

GC – Mock Assessment 6 – SQL and Databases – Franks Solution

```
-- #1 Create a database called CompanyDb. (Make sure to use it as well)
--   This is done using the Object Explorer in SSMS
--   a. Right click on "Databases" and choose new database
--   b. Enter 'CompanyDb' as the database name and click OK
--   c. Open the "Databases" item in Object Explorer and verify there is a database called 'CompanyDb'
--   d. Right click on "CompanyDb" add click on "New Query"
--   e. You should now have a query window to enter your SQL

-- #2 Construct a table called Department. Department has the following columns:
--
--   Id- int PK auto-increment
--   Name - nvarchar 25 NOT NULL
--   Hiring - bit
--   Location - nvarchar 50

-- Delete any existing version of the new table
Drop table if exists Department

-- Create the table with the columns as requested
Create table Department (
  Id      int      Identity, -- Identity makes the column and auto-incremented column.
                                -- ie. a unique value ia automatically generated
  Name    nvarchar(25) not null,
  Hiring  bit,              -- SQL Server does not have a Boolean type, so bit is used. 1=true, 0=false
  Location nvarchar(50)
)
```

GC – Mock Assessment 6 – SQL and Databases – Franks Solution

-- #3 Add the two departments from above to the table.

```
insert into Department
(Name, Hiring, Location)
Values('Technology', 1, '1570 Woodward Ave')
```

```
insert into Department
(Name, Hiring, Location)
Values('Shipping & Receiving', 0, 'Southfield')
```

```
select * from Department -- verify the departments were added correctly (optional)
```

-- #4 Construct a table named Employee. Employee has the following columns.

```
-- SSN - nchar 9 PK
-- DOB - DATE
-- Phone - nchar 10
-- FirstName - nvarchar 30 NOT NULL
-- LastName - nvarchar 30 NOT NULL
-- DepartmentId- This is a foreign key reference to Departments. Choose the appropriate data type.
```

-- Delete any existing version of the new table

```
drop table if exists Employee
```

-- Create the table with the columns as requested

```
Create table Employee (
SSN          nchar(9) primary key,
DOB          date,
phone        nchar(10),
FirstName    nvarchar(30) not null,
LastName     nvarchar(30) not null,
DepartmentId int
)
```

GC – Mock Assessment 6 – SQL and Databases – Franks Solution

```
-- #5 Add the 4 employees from above to the table.
--
-- Note: Since the DepartmentId is auto-generated, we don't know what it is
--       So we use a SELECT to the the DepartmentId that was generated
--
insert into Employee
(SSN, DOB, Phone, FirstName, LastName, DepartmentId)
Values('123456789', '1/5/1975', '313555505', 'Charles', 'Charleston', (select id from department where Name = 'Shipping &
Receiving'))

insert into Employee
(SSN, DOB, Phone, FirstName, LastName, DepartmentId)
Values('987654321', '5/2/1990', '2489995555', 'Jimmy', 'Scrambles', (select id from department where Name = 'Technology'))

insert into Employee
(SSN, DOB, Phone, FirstName, LastName, DepartmentId)
Values('456789123', '8/6/2001', '7895555525', 'Vince', 'Jabowski', (select id from department where Name = 'Technology'))

insert into Employee
(SSN, DOB, Phone, FirstName, LastName, DepartmentId)
Values('654854632', '4/6/1988', '3139721400', 'John', 'Johnston', (select id from department where Name = 'Shipping &
Receiving'))

select * from Employee -- verify the employees were added correctly (optional)
```

GC – Mock Assessment 6 – SQL and Databases – Franks Solution

```
-- #6 Update the youngest employee to have a LastName of 'Spiderman'.
--
-- Note: Since we don't know what DOB is the largest,
--       We use a subquery to get the value from database to use in the WHERE
update employee
    Set LastName = 'Spiderman'
--where DOB = '8/6/2001' -- Not a good choice as data WILL change over time
where dob = (select max(dob) from employee)
```

```
Select * from Employee -- verify the update worked correctly (optional)
```

```
-- #7 Select all employees with the LastName that starts with J
select * from Employee
where LastName like 'J%'
```

```
-- #8 Remove the oldest employee
-- Note: Since we don't know what DOB is the smallest
--       We use a subquery to get the value from database to use in the WHERE
delete from employee
--where dob = '1/5/1975' -- Not a good choice as data WILL change over time
where dob = (select min(dob) from employee)
```

```
select * from employee -- verify the delete worked correctly (optional)
```

```
-- #9 Select all employees with a 313 area code
select *
    from employee
where phone like '313%'
```

```
-- #10 Select all employees that have a DOB before 9/9/1999
select *
    from employee
where dob > '9/9/1999'
```

GC – Mock Assessment 6 – SQL and Databases – Franks Solution

```
-- #11 Update the employees with DOB after 2000 to have no phone number
-- NULL represents missing or unknown values
Update employee
    set phone = null
where year(dob) > 2000

-- #12 Select all employees that do not have a phone number
--
-- Note: = and != does not work with nulls
--       Use IS NULL or IS NOT NULL instead
select * from employee where phone is null

-- #13 Use one statement to list all employees along with the name and location of the department the employee belongs to
--
-- Note: We need columns from multiple tables
--       We want all Employee table columns but only two columns from Department table
select employee.*, department.name, location
from employee
    inner join
    department
on employee.DepartmentId = Department.Id

-- #14 Delete both tables and all their contents from the database
Drop table employee
Drop table department

-- You can verify the DROPs worked by looking in Object Explorer to see if the tables are gone.
```