**IST 659 Lab 7 SQL III**
**Instructions**

**Business Case (this is the same case you have worked on for labs 4, 5 and 6)**

Career Services of our school wants to keep a track of all interviews (positions, candidates, company, and interviewer) that take place. They want to keep track of all the companies, the potential candidates, interviewer, positions available at companies etc. Sometimes the career services needs to contact the companies for verification or other inquiries. We need to build a database that would assist the career services in recording this information.

In this database system, each company and candidate will have their own profiles which include their names and contact information such as phone numbers, postal addresses. Candidates would need to provide information about their primary Experience domain, and relevant experience. Interviewers may or may not provide their office hour information. The schedule or office hour information should be a text describing when the interviewer's office is open, e.g. 9am-5pm Monday – Thursday.

An interviewer can conduct one or multiple interviews of candidates. A candidate can have one or more interviews. Each interview must have an interview date and round number along with information about the Candidate (CandidateId) and Interviewer (InterviewerId).

The database should also maintain information about the positions a company is looking to hire for. Details about position level, and position name should also be given. Information about whether the position is still available or not should also be stored in the database (this field will be either "yes" or "no").

In this lab we have already created the ERD model for the career services database (see below).

**Lab 7 instruction**

Imagine you are hired to design a new database to support this platform. In lab 4 you have created and populated the tables. In lab 5 you have revised the tables and written queries to answer easy data questions. In lab 6 you wrote complex queries to join two or more tables to answer data questions. In this lab you are also going to create views and transactions.

Now, please write SQL statements to finish the following tasks:

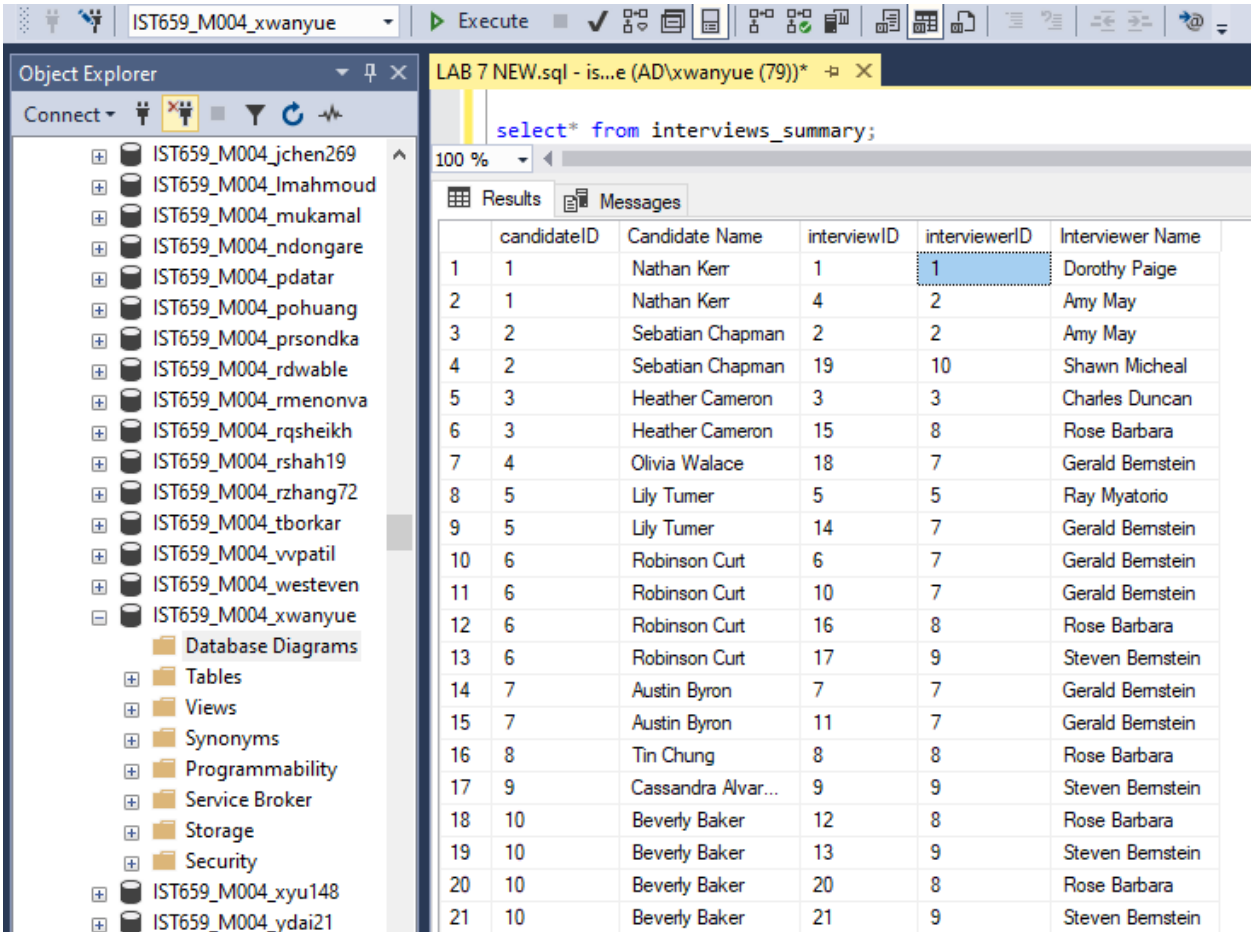
1. Views: create views based on the following requirements

    a) interviews_summary: for all the candidates, find their interviews and the corresponding interviewers. This view should show the candidate ID, candidate name, interview ID, interviewer ID, and interviewer name.

```
/* Using left join to find out all the candidates and their interview records */
create view interviews_summary as
SELECT c.candidateID, concat(c.CFName, ' ', c.CLName) as "Candidate Name",
i.interviewID, iw.interviewerID, concat(iw.IFName, ' ', iw.ILName) as "Interviewer
Name"
from candidates c
left join interviews i
```

```
on c.candidateID = i.candidateID
left join interviewers iw
on i.interviewerID = iw.interviewerID;

select* from interviews_summary;
```

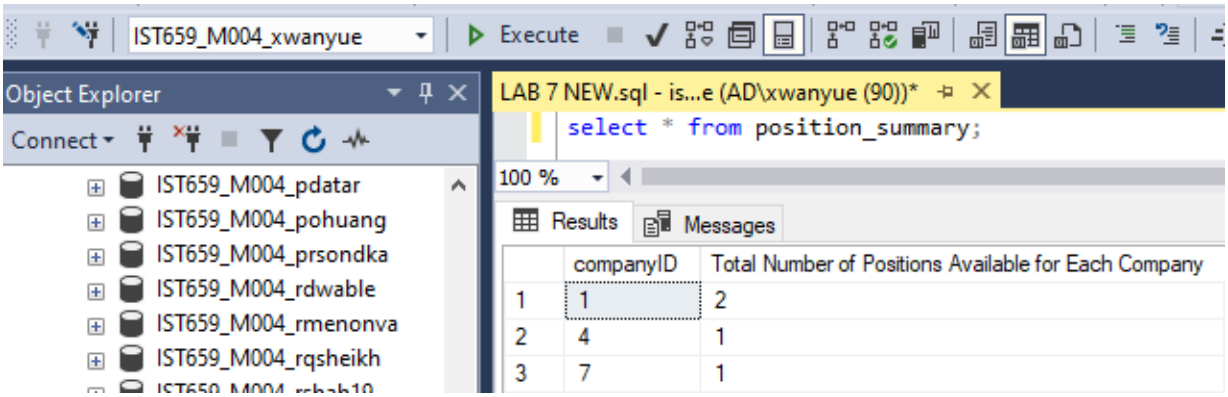



| | candidateID | Candidate Name | interviewID | interviewerID | Interviewer Name |
|---|---|---|---|---|---|
| 1 | 1 | Nathan Kerr | 1 | 1 | Dorothy Paige |
| 2 | 1 | Nathan Kerr | 4 | 2 | Amy May |
| 3 | 2 | Sebatian Chapman | 2 | 2 | Amy May |
| 4 | 2 | Sebatian Chapman | 19 | 10 | Shawn Micheal |
| 5 | 3 | Heather Cameron | 3 | 3 | Charles Duncan |
| 6 | 3 | Heather Cameron | 15 | 8 | Rose Barbara |
| 7 | 4 | Olivia Walace | 18 | 7 | Gerald Bernstein |
| 8 | 5 | Lily Turner | 5 | 5 | Ray Myatorio |
| 9 | 5 | Lily Turner | 14 | 7 | Gerald Bernstein |
| 10 | 6 | Robinson Curt | 6 | 7 | Gerald Bernstein |
| 11 | 6 | Robinson Curt | 10 | 7 | Gerald Bernstein |
| 12 | 6 | Robinson Curt | 16 | 8 | Rose Barbara |
| 13 | 6 | Robinson Curt | 17 | 9 | Steven Bernstein |
| 14 | 7 | Austin Byron | 7 | 7 | Gerald Bernstein |
| 15 | 7 | Austin Byron | 11 | 7 | Gerald Bernstein |
| 16 | 8 | Tin Chung | 8 | 8 | Rose Barbara |
| 17 | 9 | Cassandra Alvar... | 9 | 9 | Steven Bernstein |
| 18 | 10 | Beverly Baker | 12 | 8 | Rose Barbara |
| 19 | 10 | Beverly Baker | 13 | 9 | Steven Bernstein |
| 20 | 10 | Beverly Baker | 20 | 8 | Rose Barbara |
| 21 | 10 | Beverly Baker | 21 | 9 | Steven Bernstein |

b) position_summary: find the total number of positions available for each company in the second half of 2013 (Jul, Aug, Sep, Oct, Nov, Dec).

```
/* Using format function to abstract the year and month value;
   In the subquery, it gets result of distinct ID position which satisfies the
requirements (in the second half of 2013)
   The count() function will calculate the number of times that each company
appears in the positions table */
create view position_summary as
select p.companyID, count(p.companyID) as "Total Number of Positions Available for
Each Company"
from positions p
where p. positionavailable = 'yes' and p. positionID in
(select distinct positionID
 from interviews
 where format(interviewDate, 'yyyy') = 2013 and format(interviewDate, 'MM') in
('07','08','09','10','11','12'))
group by p.companyID;
```

```
select * from position_summary;
```

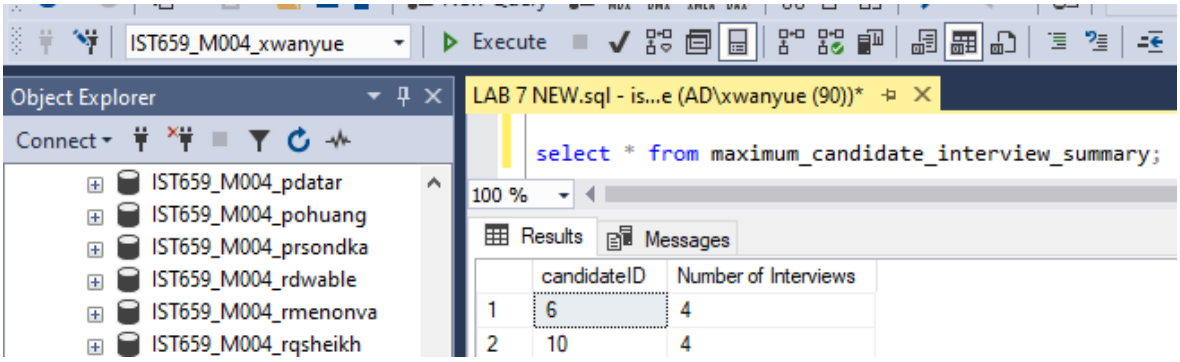



c) maximum_candidate_interview_summary: find the candidate who had the largest number of interviews.

```
/* In the subquery, it gets the result of ID of candidate who has the highest
number of interview*/

create view maximum_candidate_interview_summary as
select candidateID, count(candidateID) as "Number of Interviews"
from interviews
group by candidateID
having count(candidateID) = (select top 1 count(candidateID) from interviews group
by candidateID order by count(candidateID) desc);

select * from maximum_candidate_interview_summary;
```

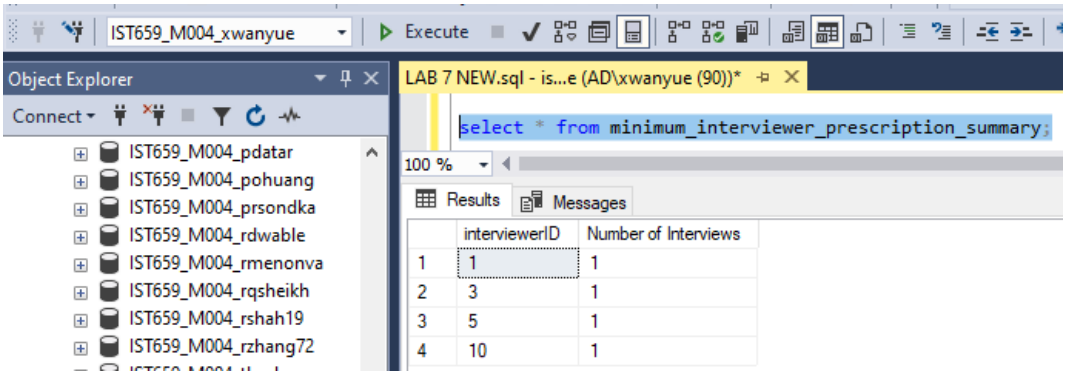



d) minimum_interviewer_prescription_summary: find the interviewer who had the lowest number of interviews.

```
/* Similar with question c */

create view minimum_interviewer_prescription_summary as
select interviewerID, count(interviewerID) as "Number of Interviews"
from interviews
group by interviewerID
having count(interviewerID) = (SELECT top 1 count(interviewerID) FROM interviews
group by interviewerID order by count(interviewerID));
```

```
select * from minimum_interviewer_prescription_summary;
```

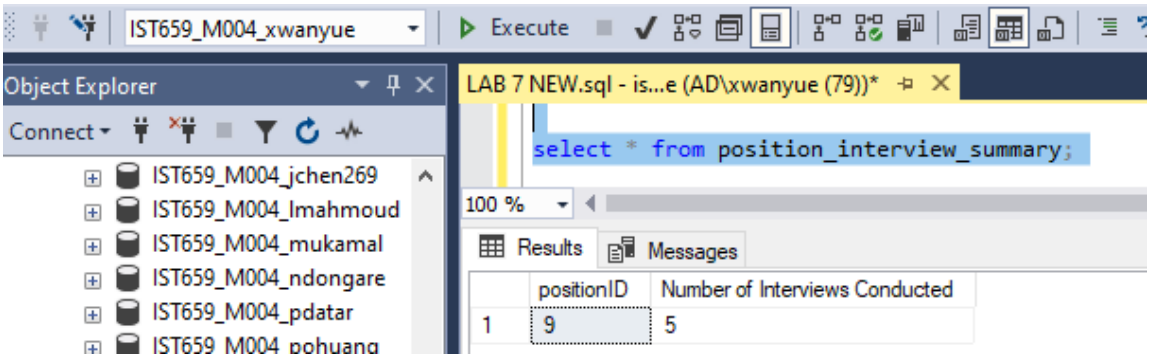



e) position_interview_summary: find the position for which the largest number of interviews were conducted.

```
/* similar with question c and question d.*/

create view position_interview_summary as
select positionID, count(positionID) as "Number of Interviews Conducted"
from interviews
group by positionID
having count(positionID) = (select top 1 count(positionID) from interviews group
by positionID order by count(positionID) desc);

select * from position_interview_summary;
```



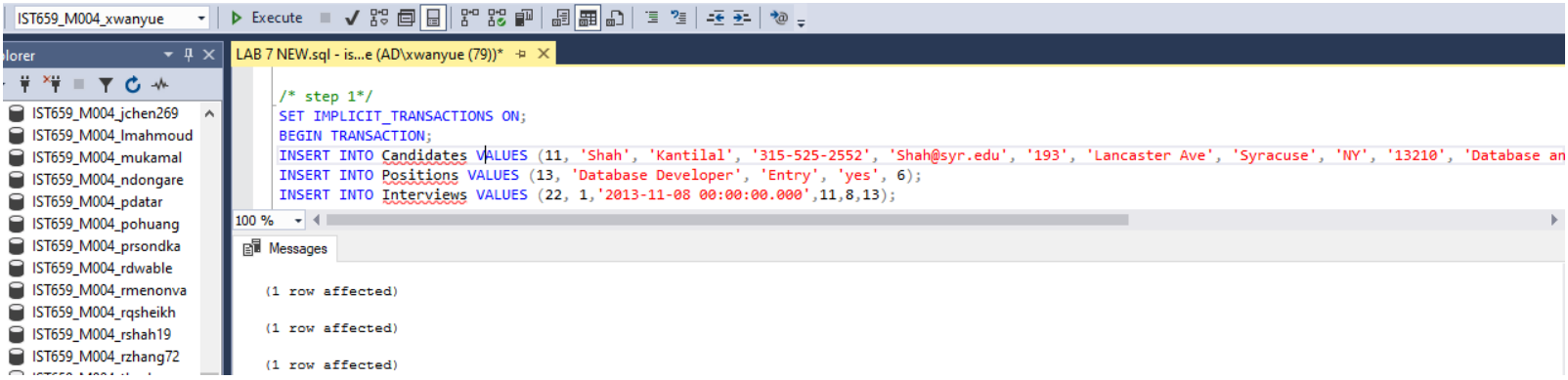

2. Create two transactions:

a) Create a new interview for a new candidate. New rows should be inserted into the tables Candidate, Interview, and Position.

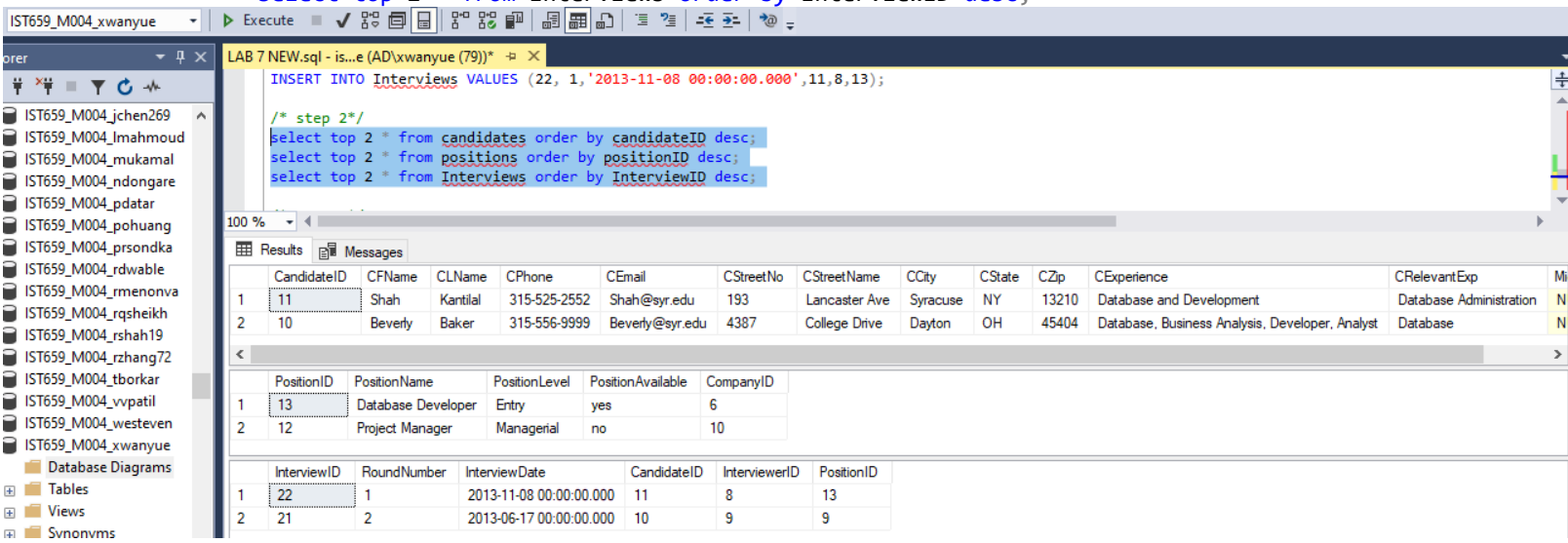
Step 1: create a new candidate record whose first name is “Shah”, last name is “Kantilal”, and phone number is “315-525-2552”. His address is “193 Lancaster Ave, Syracuse, New York, 13210”. Candidate experience is in “Database and Development” and Relevant Experience is in “Database Administration”. He was interviewed on “2013-11-08 00:00:00.000” for his 1st round of interview for “Database Developer” position at Google. Position level is “Entry” and position is available. His interviewer was “Rose Barbara”.
Step 2: display records from Candidate, Interview, and Position table.

Step 3: ROLLBACK transaction.
Step 4: display records from Candidate, Interview, and Position table.

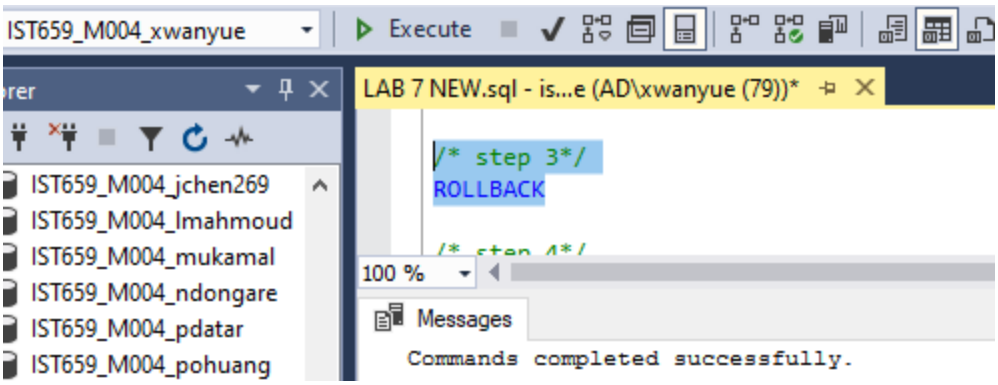
```
/* step 1*/
SET IMPLICIT_TRANSACTIONS ON;
BEGIN TRANSACTION;
INSERT INTO Candidates VALUES (11, 'Shah', 'Kantilal', '315-525-2552',
'Shah@syr.edu', '193', 'Lancaster Ave', 'Syracuse', 'NY', '13210', 'Database and
Development','Database Administration', NULL);
INSERT INTO Positions VALUES (13, 'Database Developer', 'Entry', 'yes', 6);
INSERT INTO Interviews VALUES (22, 1,'2013-11-08 00:00:00.000',11,8,13);
```



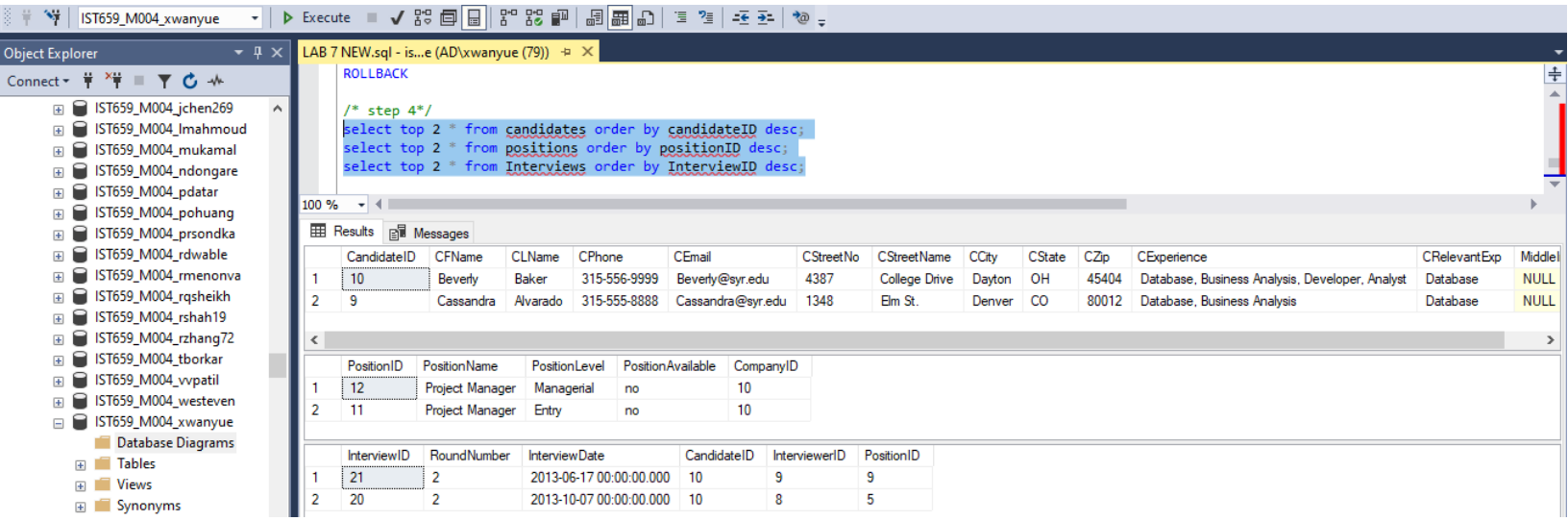

```
/* step 2*/
select top 2 * from candidates order by candidateID desc;
select top 2 * from positions order by positionID desc;
select top 2 * from Interviews order by InterviewID desc;
```




```
/* step 3*/
ROLLBACK
```




```
/* step 4*/
select top 2 * from candidates order by candidateID desc;
select top 2 * from positions order by positionID desc;
select top 2 * from Interviews order by InterviewID desc;
```

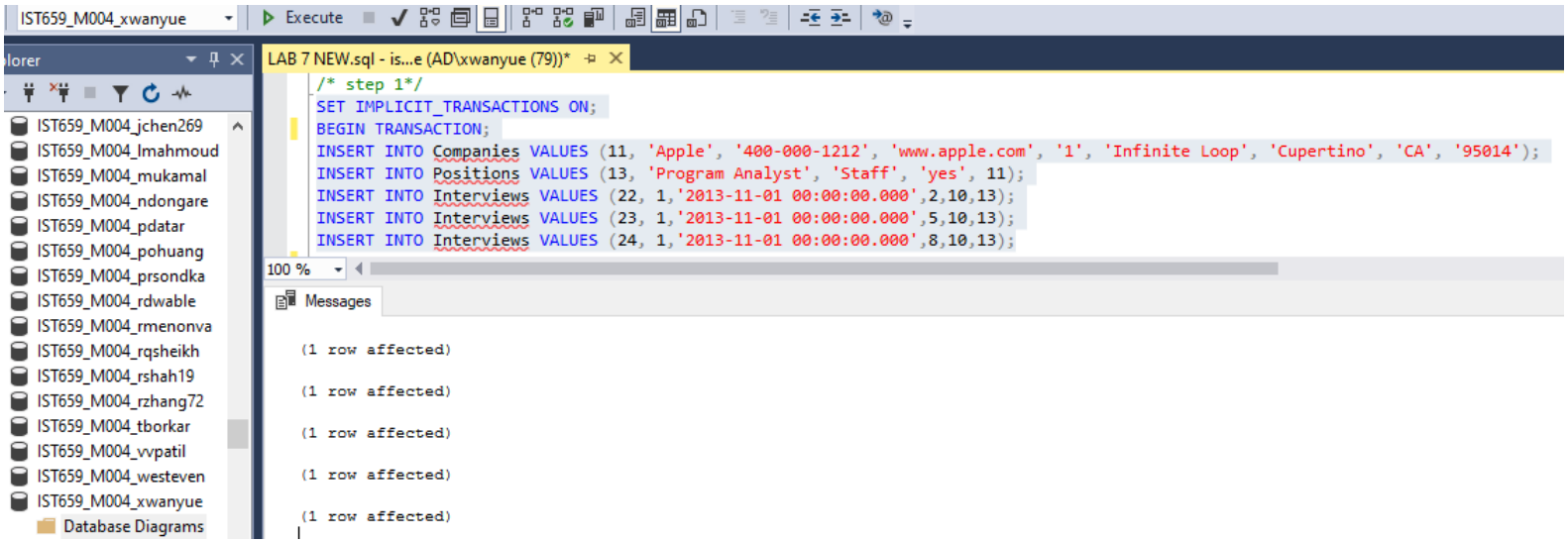



b) Add a new position for a company. New rows should be inserted into the tables Company, Position and Interview.

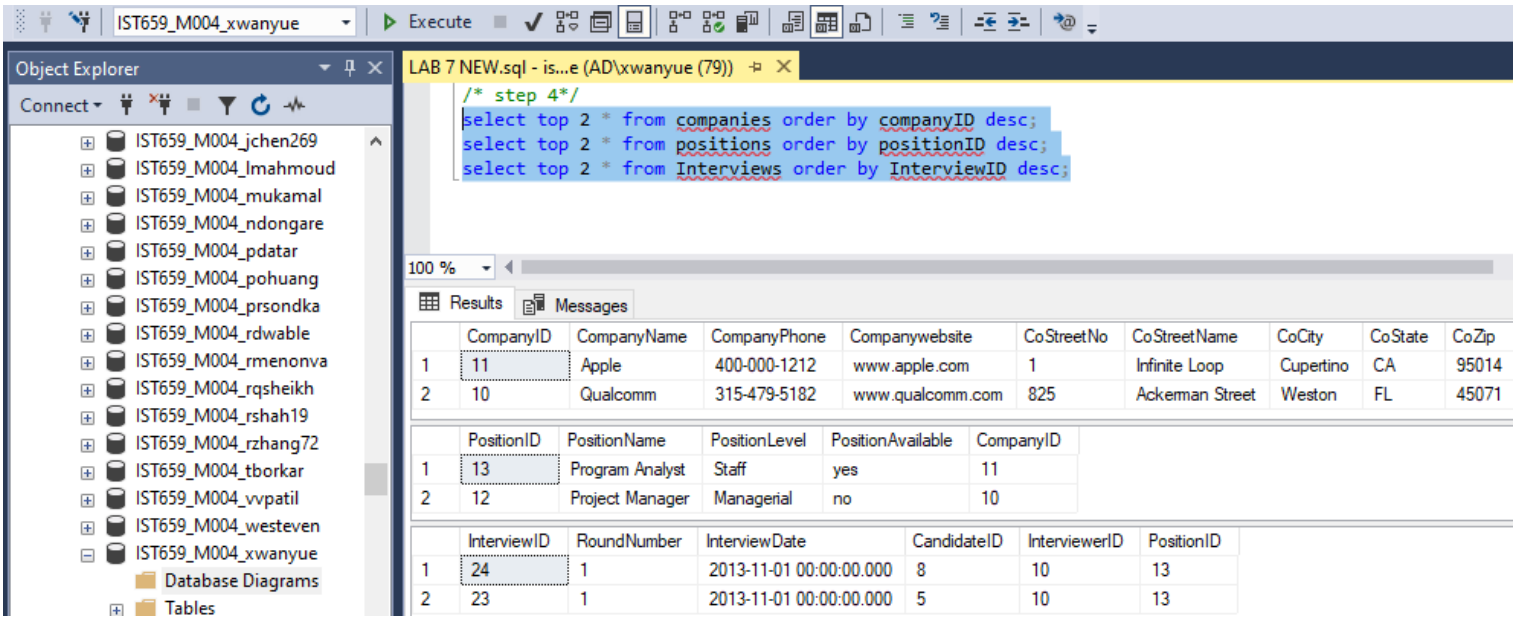
Step 1: create a new record for the company “Apple”. The company has a website URL www.apple.com and company phone is “400-000-1212”. Company Headquarters’ address is “1 Infinite Loop Cupertino, CA 95014”. The position they are hiring for is “Program Analyst” which is a “Staff” position. The 1st round of interviews were conducted for 3 candidates on “2013-11-01 00:00:00.000”. The 3 candidates were “Sebastian Chapman”, “Lily Turner” and “Tin Chung”. Also the interviewer was “Shawn Micheal”
Step 2: display records from Company, Position and Interview table.
Step 3: COMMIT transaction.
Step 4: display records from Company, Position and Interview table.

```
/* step 1*/
SET IMPLICIT_TRANSACTIONS ON;
BEGIN TRANSACTION;
```

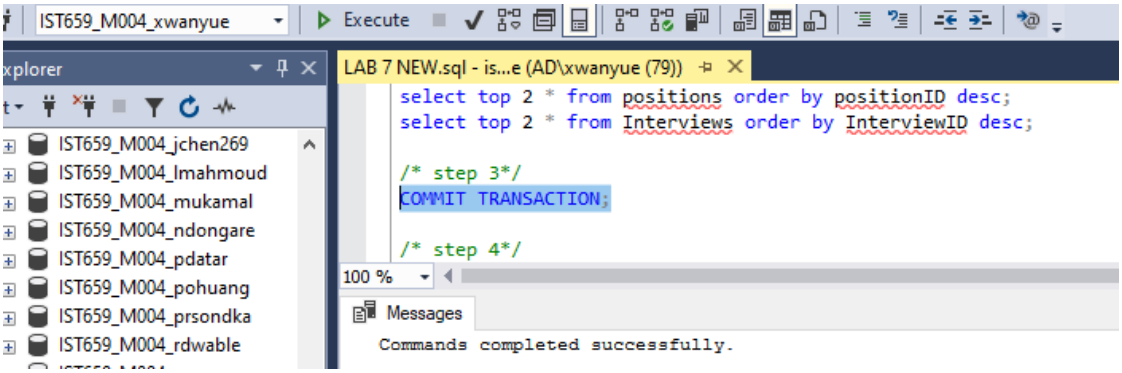
```
INSERT INTO Companies VALUES (11, 'Apple', '400-000-1212', 'www.apple.com', '1',
'Infinite Loop', 'Cupertino', 'CA', '95014');
INSERT INTO Positions VALUES (13, 'Program Analyst', 'Staff', 'yes', 11);
INSERT INTO Interviews VALUES (22, 1,'2013-11-01 00:00:00.000',2,10,13);
INSERT INTO Interviews VALUES (23, 1,'2013-11-01 00:00:00.000',5,10,13);
INSERT INTO Interviews VALUES (24, 1,'2013-11-01 00:00:00.000',8,10,13);
```



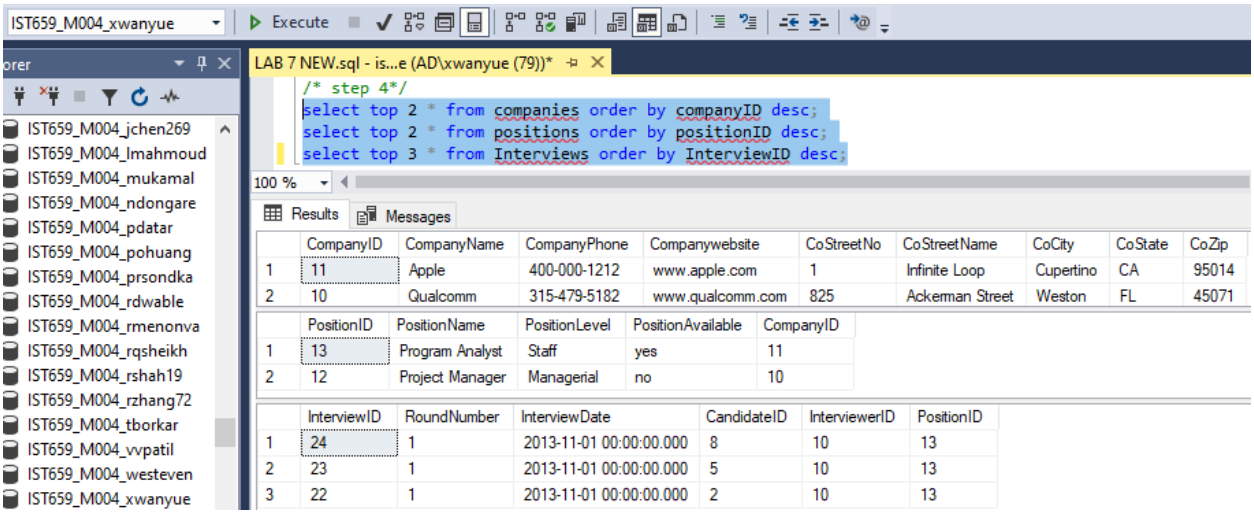

```
/* step 2*/
select top 2 * from companies order by companyID desc;
select top 2 * from positions order by positionID desc;
select top 2 * from Interviews order by InterviewID desc;
```




```
/* step 3*/
COMMIT TRANSACTION;
```

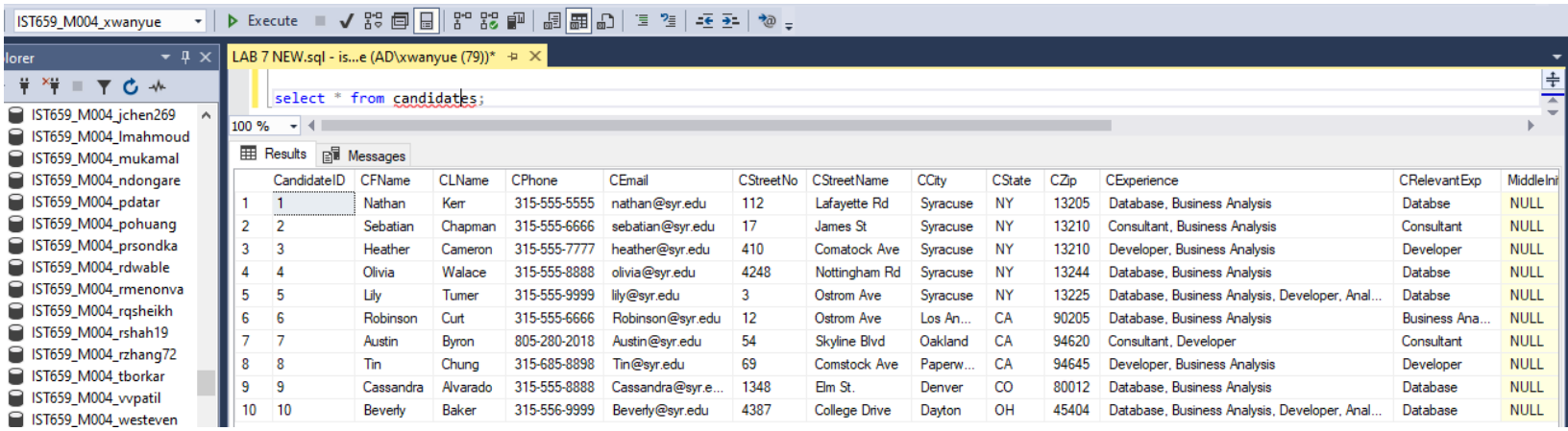



```
/* step 4*/
select top 2 * from companies order by companyID desc;
select top 2 * from positions order by positionID desc;
select top 3 * from Interviews order by InterviewID desc;
```

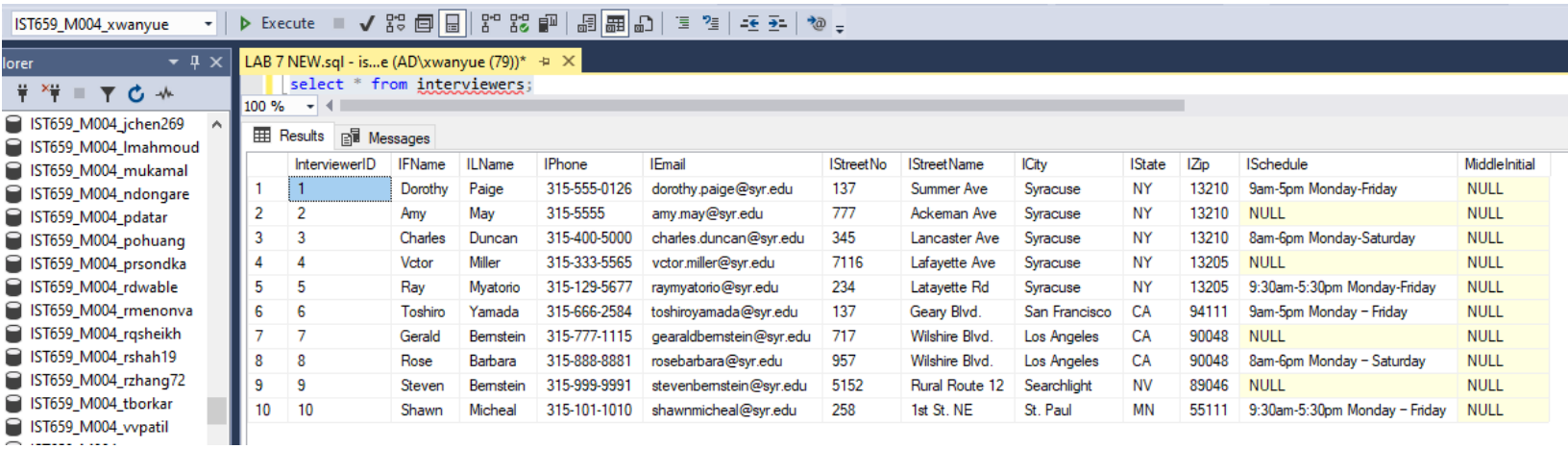



## Screenshot of Each Table:

### *Candidates:*

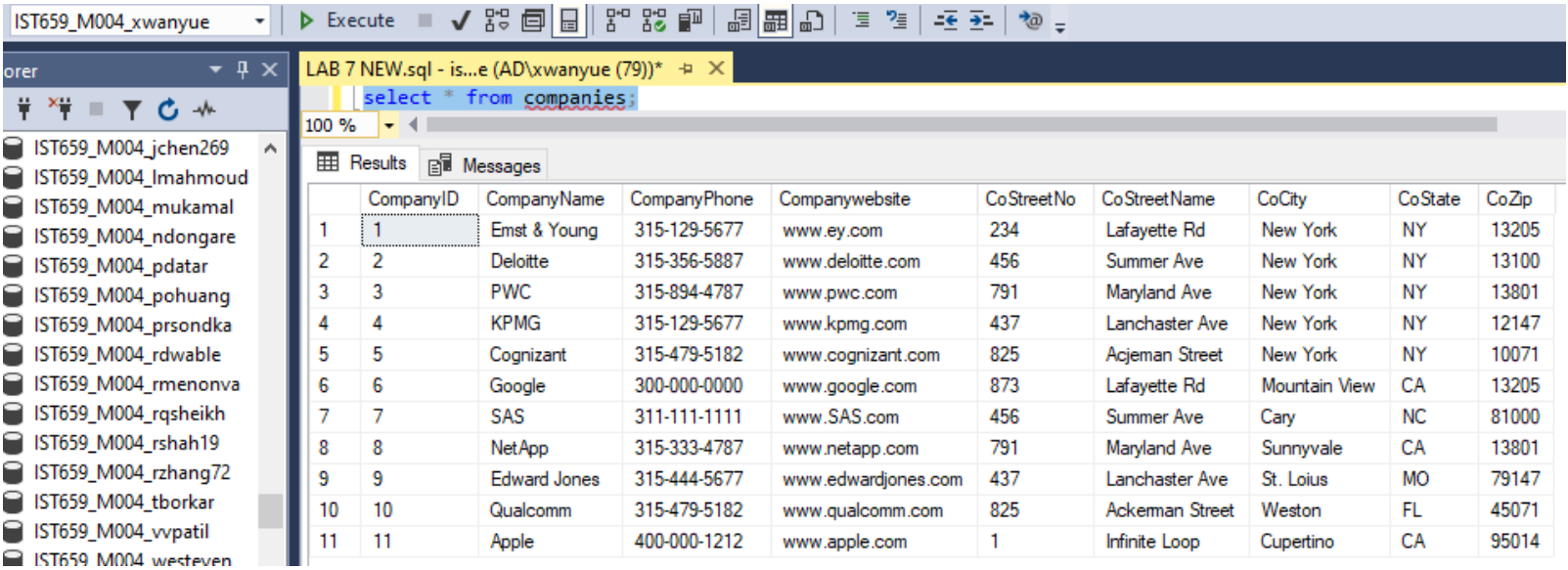


***Interviewers:***



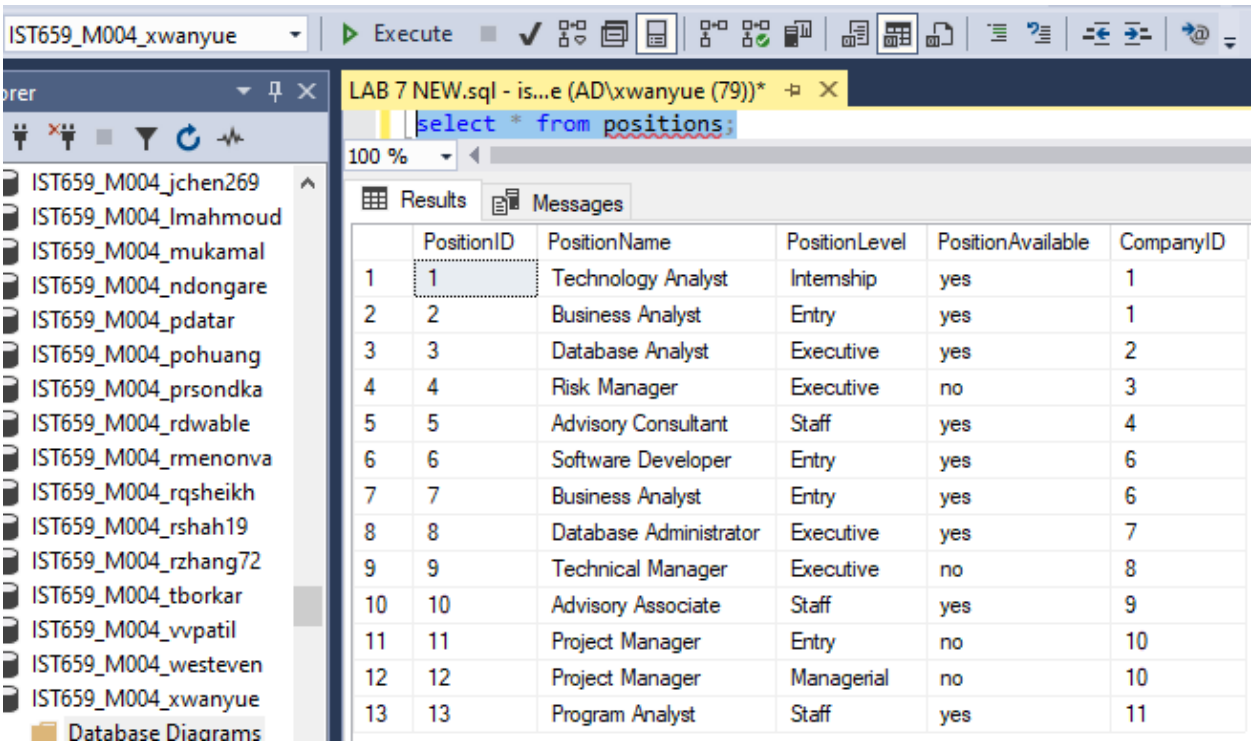

| | InterviewerID | IFName | ILName | IPhone | IEmail | IStreetNo | IStreetName | ICity | IState | IZip | ISchedule | MiddleInitial |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Dorothy | Paige | 315-555-0126 | dorothy.paige@syr.edu | 137 | Summer Ave | Syracuse | NY | 13210 | 9am-5pm Monday-Friday | NULL |
| 2 | 2 | Amy | May | 315-5555 | amy.may@syr.edu | 777 | Ackeman Ave | Syracuse | NY | 13210 | NULL | NULL |
| 3 | 3 | Charles | Duncan | 315-400-5000 | charles.duncan@syr.edu | 345 | Lancaster Ave | Syracuse | NY | 13210 | 8am-6pm Monday-Saturday | NULL |
| 4 | 4 | Vctor | Miller | 315-333-5565 | vctor.miller@syr.edu | 7116 | Lafayette Ave | Syracuse | NY | 13205 | NULL | NULL |
| 5 | 5 | Ray | Myatorio | 315-129-5677 | raymyatorio@syr.edu | 234 | Latayette Rd | Syracuse | NY | 13205 | 9:30am-5:30pm Monday-Friday | NULL |
| 6 | 6 | Toshiro | Yamada | 315-666-2584 | toshiroyamada@syr.edu | 137 | Geary Blvd. | San Francisco | CA | 94111 | 9am-5pm Monday – Friday | NULL |
| 7 | 7 | Gerald | Bernstein | 315-777-1115 | gearaldbernstein@syr.edu | 717 | Wilshire Blvd. | Los Angeles | CA | 90048 | NULL | NULL |
| 8 | 8 | Rose | Barbara | 315-888-8881 | rosebarbara@syr.edu | 957 | Wilshire Blvd. | Los Angeles | CA | 90048 | 8am-6pm Monday – Saturday | NULL |
| 9 | 9 | Steven | Bernstein | 315-999-9991 | stevenbernstein@syr.edu | 5152 | Rural Route 12 | Searchlight | NV | 89046 | NULL | NULL |
| 10 | 10 | Shawn | Micheal | 315-101-1010 | shawnmicheal@syr.edu | 258 | 1st St. NE | St. Paul | MN | 55111 | 9:30am-5:30pm Monday – Friday | NULL |

***Companies:***



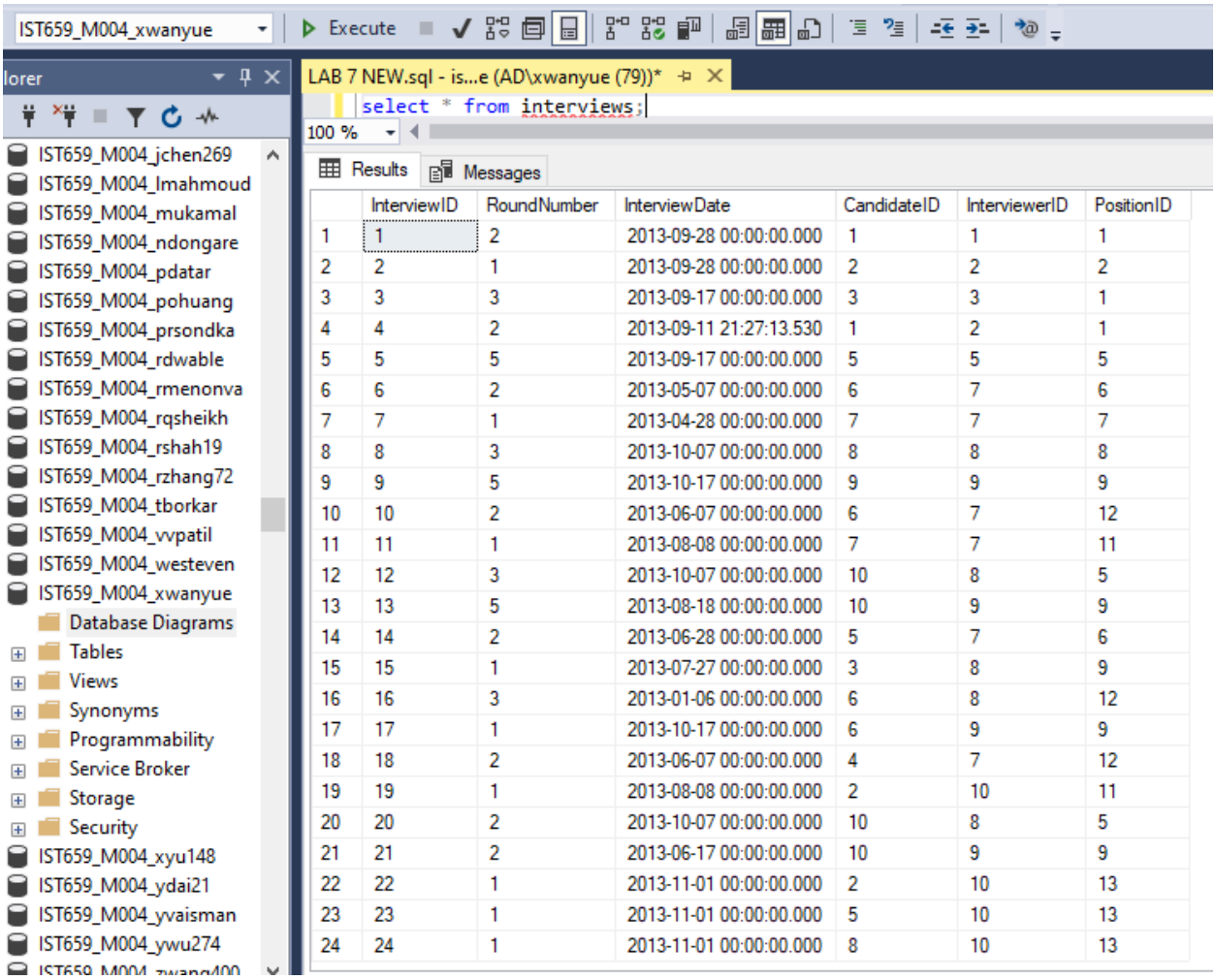

| | CompanyID | CompanyName | CompanyPhone | Companywebsite | CoStreetNo | CoStreetName | CoCity | CoState | CoZip |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Emst & Young | 315-129-5677 | www.ey.com | 234 | Lafayette Rd | New York | NY | 13205 |
| 2 | 2 | Deloitte | 315-356-5887 | www.deloitte.com | 456 | Summer Ave | New York | NY | 13100 |
| 3 | 3 | PWC | 315-894-4787 | www.pwc.com | 791 | Maryland Ave | New York | NY | 13801 |
| 4 | 4 | KPMG | 315-129-5677 | www.kpmg.com | 437 | Lanchaster Ave | New York | NY | 12147 |
| 5 | 5 | Cognizant | 315-479-5182 | www.cognizant.com | 825 | Acjeman Street | New York | NY | 10071 |
| 6 | 6 | Google | 300-000-0000 | www.google.com | 873 | Lafayette Rd | Mountain View | CA | 13205 |
| 7 | 7 | SAS | 311-111-1111 | www.SAS.com | 456 | Summer Ave | Cary | NC | 81000 |
| 8 | 8 | NetApp | 315-333-4787 | www.netapp.com | 791 | Maryland Ave | Sunnyvale | CA | 13801 |
| 9 | 9 | Edward Jones | 315-444-5677 | www.edwardjones.com | 437 | Lanchaster Ave | St. Loius | MO | 79147 |
| 10 | 10 | Qualcomm | 315-479-5182 | www.qualcomm.com | 825 | Ackerman Street | Weston | FL | 45071 |
| 11 | 11 | Apple | 400-000-1212 | www.apple.com | 1 | Infinite Loop | Cupertino | CA | 95014 |

***Positions:***




| | PositionID | PositionName | PositionLevel | PositionAvailable | CompanyID |
|---|---|---|---|---|---|
| 1 | 1 | Technology Analyst | Internship | yes | 1 |
| 2 | 2 | Business Analyst | Entry | yes | 1 |
| 3 | 3 | Database Analyst | Executive | yes | 2 |
| 4 | 4 | Risk Manager | Executive | no | 3 |
| 5 | 5 | Advisory Consultant | Staff | yes | 4 |
| 6 | 6 | Software Developer | Entry | yes | 6 |
| 7 | 7 | Business Analyst | Entry | yes | 6 |
| 8 | 8 | Database Administrator | Executive | yes | 7 |
| 9 | 9 | Technical Manager | Executive | no | 8 |
| 10 | 10 | Advisory Associate | Staff | yes | 9 |
| 11 | 11 | Project Manager | Entry | no | 10 |
| 12 | 12 | Project Manager | Managerial | no | 10 |
| 13 | 13 | Program Analyst | Staff | yes | 11 |

***Interviews:***




| | InterviewID | RoundNumber | InterviewDate | CandidateID | InterviewerID | PositionID |
|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 2013-09-28 00:00:00.000 | 1 | 1 | 1 |
| 2 | 2 | 1 | 2013-09-28 00:00:00.000 | 2 | 2 | 2 |
| 3 | 3 | 3 | 2013-09-17 00:00:00.000 | 3 | 3 | 1 |
| 4 | 4 | 2 | 2013-09-11 21:27:13.530 | 1 | 2 | 1 |
| 5 | 5 | 5 | 2013-09-17 00:00:00.000 | 5 | 5 | 5 |
| 6 | 6 | 2 | 2013-05-07 00:00:00.000 | 6 | 7 | 6 |
| 7 | 7 | 1 | 2013-04-28 00:00:00.000 | 7 | 7 | 7 |
| 8 | 8 | 3 | 2013-10-07 00:00:00.000 | 8 | 8 | 8 |
| 9 | 9 | 5 | 2013-10-17 00:00:00.000 | 9 | 9 | 9 |
| 10 | 10 | 2 | 2013-06-07 00:00:00.000 | 6 | 7 | 12 |
| 11 | 11 | 1 | 2013-08-08 00:00:00.000 | 7 | 7 | 11 |
| 12 | 12 | 3 | 2013-10-07 00:00:00.000 | 10 | 8 | 5 |
| 13 | 13 | 5 | 2013-08-18 00:00:00.000 | 10 | 9 | 9 |
| 14 | 14 | 2 | 2013-06-28 00:00:00.000 | 5 | 7 | 6 |
| 15 | 15 | 1 | 2013-07-27 00:00:00.000 | 3 | 8 | 9 |
| 16 | 16 | 3 | 2013-01-06 00:00:00.000 | 6 | 8 | 12 |
| 17 | 17 | 1 | 2013-10-17 00:00:00.000 | 6 | 9 | 9 |
| 18 | 18 | 2 | 2013-06-07 00:00:00.000 | 4 | 7 | 12 |
| 19 | 19 | 1 | 2013-08-08 00:00:00.000 | 2 | 10 | 11 |
| 20 | 20 | 2 | 2013-10-07 00:00:00.000 | 10 | 8 | 5 |
| 21 | 21 | 2 | 2013-06-17 00:00:00.000 | 10 | 9 | 9 |
| 22 | 22 | 1 | 2013-11-01 00:00:00.000 | 2 | 10 | 13 |
| 23 | 23 | 1 | 2013-11-01 00:00:00.000 | 5 | 10 | 13 |
| 24 | 24 | 1 | 2013-11-01 00:00:00.000 | 8 | 10 | 13 |

**Submission**

Please submit your lab report in one doc file named “lastname-firstname-lab7.doc”. Please submit your lab report in one Word file to BB. You can use MS OneNote/snipping tool to capture and edit screenshots of the SQL statements and their results. Remember to add comments to your SQL statements to explain the purpose of the code blocks.

The lab report should follow the template in lab 6 solutions. After each question, copy and paste your SQL statement, followed by the screenshot to show that your SQL statement has been successfully executed. **Both SQL statement and the result should be visible in the screenshot.** Remember to add comments to your SQL statements to explain the purpose of the code blocks. Also **add all the final table screenshots** at the end of the lab. Include Screen shots of the SQL Server Management Studio with your username which is there at the bottom right hand corner.

# Comment Summary

Page 2

1. you can also use YEAR() function.
2. You can also use MONTH() function in SQL Server