**IST 659 Lab 5 SQL II**

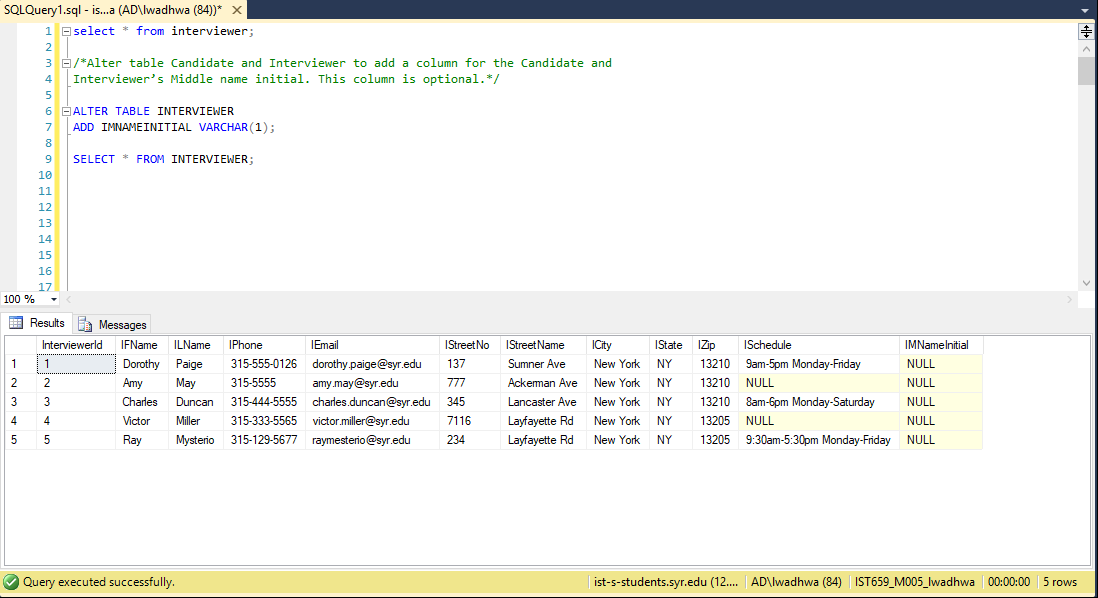
Alter table Candidate and Interviewer to add a column for the Candidate and Interviewer’s Middle name initial. This column is optional.

Query

alter table interviewer

add IMNameInitial varchar(1);

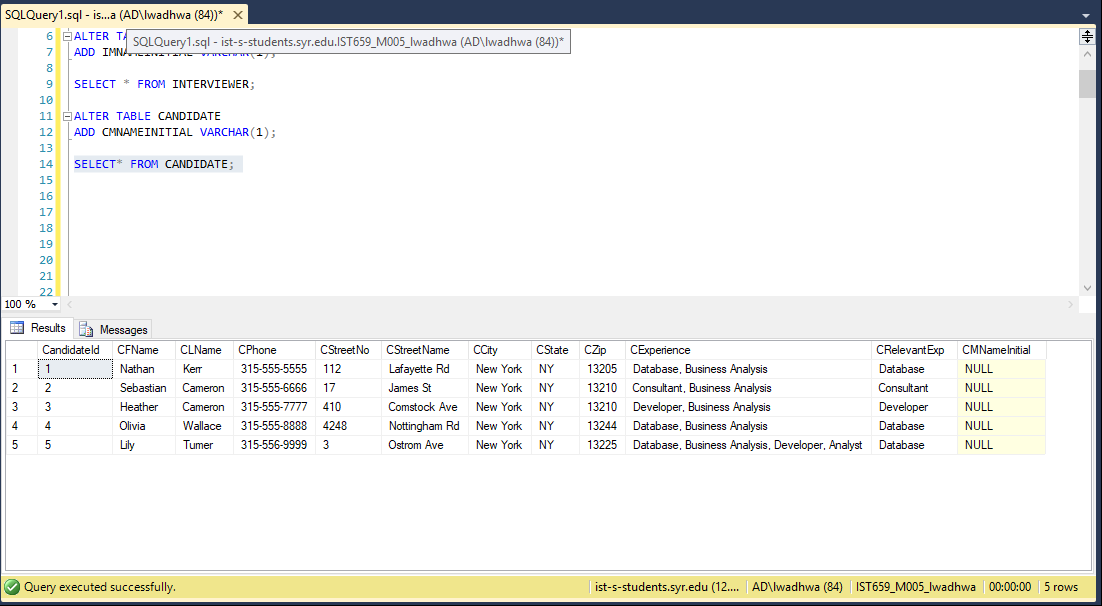
select \* from interviewer;



alter table Candidate

add CMNameInitial varchar(1);

select\* from CANDIDATE;



2) Alter tables to let users input state abbreviate name instead of full name. For example, if state name is “New York”, change it to “NY”. Similarly, if state name is “California”, change it to “CA” and so on. Change all the tables that have the state column. (Candidate, Interviewer, Company). Also alter the data we inserted to fit this new datatype. Hint: if there are already data in these tables and the data do not conform to the new data type, you might need to remove the data before you can alter the table.

--Update Statements

UPDATE CANDIDATE

SET CSTATE = 'NY'

WHERE CSTATE = 'NEW YORK';

UPDATE INTERVIEWER

SET ISTATE = 'NY'

WHERE ISTATE = 'NEW YORK';

UPDATE COMPANY

SET CSTATE = 'NY'

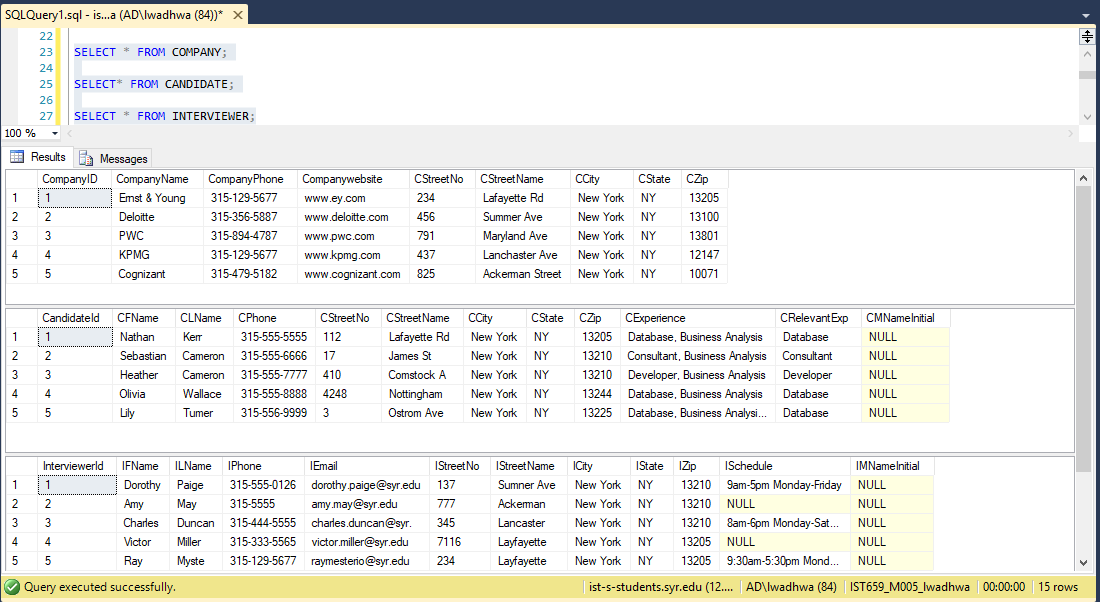
WHERE CSTATE = 'NEW YORK';

--Select Statements

SELECT \* FROM COMPANY;

SELECT\* FROM CANDIDATE;

SELECT \* FROM INTERVIEWER;



--Alter table statements

ALTER TABLE CANDIDATE

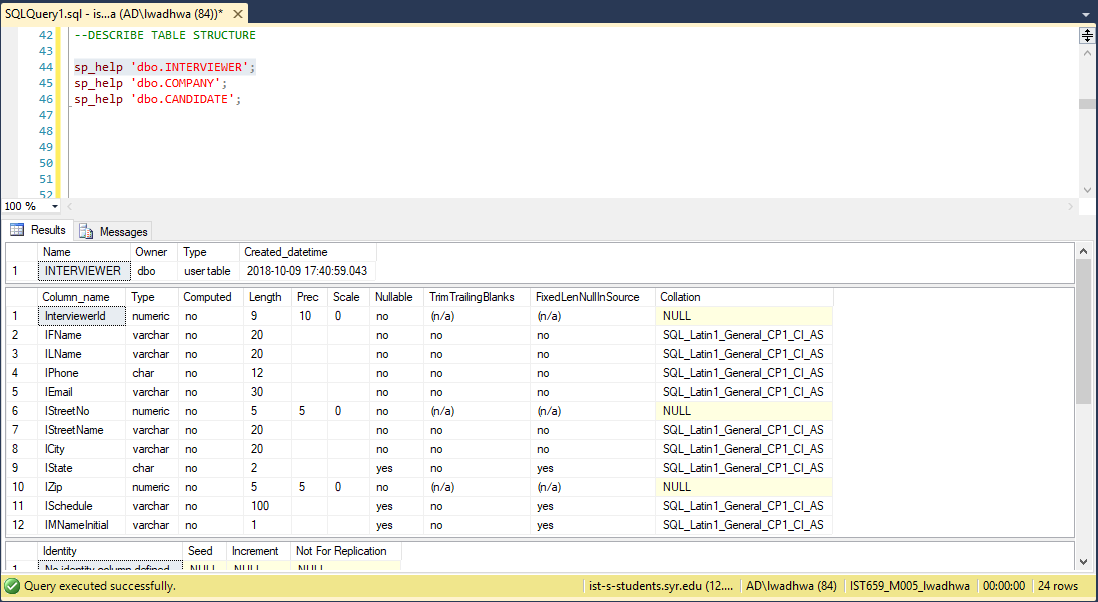
ALTER COLUMN CSTATE CHAR(2);

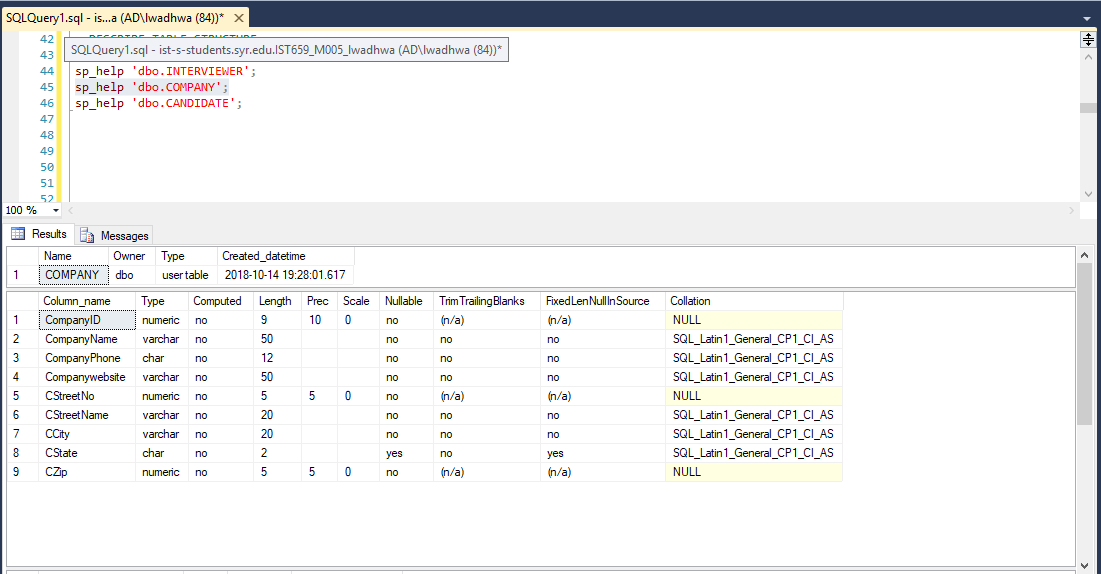
ALTER TABLE INTERVIEWER

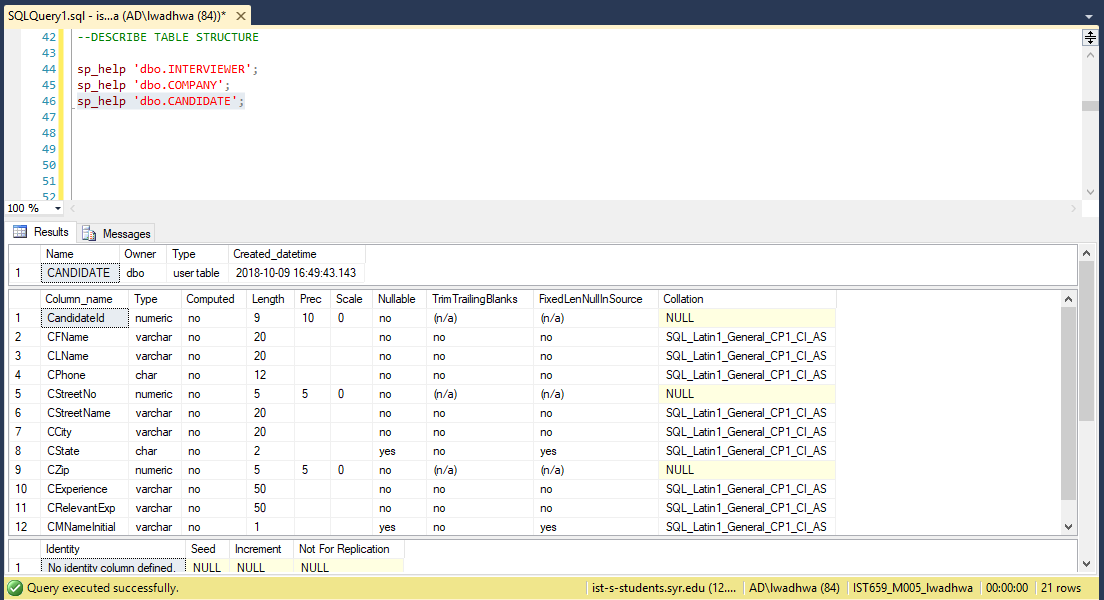
ALTER COLUMN ISTATE CHAR(2);

ALTER TABLE COMPANY

ALTER COLUMN CSTATE CHAR(2);



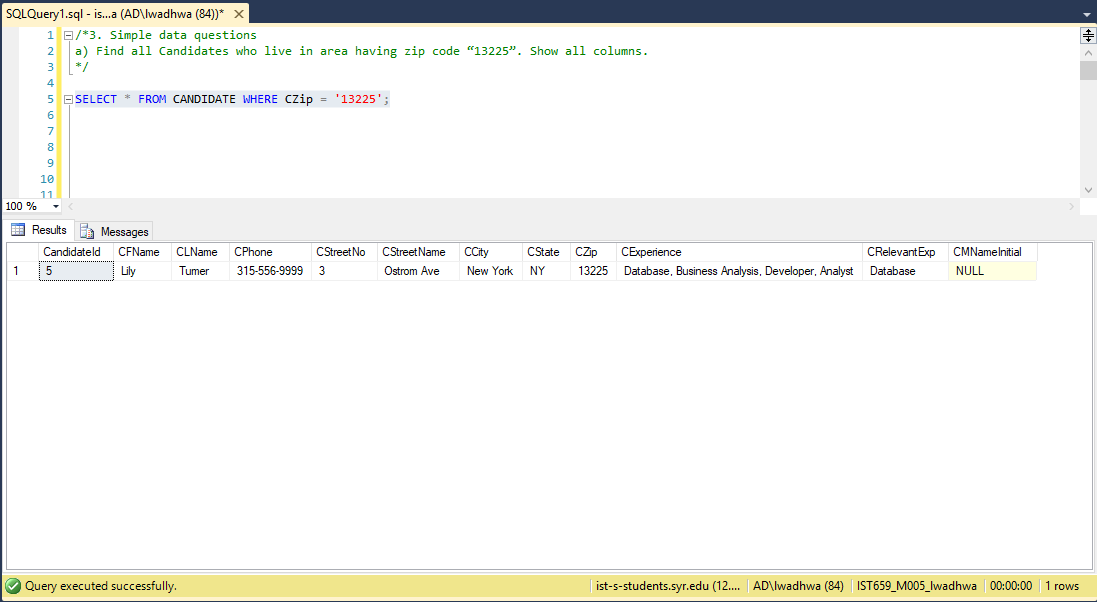




3. Simple data questions

a) Find all Candidates who live in area having zip code “13225”. Show all columns.

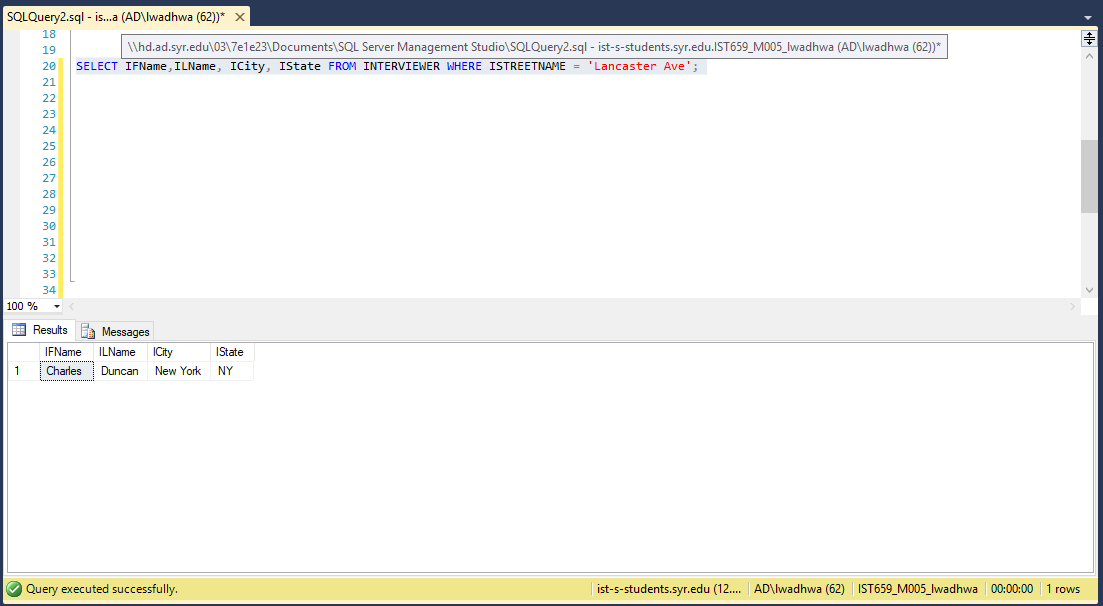
SELECT \* FROM CANDIDATE WHERE CZip = '13225';



b) Find all Interviewers living on “Lancaster Ave”. Show Interviewer name, City and

State.

SELECT IFName,ILName, ICity, IState FROM INTERVIEWER WHERE ISTREETNAME = 'Lancaster Ave';



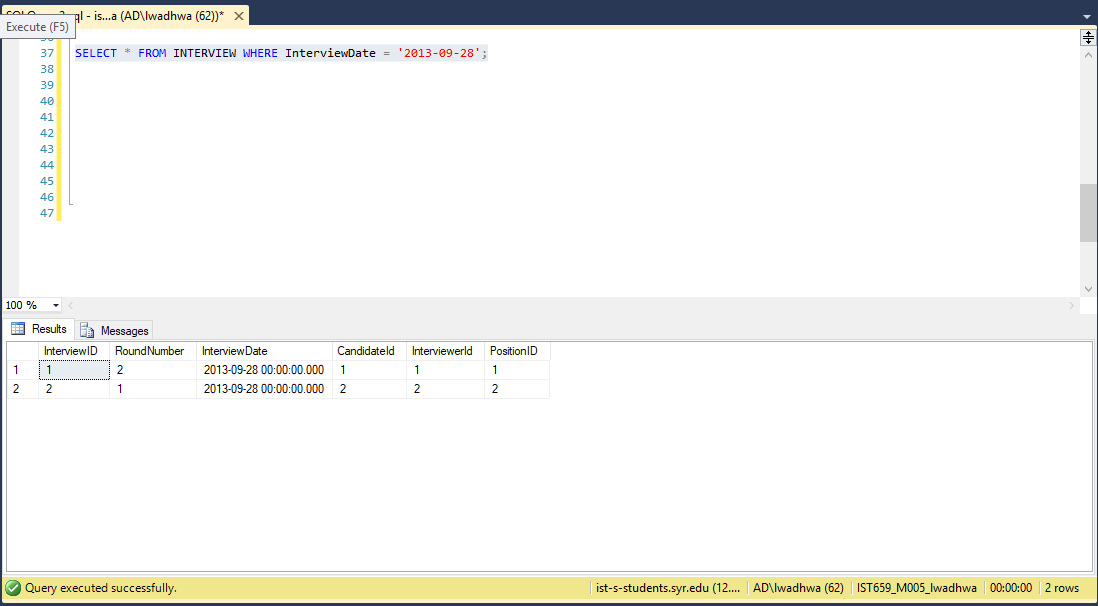
c) Update the table ‘Interview’, set date as ‘2013-09-28 00:00:00.000’ where interviewID is ‘1’. Find all interviews that took place on 28th day in the month of September, 2013. Show Interviewer ID and Position ID.

UPDATE INTERVIEW

SET InterviewDate = '2013-09-28 00:00:00.000'

WHERE InterviewID = '1';

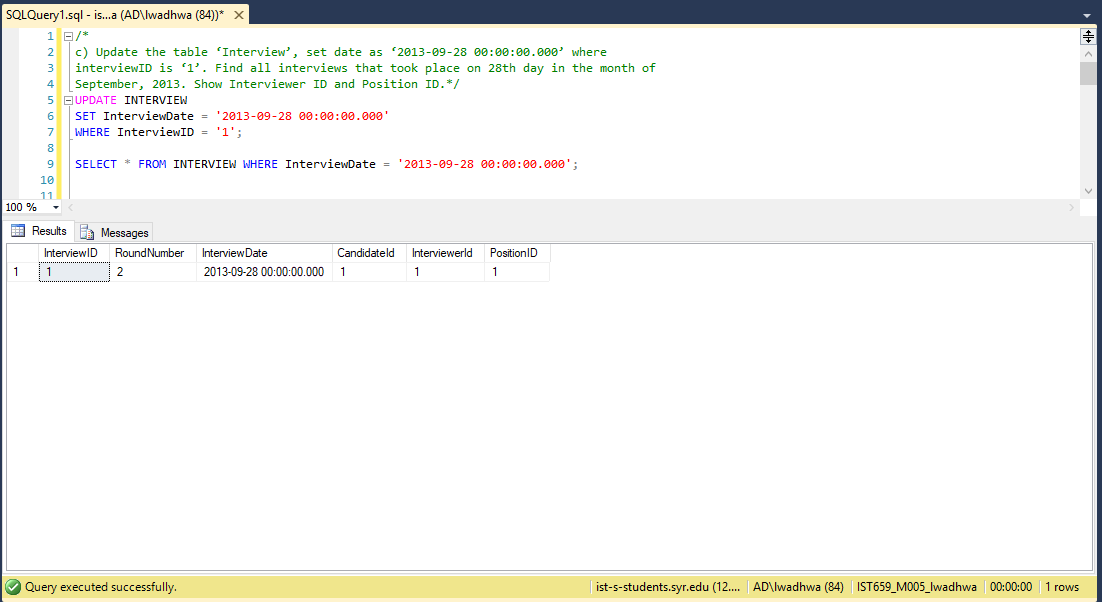
SELECT \* FROM INTERVIEW WHERE InterviewDate = '2013-09-28';



d) Find all positions that are not available for level “Executive”. Show Position Name

and position available only.

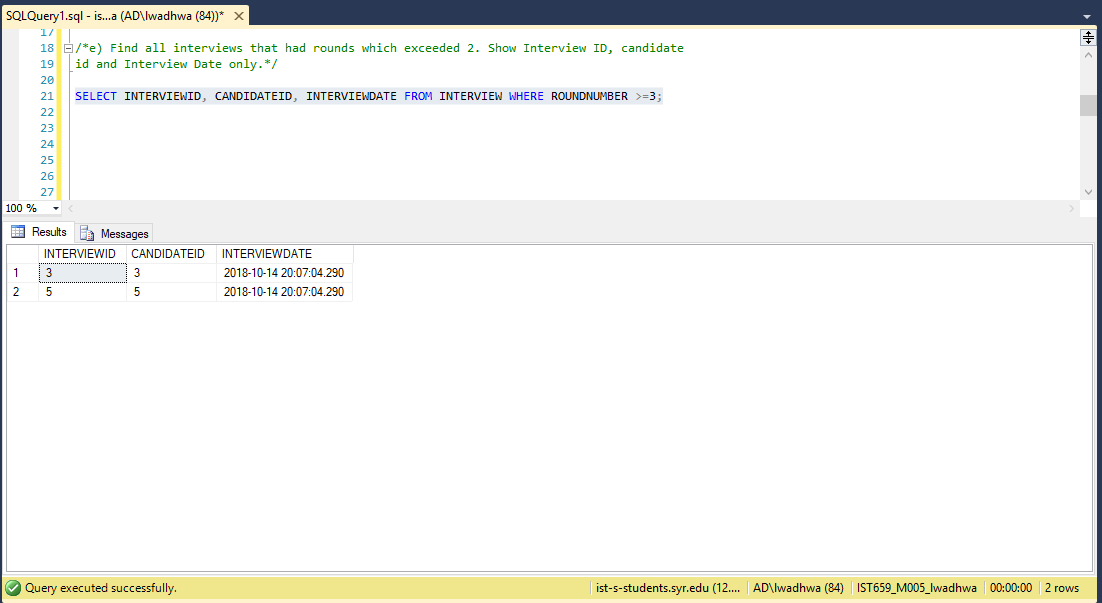
SELECT PositionName,PositionAvailable FROM POSITION where PositionLevel != 'Executive';



e) Find all interviews that had rounds which exceeded 2. Show Interview ID, candidate

id and Interview Date only.

select InterviewID, candidateId, InterviewDate from INTERVIEW where RoundNumber >=3;

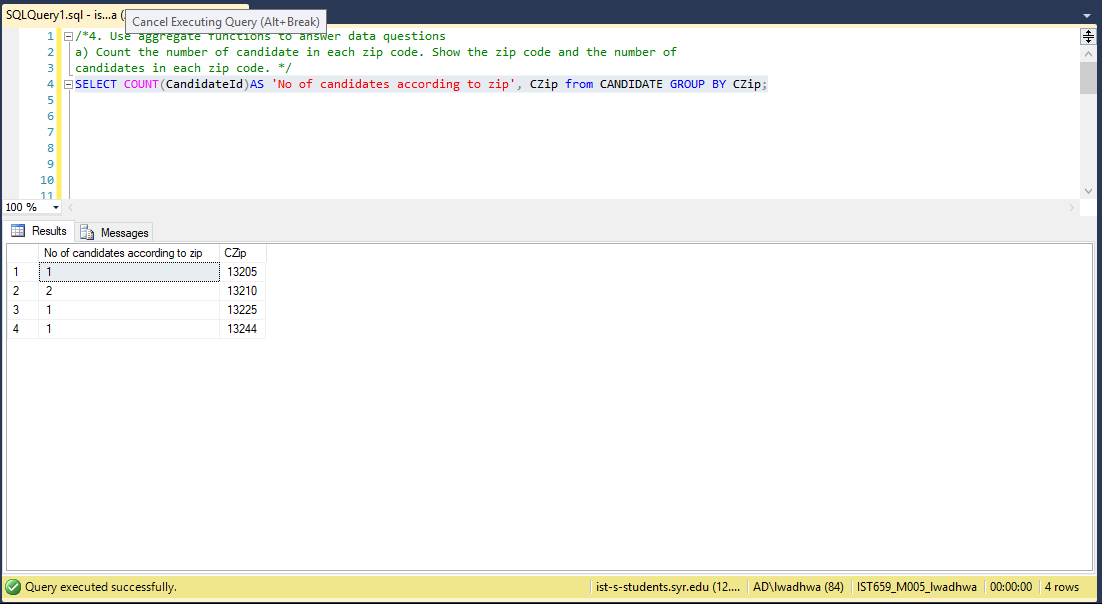


4. Use aggregate functions to answer data questions

a) Count the number of candidate in each zip code. Show the zip code and the number of

candidates in each zip code.

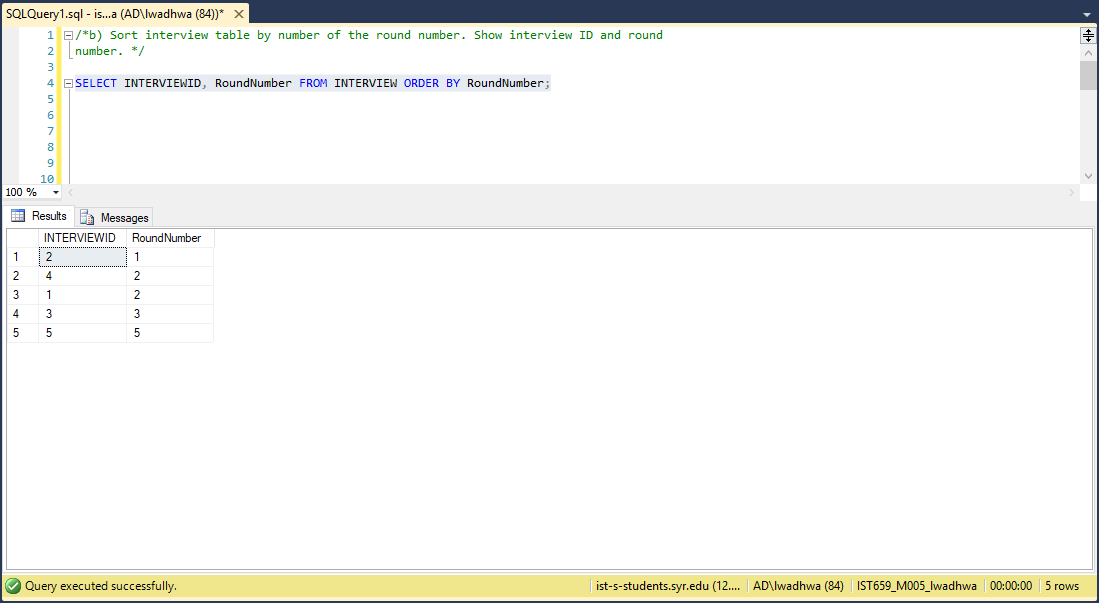
SELECT COUNT(CandidateId), CZip from CANDIDATE GROUP BY CZip;



b) Sort interview table by number of the round number. Show interview ID and round

number.

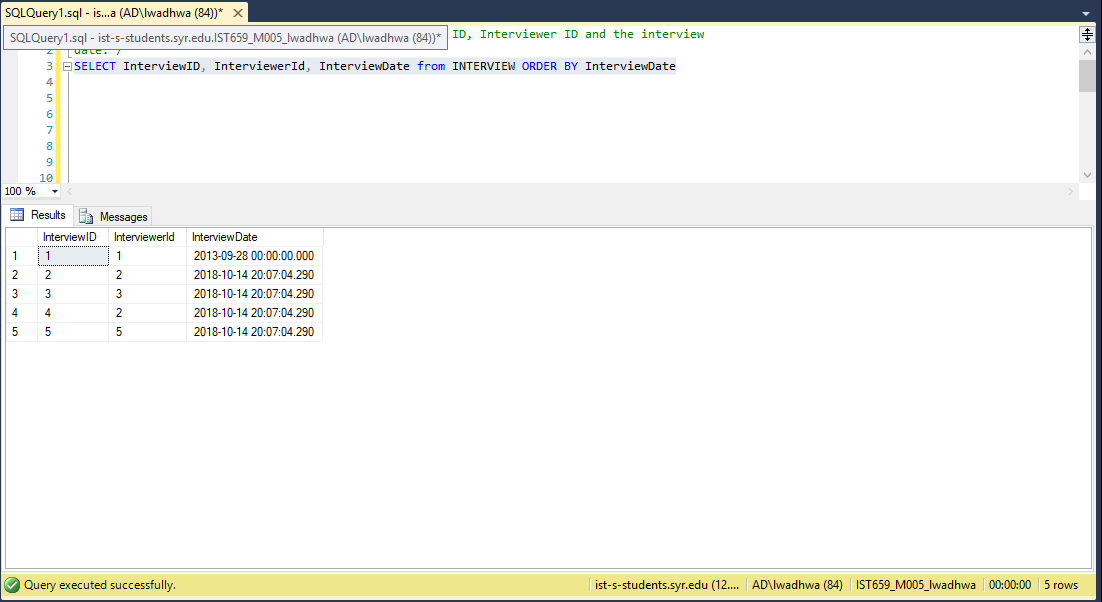
SELECT INTERVIEWID, RoundNumber FROM INTERVIEW ORDER BY RoundNumber;



c) Sort interview by interview date. Show interview ID, Interviewer ID and the interview

date.

SELECT InterviewID, InterviewerId, InterviewDate from INTERVIEW ORDER BY InterviewDate



d) For each candidate, calculate the average, min, and max round number of the

interviews that this candidate had. Show CandidateId, average Round Number, min

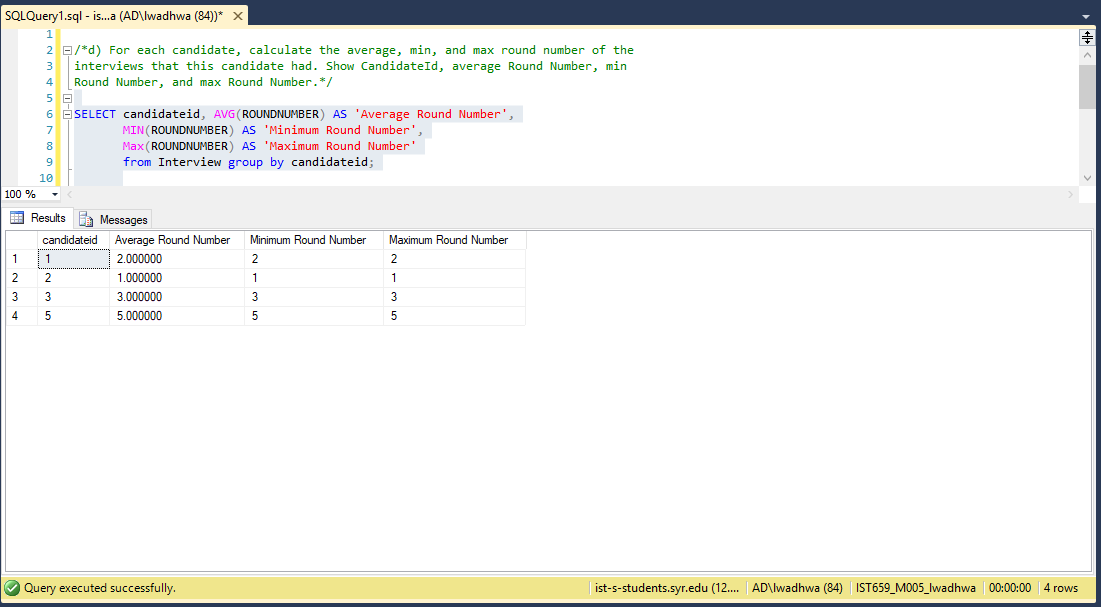
Round Number, and max Round Number.

SELECT candidateid, AVG(ROUNDNUMBER) AS 'Average Round Number',

MIN(ROUNDNUMBER) AS 'Minimum Round Number',

Max(ROUNDNUMBER) AS 'Maximum Round Number'

from Interview group by candidateid;



e) Find all whose average round number of interviews is below 3 (i.e. average

round number equals 1 or 2). Show the CandidateID and average round number.

SELECT candidateid, AVG(ROUNDNUMBER) AS Average\_Round\_Number from Interview

group by CandidateId

HAVING AVG(ROUNDNUMBER) < 3

