1. Create the tables **Customers** and **Orders** with the following columns. ( do not declare the corresponding primary and foreign keys )

|  |
| --- |
| **Customers** |
| Customerid char(5) not null |
| CompanyNamevarchar(40) not null |
| contactName char(30) null |
| Address varchar(60) null |
| City char(15) null |
| Phone char(24) null |
| Fax char(24) null |

|  |
| --- |
| **Orders** |
| OrderId integer not null |
| customerId char(5) not null |
| Orderdatedatetime null |
| Shippeddatedatetime null |
| Freight money null |
| Shipnamevarchar(40) null |
| Shipaddresvarchar(60) null |
| Quantity integer null |

**Ans :** CREATE TABLE Customer(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 int not null,CustomerId char(5) not null,OrderDate datetime null,Shippeddatetime datetime null,Freight money null,ShipName varchar(40) null,Shipaddress varchar(60) null,Quantity int null);

1. Using the ALTER TABLE statement, add a new column named **shipregion** to the **Orders** table. The fields should be nullable and contain integers.

**Ans**: ALTER TABLE Orders ADD shipregion int null;

1. 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.

**Ans** :ALTER TABLE ORDERS MODIFY COLUMN shipregion char(8) null;

1. Delete the formerly created column **shipregion.**

**Ans :** ALTER TABLE DROP COLUMN shipregion;

1. Using the SQL Server Management Studio, try to instert a new row into the **Orders** table with the following values:

( 10, ‘ord01’, getdate(), getdate(), 100.0, ‘Windstar’, ‘Ocean’ ,1)

**Ans:** INSERT INTO Orders (10,’ord01’,getdate(),getdate(),100.0,’Windstar’,’Ocean’,1);

1. Using the ALTER TABLE statement, add the current system date and time as the default value to the **orderdate** column of the **Orders** table.

**Ans**: SET @current\_date= cast(now() as date);

ALTER TABLE Orders MODIFY COLUMN orderdate date @current\_date;

1. Rename the city column of the **Customers** table. The new name is **Town.**

**Ans :** ALTER TABLE Customers RENAME COLUMN City TO town;

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

**Department**

|  |  |  |
| --- | --- | --- |
| **Dept\_no** | **Dept\_name** | **location** |
| d1 | Research | Dallas |
| d2 | Accounting | Seattle |
| d3 | Marketing | Dallas |

**Employee**

|  |  |  |  |
| --- | --- | --- | --- |
| **emp\_no** | **emp\_fname** | **emp\_lname** | **dept\_no** |
| 25348 | Matthew | Smith | d3 |
| 10102 | Ann | Jones | d3 |
| 18316 | John | Barrimore | d1 |
| 29346 | James | James | d2 |

**Project**

|  |  |  |
| --- | --- | --- |
| **project\_no** | **project\_name** | **Budget** |
| p1 | Apollo | 120000 |
| p2 | Gemini | 95000 |
| p3 | Mercury | 185600 |

**Works\_on**

|  |  |  |  |
| --- | --- | --- | --- |
| **emp\_no** | **project\_no** | **Job** | **enter\_date** |
| 10102 | p1 | Analyst | 1997.10.1 |
| 10102 | p3 | manager | 1999.1.1 |
| 25348 | p2 | Clerk | 1998.2.15 |
| 18316 | p2 | NULL | 1998.6.1 |
| 29346 | p2 | NULL | 1997.12.15 |
| 2581 | p3 | Analyst | 1998.10.15 |
| 9031 | p1 | Manager | 198.4.15 |
| 28559 | p1 | NULL | 198.8.1 |
| 28559 | p2 | Clerk | 1992.2.1 |
| 9031 | p3 | Clerk | 1997.11.15 |
| 29346 | p1 | Clerk | 1998.1.4 |
|  |  |  |  |

**Ans:** CREATE TABLE Department(Dept\_no varchar(2) not null PRIMARY KEY,Dept char(20) not null,Location char(40));

CREATE TABLE Employee(emp\_no int not null,emp\_fname char(20),emp\_lname char(20),dept\_no varchar(2),FOREIGN KEY (dept\_no) REFERENCES Department(dept\_no));

CREATE TABLE Project(project\_no varchar(2) not null,Project\_name char(20),Budget int));

CREATE TABLE Works\_on(emp\_no int,project\_no int,job char(20),enter\_date date, FOREIGN KEY (emp\_no) REFERENCES Employee(emp\_no), FOREIGN KEY (project\_no) REFERENCES Project(project\_no));

Simple Queries

1. Get all row of the **works\_on** table.

**Ans** : SELECT \* FROM Works\_on;

1. Get the employee numbers for all clerks

**Ans** : SELECT emp\_no FROM works\_on WHERE Job=’Clerk’;

1. Get the employee numbers for employees working in project p2, and having employee numbers smaller than 10000.

**Ans** : SELECT emp\_no FROM works\_on WHERE project\_no=’p2’ and emp\_no <1000;

1. Get the employee numbers for all employees who didn’t enter their project in 1998.

**Ans** : SELECT emp\_no FROM works\_on WHERE enter\_date >1998-01-01 AND enter\_date< 1998-12-31;

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

**Ans:** SELECT emp\_no FROM works\_on WHERE project\_no=’p2’ AND (Job=’manager’ OR Job=’Analyst’);

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

**Ans**: SELECT enter\_date FROM works\_on WHERE project\_no=’p1’ AND Job IS NULL;

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

**Ans:** SELECT emp\_no,firstName FROM Employee where LEN(firstName)-LEN(REPLACE(firstName.’T’,’ ’))=2;

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

**Ans:** SELECT emp\_no,lastName FROM Employee WHERE RIGHT(lastName,2)=’es’ AND (LIKE ‘-[o]%’ OR LIKE ‘\_[a]%’);

1. Get the employee numbers of all employees whose departments are located in Seattle.

**Ans** : SELECT emp\_no FROM Employee INNERJOIN (SELECT dept FROM Department WHERE location=’Seattle’) ON Department.dept = Employee.dept;

1. Find the last and first names of all employess who entered their projects on 04.01.1998

**Ans:** SELECT emp\_no FROM Employee INNER JOIN (SELECT emp\_no FROM works\_on WHERE enter\_date=’1998-04-01’) ON Employee.emp\_no =Works\_on.emp\_no;

1. Group all departments using their locations.
2. Find the biggest employee number.

**Ans**: SELECT project\_no from (SELECT COUNT(emp\_no) AS count,project\_no FROM works\_on GROUP BY project\_no) WHERE count>2;

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

**Ans:** SELECT MAX(emp\_no) FROM Employee as MoreJobs;

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

**Ans** : SELECT emp\_no FROM Employee WHERE dept=’d3’ UNION SELECT emp\_no FROM works\_on WHERE JoB=’Clerk’;