Assignment 15

--1. Create the tables Customersand Orderswith the following columns. ( do not declare the corresponding primary and foreign keys

Create table Customers (CustomerId char(5) not null,

CompanyName varchar(40) not null, contactName char(30) null,

Address Varchar(60) null, City char(15) null, Phone char(24) null,

Fax Char(24) null);

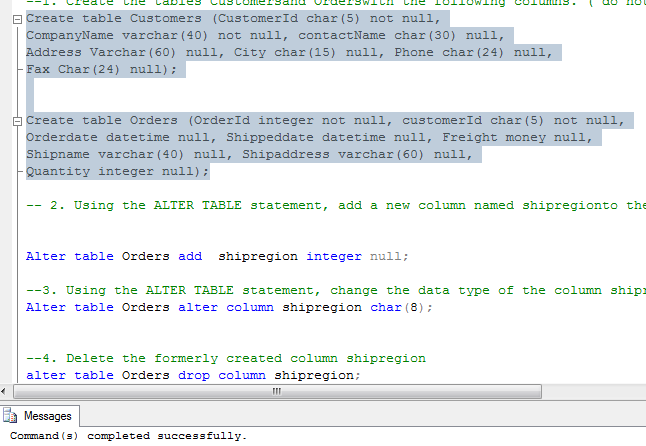
Create table Orders (OrderId integer not null, customerId char(5) not null,

Orderdate datetime null, Shippeddate datetime null, Freight money null,

Shipname varchar(40) null, Shipaddress varchar(60) null,

Quantity integer null);

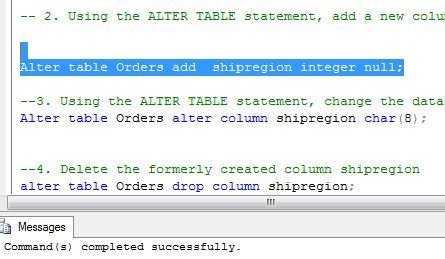
Output:



-- 2. Using the ALTER TABLE statement, add a new column named shipregionto the Orderstable. The fields should be nullable and contain integers

Alter table Orders add shipregion integer null;

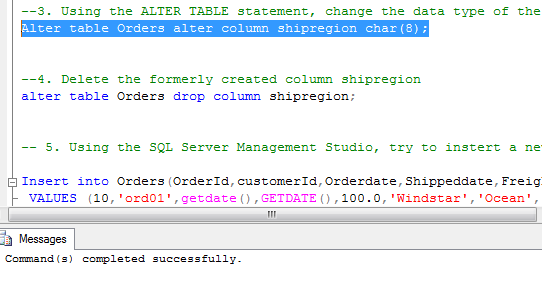
Output:



--3. Using the ALTER TABLE statement, change the data type of the column shipregion from INTEGER to CHARACTER with length 8. The fields may contain null values

Alter table Orders alter column shipregion char(8);

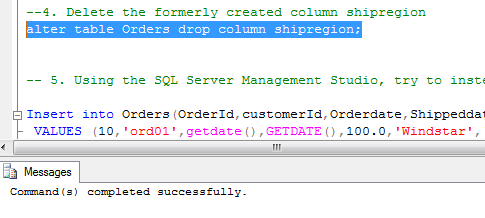
output



--4. Delete the formerly created column shipregion

alter table Orders drop column shipregion;

Output

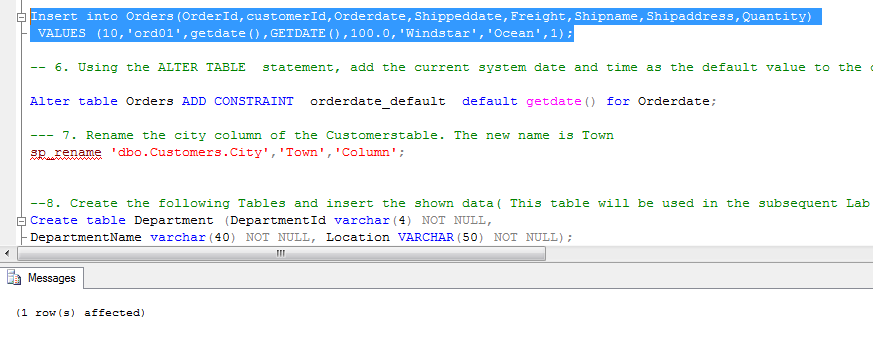


-- 5. Using the SQL Server Management Studio, try to instert a new row into the Orderstable with the following values:( 10, ‘ord01’, getdate(), getdate(), 100.0, ‘Windstar’, ‘Ocean’ ,1)

Insert into Orders(OrderId,customerId,Orderdate,Shippeddate,Freight,Shipname,Shipaddress,Quantity)

VALUES (10,'ord01',getdate(),GETDATE(),100.0,'Windstar','Ocean',1);

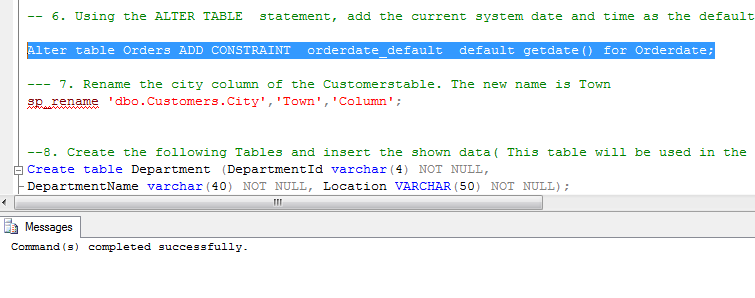
Output



-- 6. Using the ALTER TABLE statement, add the current system date and time as the default value to the orderdatecolumn of the Orderstable

Alter table Orders ADD CONSTRAINT orderdate\_default default getdate() for Orderdate;

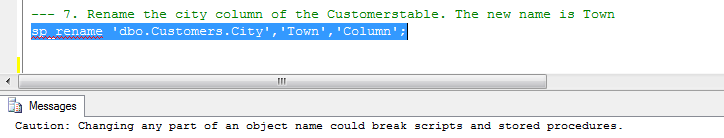
Output



--- 7. Rename the city column of the Customerstable. The new name is Town

sp\_rename 'dbo.Customers.City','Town','Column';

output



--8. Create the following Tables and insert the shown data( This table will be used in the subsequent Lab sessions

Create table Department (DepartmentId varchar(4) NOT NULL,

DepartmentName varchar(40) NOT NULL, Location VARCHAR(50) NOT NULL);

Insert into Department (DepartmentId, DepartmentName, Location) VALUES ('d1','Residency','Dallas');

Insert into Department (DepartmentId, DepartmentName, Location) VALUES ('d2','Accounts','Seattle');

Insert into Department (DepartmentId, DepartmentName, Location) VALUES ('d3','Managment','Dallas');

CREATE TABLE Employee (EmployeeId integer not null, emp\_fname Varchar(20) NOT NULL,

emp\_lname varchar(20) NOT NULL, dep\_id varchar(4) not null);

Insert into Employee (EmployeeId,emp\_fname,emp\_lname,dep\_id) VALUES (25348,'Matthew','Smith','d3');

Insert into Employee (EmployeeId,emp\_fname,emp\_lname,dep\_id) VALUES (10102,'Ann','Jones','d3');

Insert into Employee (EmployeeId,emp\_fname,emp\_lname,dep\_id) VALUES (18316,'John','Barrimor','d1');

Insert into Employee (EmployeeId,emp\_fname,emp\_lname,dep\_id) VALUES (25348,'James','James','d2');

Create table Project (Project\_no varchar(10) NOT NULL, Project\_name varchar(50) Not Null,

Budget money );

Insert into Project (Project\_no , Project\_name,Budget ) VALUES ('p1','Apollo',12000);

Insert into Project (Project\_no , Project\_name,Budget ) VALUES ('p2','Gemini',95000);

Insert into Project (Project\_no , Project\_name,Budget ) VALUES ('p3','Mercury',18560);

Create table Works\_on (emp\_no integer, Project\_no varchar(10), Job Varchar(30) NULL, enter\_date date);

SELECT \* FROM Works\_on

INSERT INTO Works\_on (emp\_no,Project\_no,Job,enter\_date) VALUES (10102,'p1','Analyst','10-01-1997');

INSERT INTO Works\_on (emp\_no,Project\_no,Job,enter\_date) VALUES (10102,'p3','Manager','01-01-1999');

INSERT INTO Works\_on (emp\_no,Project\_no,Job,enter\_date) VALUES (25348,'p2','Clerk','02-15-1998');

INSERT INTO Works\_on (emp\_no,Project\_no,Job,enter\_date) VALUES (18316,'p2',NULL,'06-01-1998');

INSERT INTO Works\_on (emp\_no,Project\_no,Job,enter\_date) VALUES (29346,'p2',NULL,'12-15-1997');

INSERT INTO Works\_on (emp\_no,Project\_no,Job,enter\_date) VALUES (2581,'p3','Analyst','10-15-1998');

INSERT INTO Works\_on (emp\_no,Project\_no,Job,enter\_date) VALUES (28559,'p1','Manager','04-15-1998');

INSERT INTO Works\_on (emp\_no,Project\_no,Job,enter\_date) VALUES (28559,'p2','Clerk','02-01-1992');

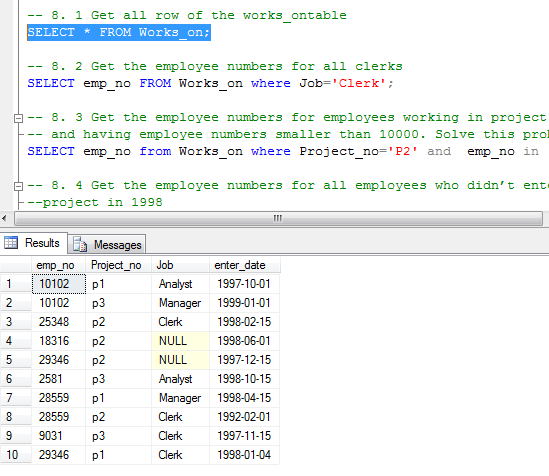
INSERT INTO Works\_on (emp\_no,Project\_no,Job,enter\_date) VALUES (9031,'p3','Clerk','11-15-1997');

INSERT INTO Works\_on (emp\_no,Project\_no,Job,enter\_date) VALUES (29346,'p1','Clerk','01-04-1998');

-- 8. 1 Get all row of the works\_ontable

SELECT \* FROM Works\_on;

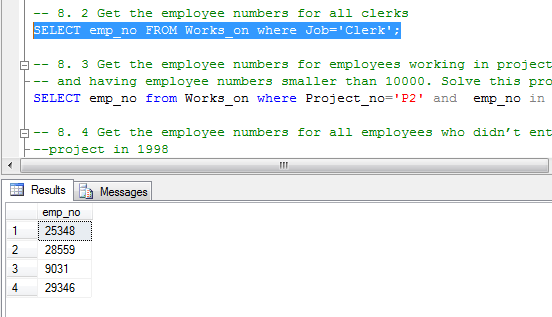
Output:



-- 8. 2 Get the employee numbers for all clerks

SELECT emp\_no FROM Works\_on where Job='Clerk';

Output:

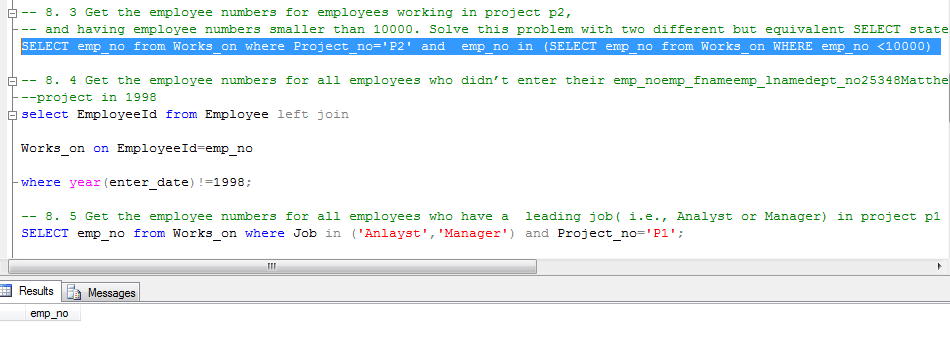


-- 8. 3 Get the employee numbers for employees working in project p2,

-- and having employee numbers smaller than 10000. Solve this problem with two different but equivalent SELECT statements

SELECT emp\_no from Works\_on where Project\_no='P2' and emp\_no in (SELECT emp\_no from Works\_on WHERE emp\_no <10000)

Output:



-- 8. 4 Get the employee numbers for all employees who didn’t enter their emp\_noemp\_fnameemp\_lnamedept\_no25348MatthewSmithd310102AnnJonesd318316JohnBarrimored129346JamesJamesd2

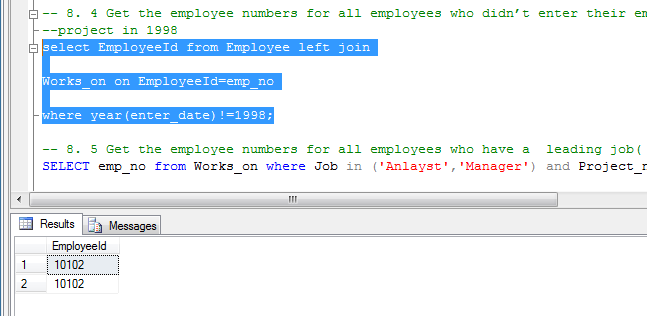
--project in 1998

select EmployeeId from Employee left join

Works\_on on EmployeeId=emp\_no

where year(enter\_date)!=1998;

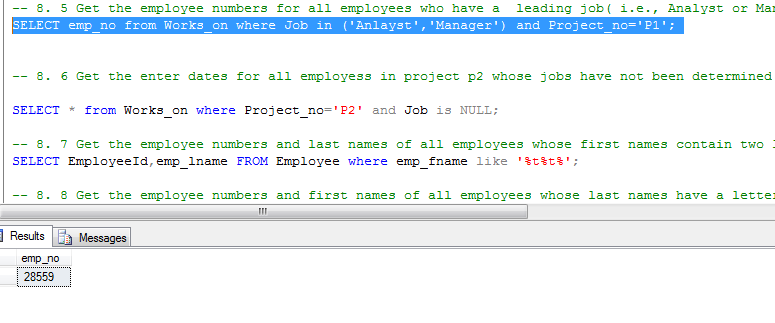
output:



-- 8. 5 Get the employee numbers for all employees who have a leading job( i.e., Analyst or Manager) in project p1

SELECT emp\_no from Works\_on where Job in ('Anlayst','Manager') and Project\_no='P1';

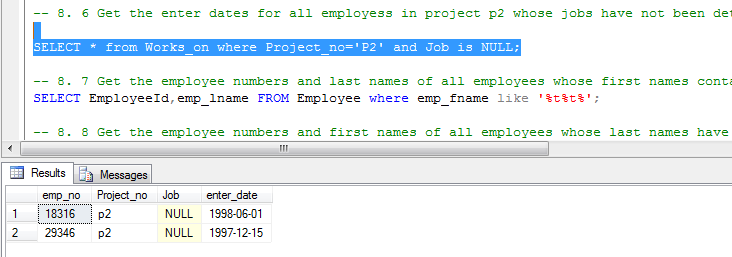
Output:



-- 8. 6 Get the enter dates for all employess in project p2 whose jobs have not been determined yet

SELECT \* from Works\_on where Project\_no='P2' and Job is NULL;

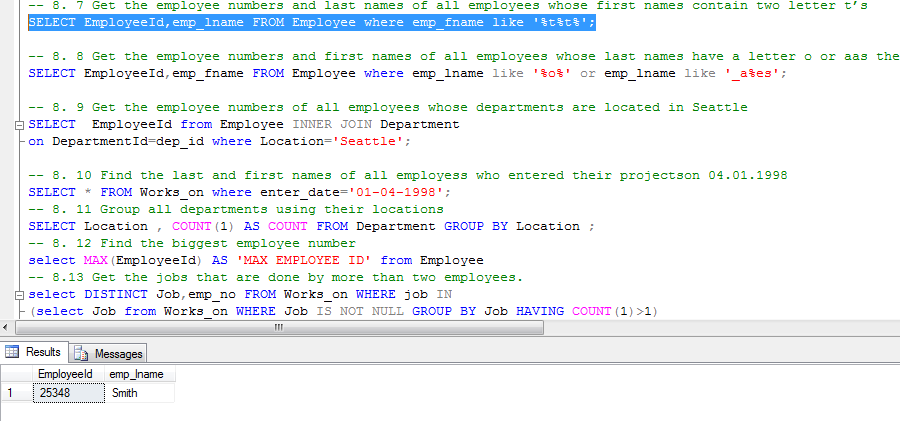
Output:



-- 8. 7 Get the employee numbers and last names of all employees whose first names contain two letter t’s

SELECT EmployeeId,emp\_lname FROM Employee where emp\_fname like '%t%t%';

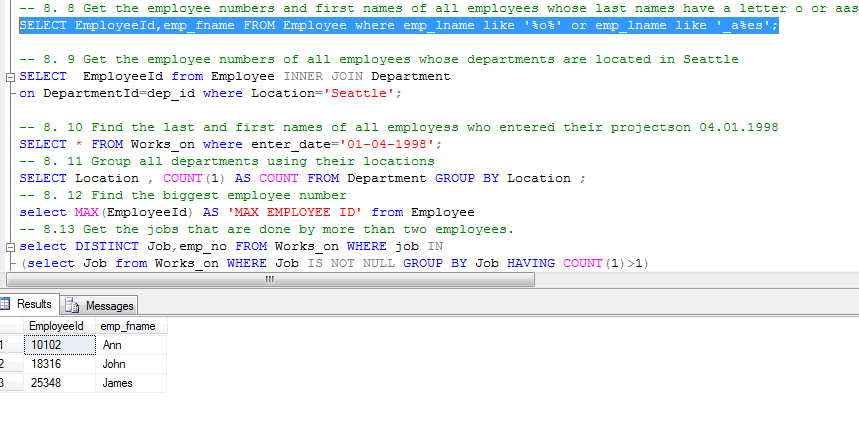
Output:



-- 8. 8 Get the employee numbers and first names of all employees whose last names have a letter o or aas the second character and end with the letters es

SELECT EmployeeId,emp\_fname FROM Employee where emp\_lname like '%o%' or emp\_lname like '\_a%es';

Output:

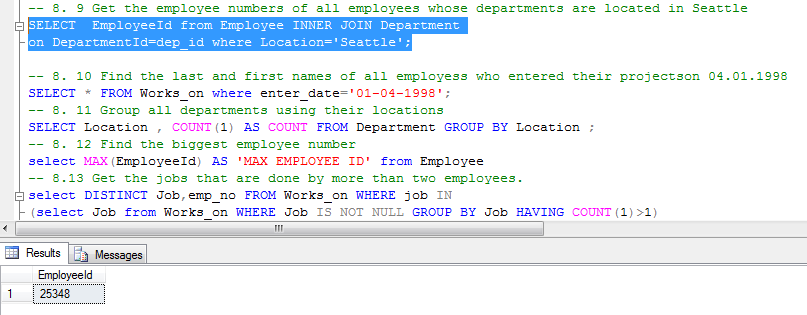


-- 8. 9 Get the employee numbers of all employees whose departments are located in Seattle

SELECT EmployeeId from Employee INNER JOIN Department

on DepartmentId=dep\_id where Location='Seattle';

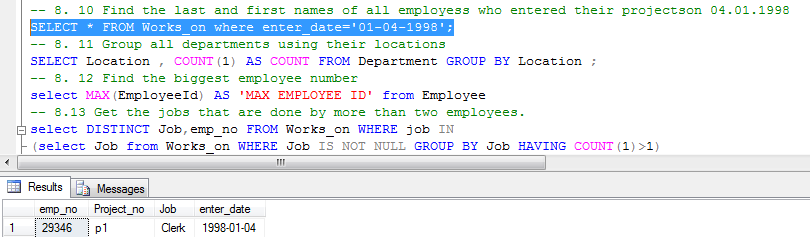
output:



-- 8. 10 Find the last and first names of all employess who entered their projectson 04.01.1998

SELECT \* FROM Works\_on where enter\_date='01-04-1998';

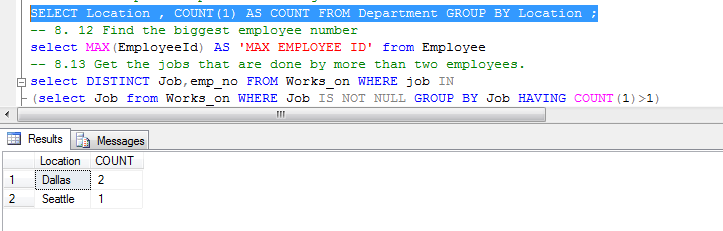
Output:



-- 8. 11 Group all departments using their locations

SELECT Location , COUNT(1) AS COUNT FROM Department GROUP BY Location ;

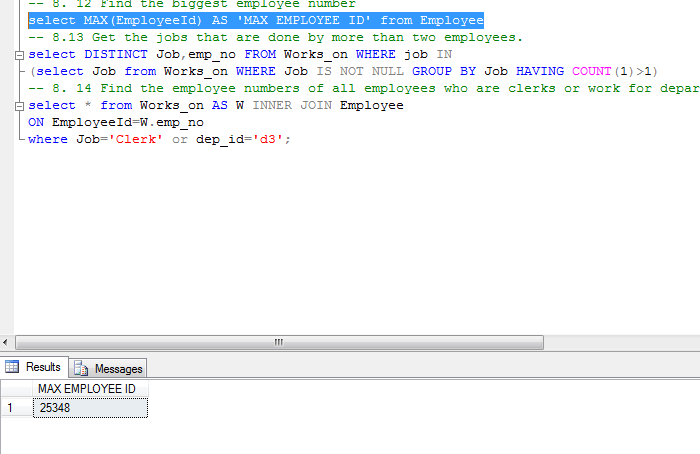
Output:



-- 8. 12 Find the biggest employee number

select MAX(EmployeeId) AS 'MAX EMPLOYEE ID' from Employee

Output:

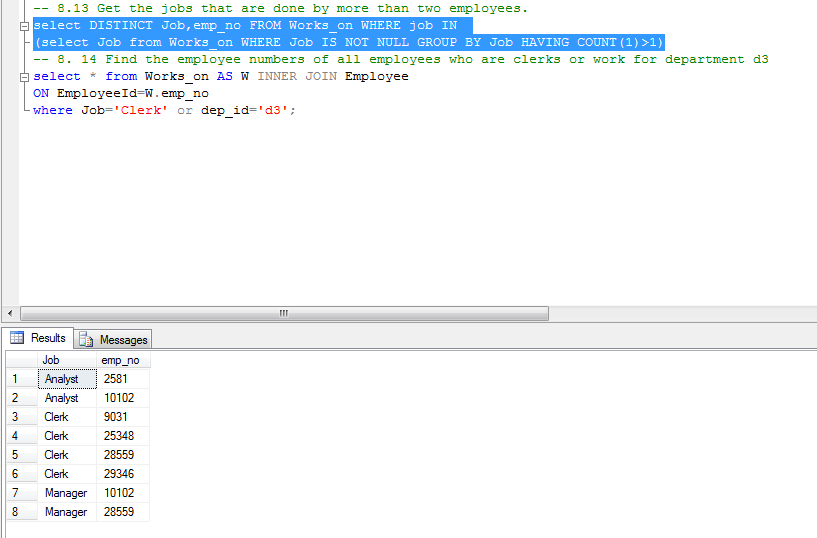


-- 8.13 Get the jobs that are done by more than two employees.

select DISTINCT Job,emp\_no FROM Works\_on WHERE job IN

(select Job from Works\_on WHERE Job IS NOT NULL GROUP BY Job HAVING COUNT(1)>1)

Output:



-- 8. 14 Find the employee numbers of all employees who are clerks or work for department d3

select \* from Works\_on AS W INNER JOIN Employee

ON EmployeeId=W.emp\_no

where Job='Clerk' or dep\_id='d3';

Output:

