'),

( 'Sales', 'Gagan'),

( 'Mktg', 'Maya'),

( 'Fin', 'Geeta')

-- Link this table with employee table

alter table employee add deptid int references department(id)

select \* from employee

update employee set deptid=1 where id=1

update employee set deptid=3 where id=2

update employee set deptid=5 where id=4

select \* from department

select \* from employee

-- Give employee name & their department names

-- INNER JOIN

select employee.name , department.name from

employee join department

on employee.deptid = department.id

select a.name , b.name from

employee a join department b

on a.deptid = b.id

-- Give me all employee names & their department details

select employee.name , department.name from

employee left outer join department

on employee.deptid = department.id

select employee.name , department.name from

employee right outer join department

on employee.deptid = department.id

select employee.name , department.name from

employee full outer join department

on employee.deptid = department.id

**Afternoon Session**

select \* from employee

select \* from department

insert into employee values(5,'Girish',35,14000, 'A 89 ,O Delhi', 2)

alter table employee add managerid int references employee (id)

update employee set managerid=1 where id IN(2,3)

-- give employee name & his manager name

-- self join

select a.name As "Employee Name" ,

b.name as "Manager Name"

from employee a join employee b

on a.managerid = b.id

-- Cross Join

create table course(id int , name varchar(20))

insert into course values

(1,'C#'),

(2,'C'),

(3,'C++'),

(4,'Java'),

(5,'PHP')

drop table batch

create table batch(id int, batchcode char(4),

startDate dateTime, duration int)

insert into batch values

(2,'B001', '10/12/2021',12),

(3,'B002', '11/22/2021',20),

(4,'B003', '10/30/2021',90)

select \* from course

select \* from batch

* - Cross Join

select \* from course

select \* from batch

select a.\* , b.\*

from course a cross join batch b

-- Copy records of one table to other table

select \* into coursenew from course

-- Copy structure of one table to other table

select \* into course2 from course

where 1=2

select \* from course2

select a.name , b.batchcode , b.startdate

into BatchDetails

from course a cross join batch b

* - String Functions

-- String Functions

select upper('ajay')

select lower('ajay')

select trim(' This ')

select len('This')

select len(' This ')

select len(trim(' This '))

select concat('This' ,' is' ,' my' ,' book')

select left('This is my Book', 3)

select right('This is my Book', 1)

select substring('This is my Book', 2,2)

select charindex('s','This is my Book')

<https://www.w3schools.com/sql/func_sqlserver_convert.asp>

Select upper(name) from course

select GetDate()

select month(GetDate())

select year(getDate())

select month(StartDate) from batch

select DATENAME(month ,StartDate) from batch

SELECT DATENAME(month, '2017/08/25')

select DATENAME(day ,StartDate) from batch

select DATENAME(WEEKDAY ,StartDate) from batch

SELECT DATEPART(yy, StartDate) from batch

select DATEADD(month, 3 , StartDate) from batch

select \* from batch

select id, batchcode , startDate from batch

select id, batchcode , Convert(varchar,startDate,1) from batch

select id, batchcode , Convert(varchar,startDate,2) from batch

select id, batchcode , Convert(varchar,startDate,3) from batch

select id, batchcode , Convert(varchar,startDate,103) from batch

Functions > Block of some statements

In Sql Server , functions

Inbuilt , User Defined Type

Inbuilt > which are pre defined

String Functions

Numeric Functions

DateTime Functions

General Functions

Functions

Scalar Functions & Aggregate Functions

Scalar Functions are the functions which takes 1 input and gives you output for that record (Single)

Len , upper , sqrt , trim

Aggregate Function also kn Group Functions which work upon group of records and thet give you single result

Sum count avg min max

select sum(salary) from employee

select min(salary) As "Min Salary",

max(salary) As "Max Salary",

sum(salary) As "Total Salary" ,

avg(salary) As "Average Salary" from employee

select count(\*) from employee