TITLE: A DATABASE OF PHARMACEUTICAL SALES REPS AND SCHOOL PERFORMANCE COMPARED WITH TOTAL STUDENT LOAN AMOUNTS.

Introduction: As one can see, an attempt to create the tables and enter the data all at once failed, so the result is 9 pages of code and associated print screens of the end results of creating the PHARMA Database piece by piece. The database is chronicled below from start to finish and contains two parent tables, an intersection child table as well as a fourth table devoted to student loan amounts, which is also a child table. The first parent table is the PHARMA_REP table which is devoted to the names and addresses of the pharmaceutical representatives in the database. The second parent table is devoted to SCHOOL_MATRICULATION and contains pertinent information about the school that each pharmaceutical representative attended. The third table is an intersection table devoted to SCHOOL_PERFORMANCE, which again is a child table. The fourth table is devoted to STUDENT_LOANS and contains total student loan amounts.

The purpose of the database is to keep track of student performance based on school attended, the school's accreditation and associated total student loan amounts. This information can be utilized by corporations for determining promotions and to chronicle the amount of student loans each representative was required to pay for their education. This information can be used to ensure that the salespeople are paying down their debt as a program can be set up to have payments deducted from the salesperson's pay before taxes are taken out if a program to do so is approved by the federal government. If the program is approved, funds used for student loan payments would not be taxed.

Body
Again, the issues that occurred were related to attempting to attempting to create the database and insert the data into the database all at once. The business rules are such that there can be one sales representative to exactly one school performance. I did not cascade the pharma rep and school performance tables, but they can be cascaded so that when a pharma rep leaves the industry, his performance in school goes away. The same can be said for the pharma rep and student loans table. Also, schools will not be deleted when school performance for a rep is deleted. The reason for this is that schools should not go away as they are likely to be associated with more than one rep.

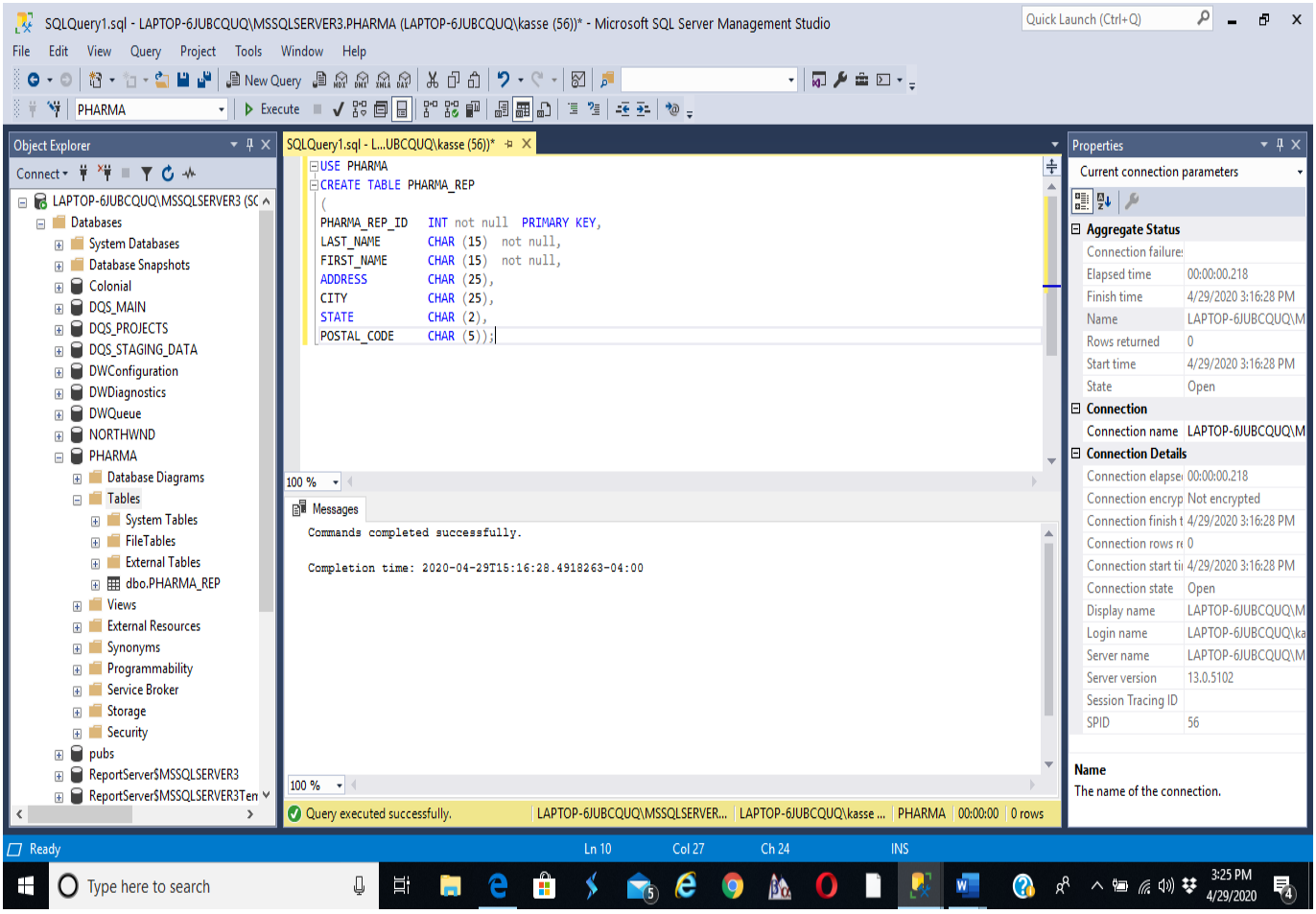
In my own research, I discovered that most pharmaceutical representatives are hired from AACSB schools. The reason for this is that these schools are apparently considered the gold standard for business education. If a sales performance table were included, it would be possible to isolate performance level in the pharmaceutical industry versus the school accreditation to determine if there is any real advantage to having an education from an AACSB school. In one study in the accounting profession there was no statistically significant difference in performance in the accounting field for students from AACSB schools who were employed in the accounting profession versus those from schools with the other two accreditations.

Inner join tables (These tables can be found after the tables showing the construction of the database along with an explanation of what each query depicts). Also included is an example of the business application for which the query can be used. In addition, three aggregate function queries are included as well.

PHARMA_REP PARENT TABLE

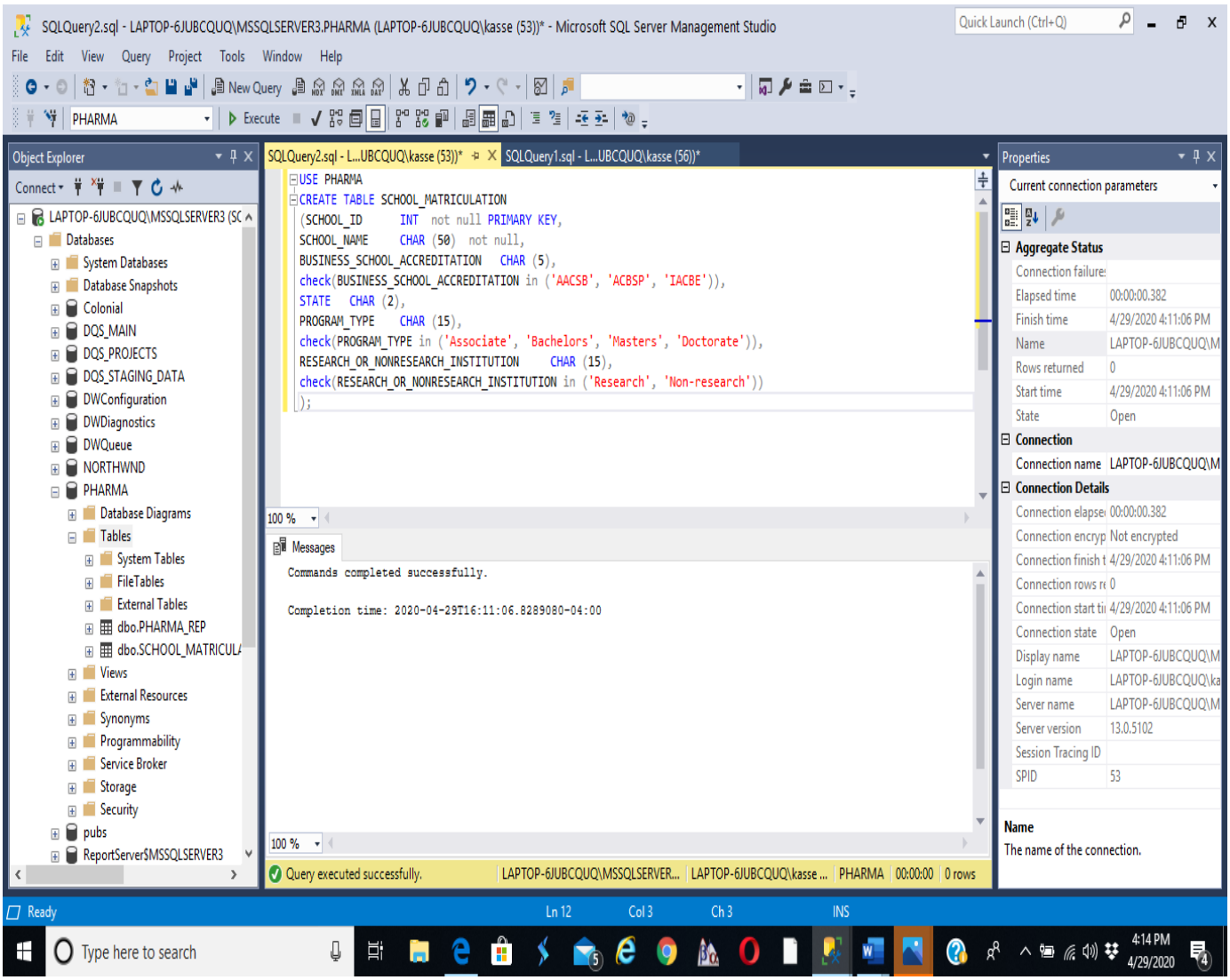
```
USE PHARMA
CREATE TABLE PHARMA_REP
(
PHARMA_REP_ID INT not null  PRIMARY KEY,
LAST_NAME           CHAR (15)  not null,
FIRST_NAME          CHAR (15)  not null,
```

```
ADDRESS                    CHAR (25),
CITY               CHAR (25),
STATE              CHAR (2),
POSTAL_CODE        CHAR (5));
```



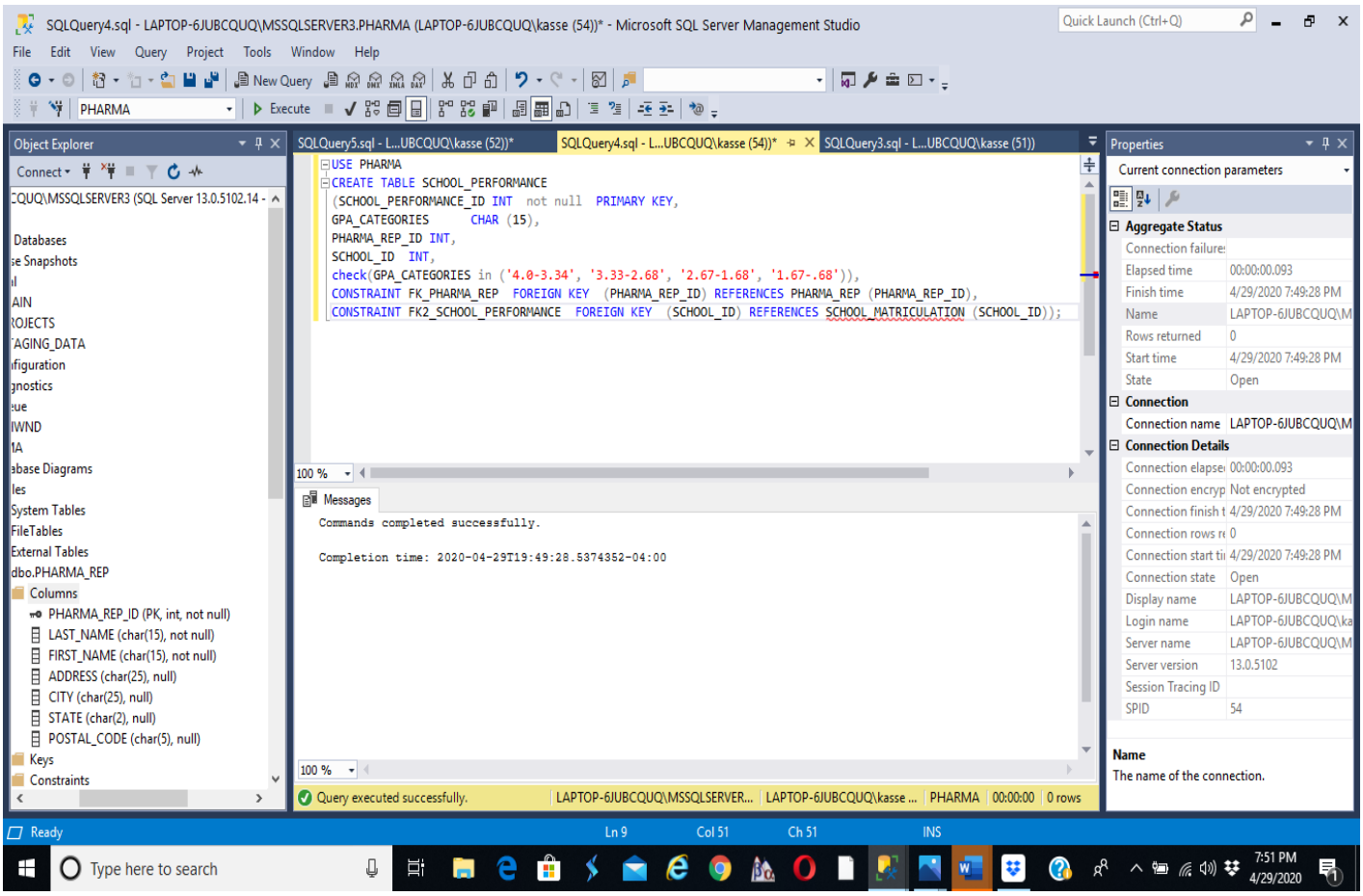
SCHOOL_MATRICULATION PARENT TABLE

```
USE PHARMA
CREATE TABLE SCHOOL_MATRICULATION
(SCHOOL_ID           INT  not null PRIMARY KEY,
SCHOOL_NAME          CHAR (50)  not null,
BUSINESS_SCHOOL_ACCREDITATION      CHAR (5),
check(BUSINESS_SCHOOL_ACCREDITATION in ('AACSB', 'ACBSP', 'IACBE')),
STATE  CHAR (2),
PROGRAM_TYPE  CHAR (15),
check(PROGRAM_TYPE in ('Associate', 'Bachelors', 'Masters', 'Doctorate')),
RESEARCH_OR_NONRESEARCH_INSTITUTION              CHAR (15),
check(RESEARCH_OR_NONRESEARCH_INSTITUTION in ('Research', 'Non-research'))
);
```



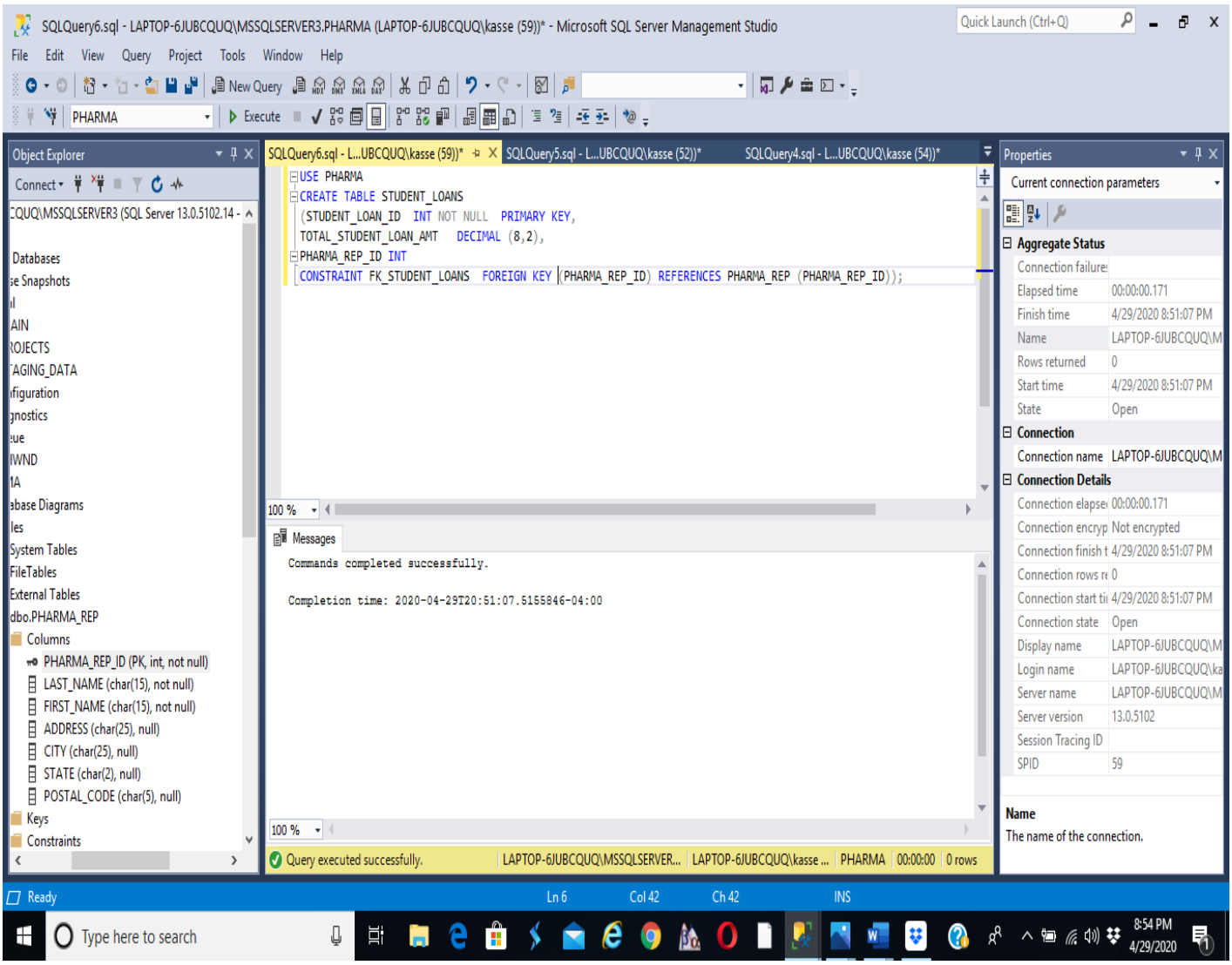
SCHOOL_PERFORMANCE CHILD INTERSECTION TABLE

```
USE PHARMA
CREATE TABLE SCHOOL_PERFORMANCE
(SCHOOL_PERFORMANCE_ID INT  not null  PRIMARY KEY,
GPA_CATEGORIES              CHAR (15),
PHARMA_REP_ID INT,
SCHOOL_ID  INT,
check(GPA_CATEGORIES in ('4.0-3.34', '3.33-2.68', '2.67-1.68', '1.67-.68')),
CONSTRAINT FK_PHARMA_REP  FOREIGN KEY  (PHARMA_REP_ID) REFERENCES PHARMA_REP
(PHARMA_REP_ID),
CONSTRAINT FK2_SCHOOL_PERFORMANCE  FOREIGN KEY  (SCHOOL_ID) REFERENCES
SCHOOL_MATRICULATION (SCHOOL_ID));
```



STUDENT_LOANS CHILD TABLE

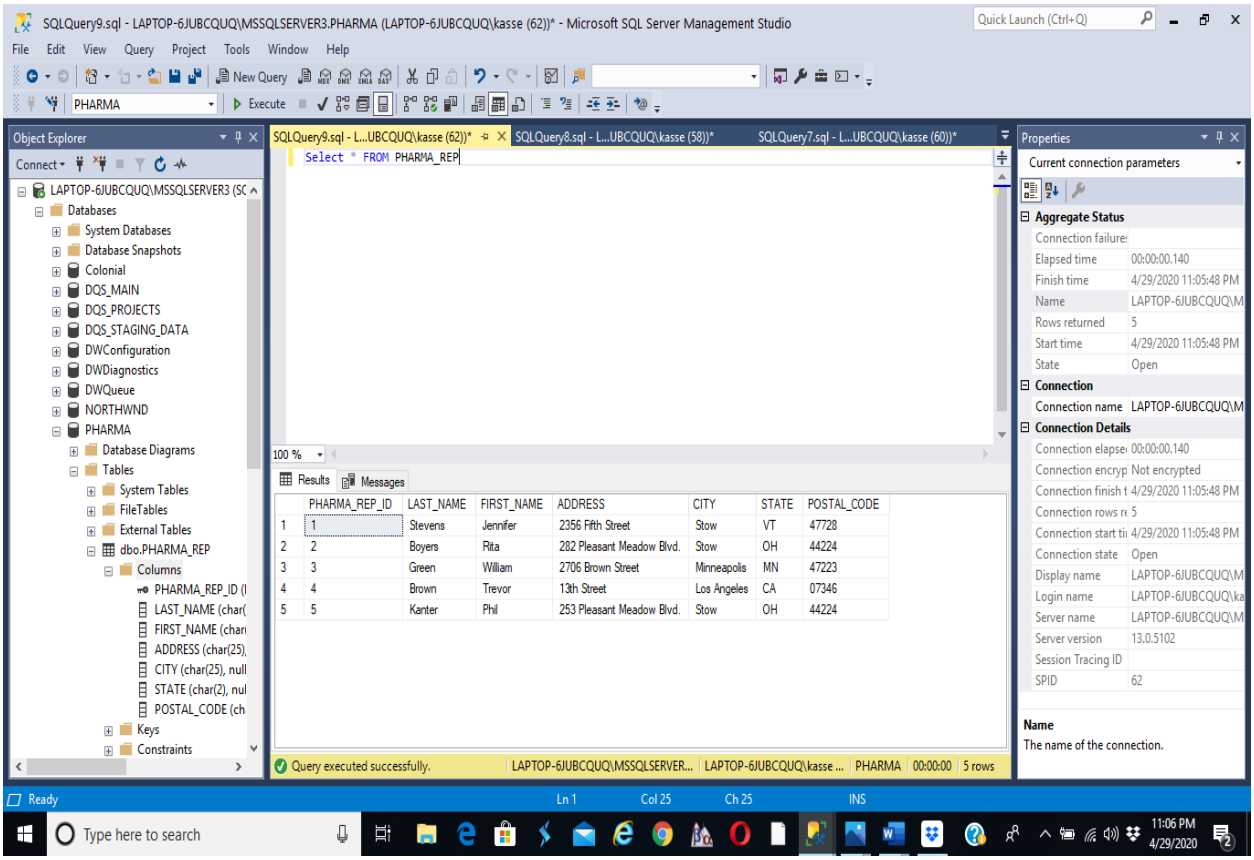
```
USE PHARMA
CREATE TABLE STUDENT_LOANS
(STUDENT_LOAN_ID  INT NOT NULL  PRIMARY KEY,
TOTAL_STUDENT_LOAN_AMT   DECIMAL (8,2),
PHARMA_REP_ID INT
CONSTRAINT FK_STUDENT_LOANS  FOREIGN KEY (PHARMA_REP_ID) REFERENCES PHARMA_REP
(PHARMA_REP_ID));
```



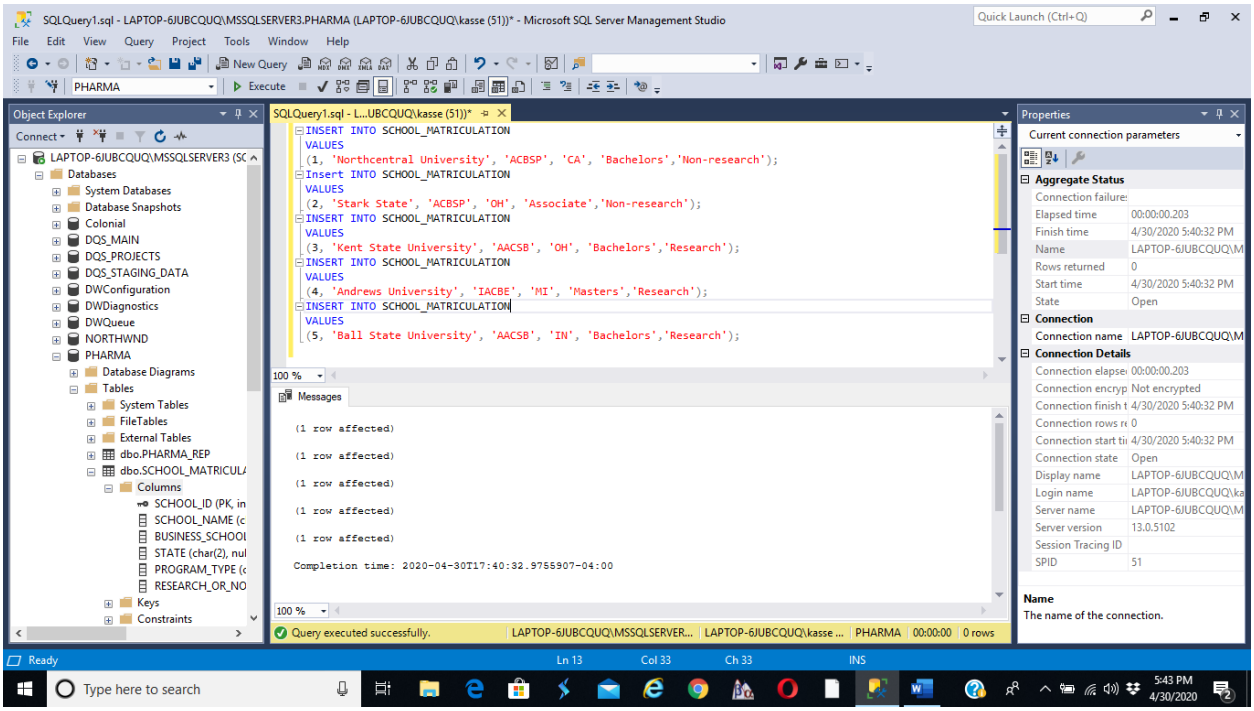

```
Select * FROM PHARMA_REP

INSERT INTO PHARMA_REP
VALUES
(1,'Stevens', 'Jennifer', '2356 Fifth Street', 'Stow', 'VT', '47728');
INSERT INTO PHARMA_REP
VALUES
(2,'Boyers', 'Rita', '282 Pleasant Meadow Blvd.', 'Stow', 'OH', '44224');
INSERT INTO PHARMA_REP
VALUES
(3,'Green', 'William', '2706 Brown Street', 'Minneapolis', 'MN', '47223');
INSERT INTO PHARMA_REP
VALUES
(4,'Brown', 'Trevor', '13th Street', 'Los Angeles', 'CA', '07346');
```

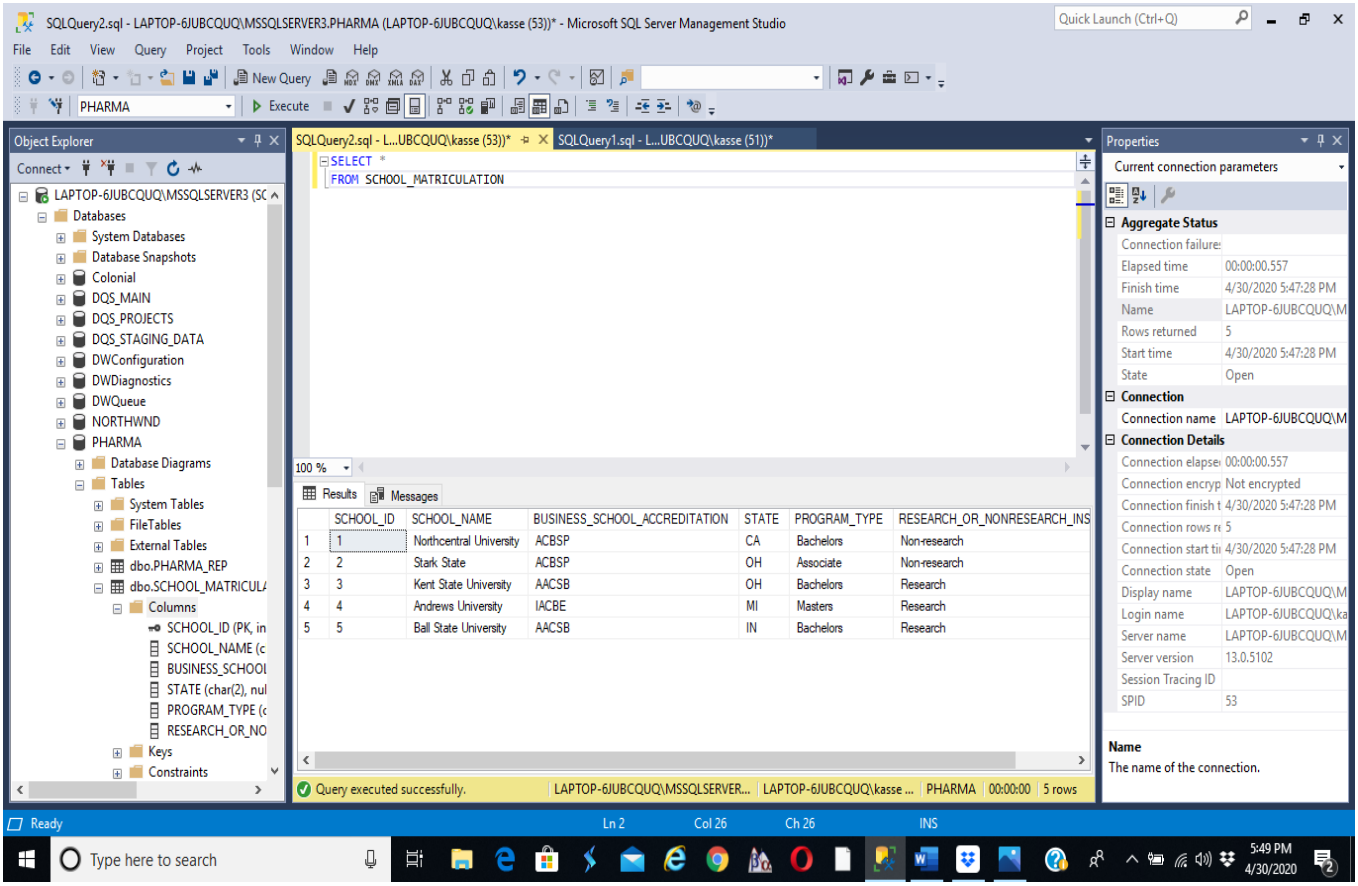
```
INSERT INTO PHARMA_REP
VALUES
(5,'Kanter', 'Phil', '1423 Buckingham Gate Blvd.', 'Cuyahoga Falls', 'OH', '44221');
```



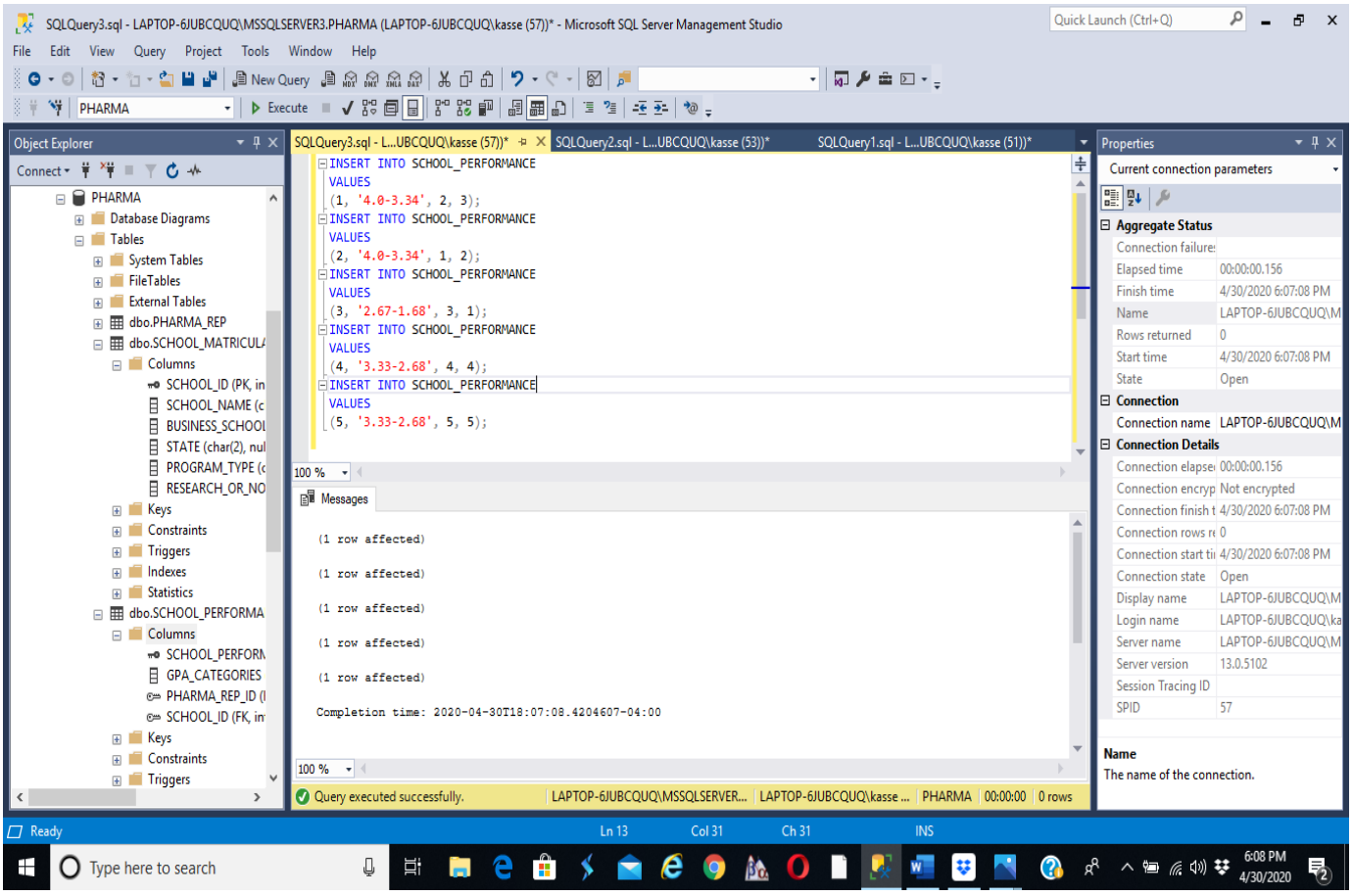

```
INSERT INTO SCHOOL_MATRICULATION
VALUES
(1, 'Northcentral University', 'ACBSP', 'CA', 'Bachelors','Non-research');
Insert INTO SCHOOL_MATRICULATION
VALUES
(2, 'Stark State', 'ACBSP', 'OH', 'Associate','Non-research');
INSERT INTO SCHOOL_MATRICULATION
VALUES
(3, 'Kent State University', 'AACSB', 'OH', 'Bachelors','Research');
INSERT INTO SCHOOL_MATRICULATION
VALUES
(4, 'Andrews University', 'IACBE', 'MI', 'Masters','Research');
INSERT INTO SCHOOL_MATRICULATION
VALUES
(5, 'Ball State University', 'AACSB', 'IN', 'Bachelors','Research');
```



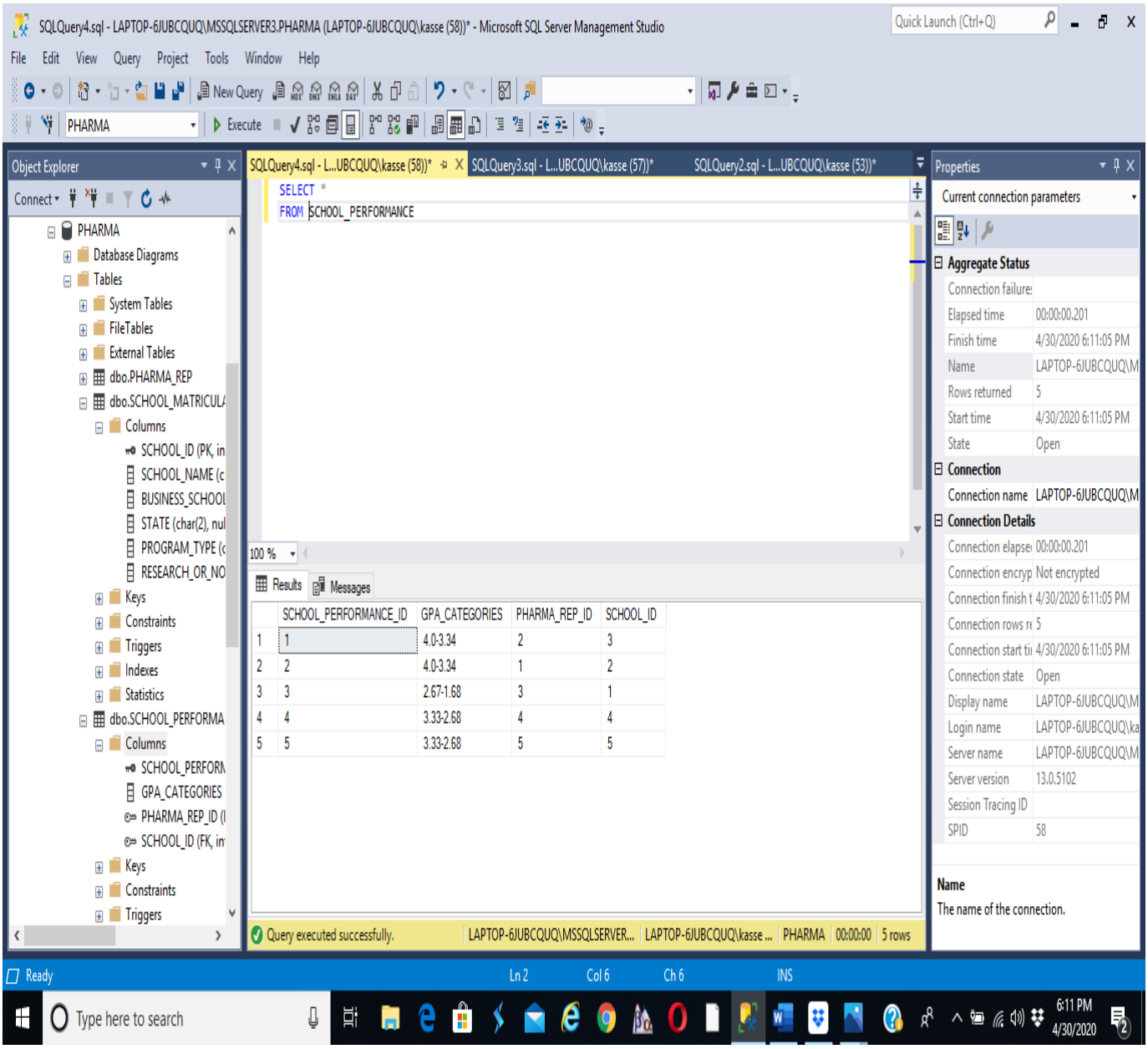

```
SELECT *
FROM SCHOOL_MATRICULATION
```



```
Data inserted into school performance table
INSERT INTO SCHOOL_PERFORMANCE
VALUES
(1, '4.0-3.34', 2, 3);
INSERT INTO SCHOOL_PERFORMANCE
VALUES
(2, '4.0-3.34', 1, 2);
INSERT INTO SCHOOL_PERFORMANCE
VALUES
(3, '2.67-1.68', 3, 1);
INSERT INTO SCHOOL_PERFORMANCE
VALUES
(4, '3.33-2.68', 4, 4);
INSERT INTO SCHOOL_PERFORMANCE
VALUES
(5, '3.33-2.68', 5, 5);
```



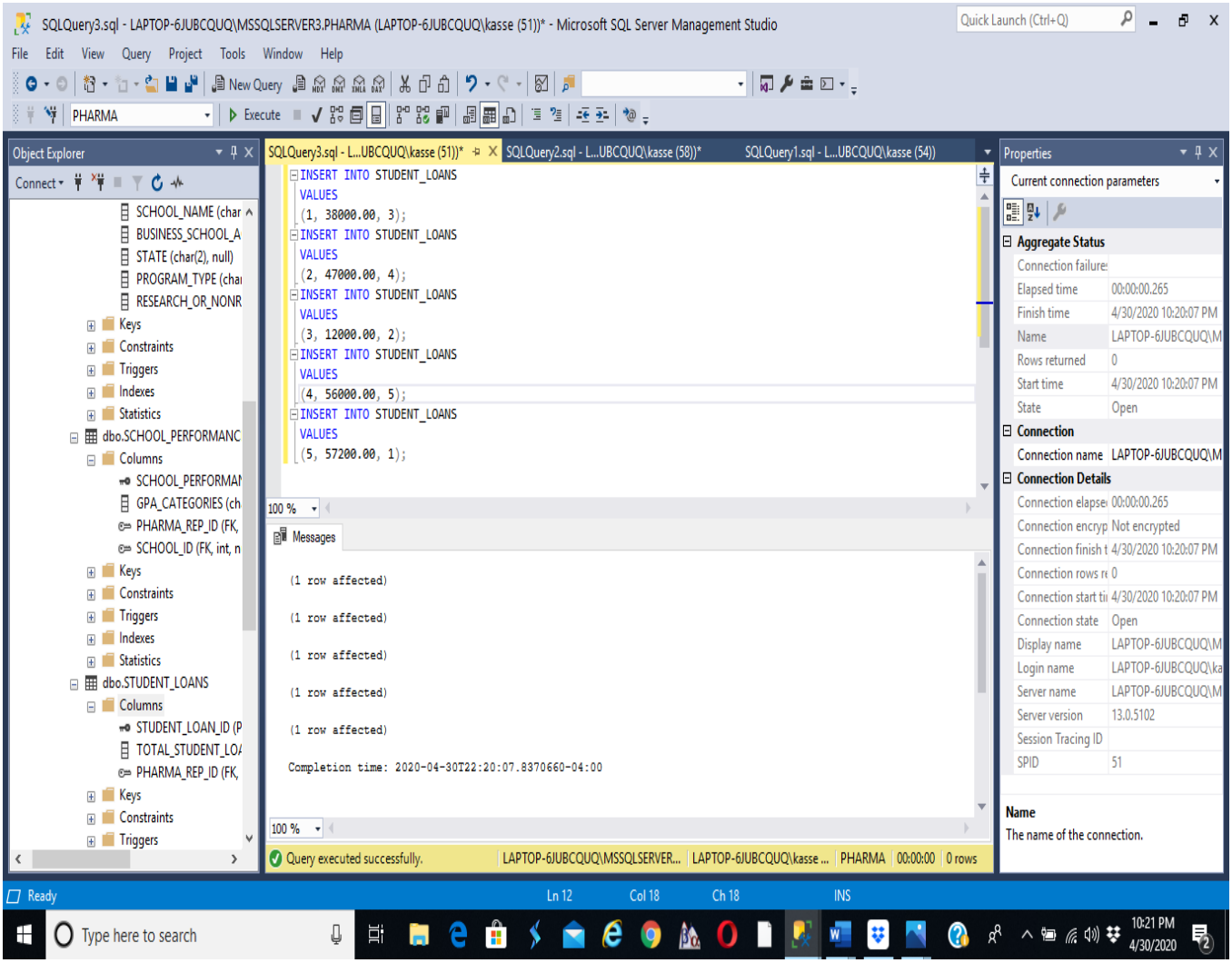
SCHOOL_PERFORMANCE TABLE DATA



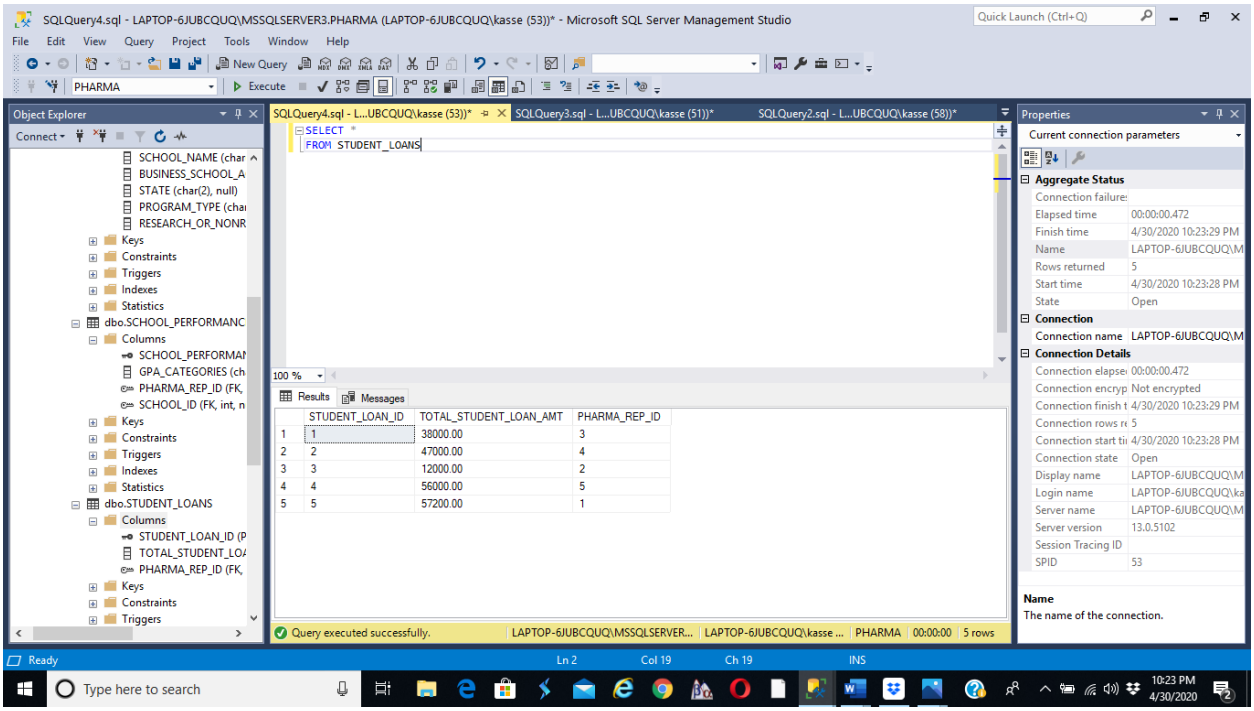
Student Loans Table Data

```
INSERT INTO STUDENT_LOANS
VALUES
(1, 38000.00, 3);
INSERT INTO STUDENT_LOANS
VALUES
(2, 47000.00, 4);
INSERT INTO STUDENT_LOANS
VALUES
(3, 12000.00, 2);
```

```
INSERT INTO STUDENT_LOANS
VALUES
(4, 56000.00, 5);
INSERT INTO STUDENT_LOANS
VALUES
(5, 57200.00, 1);
```

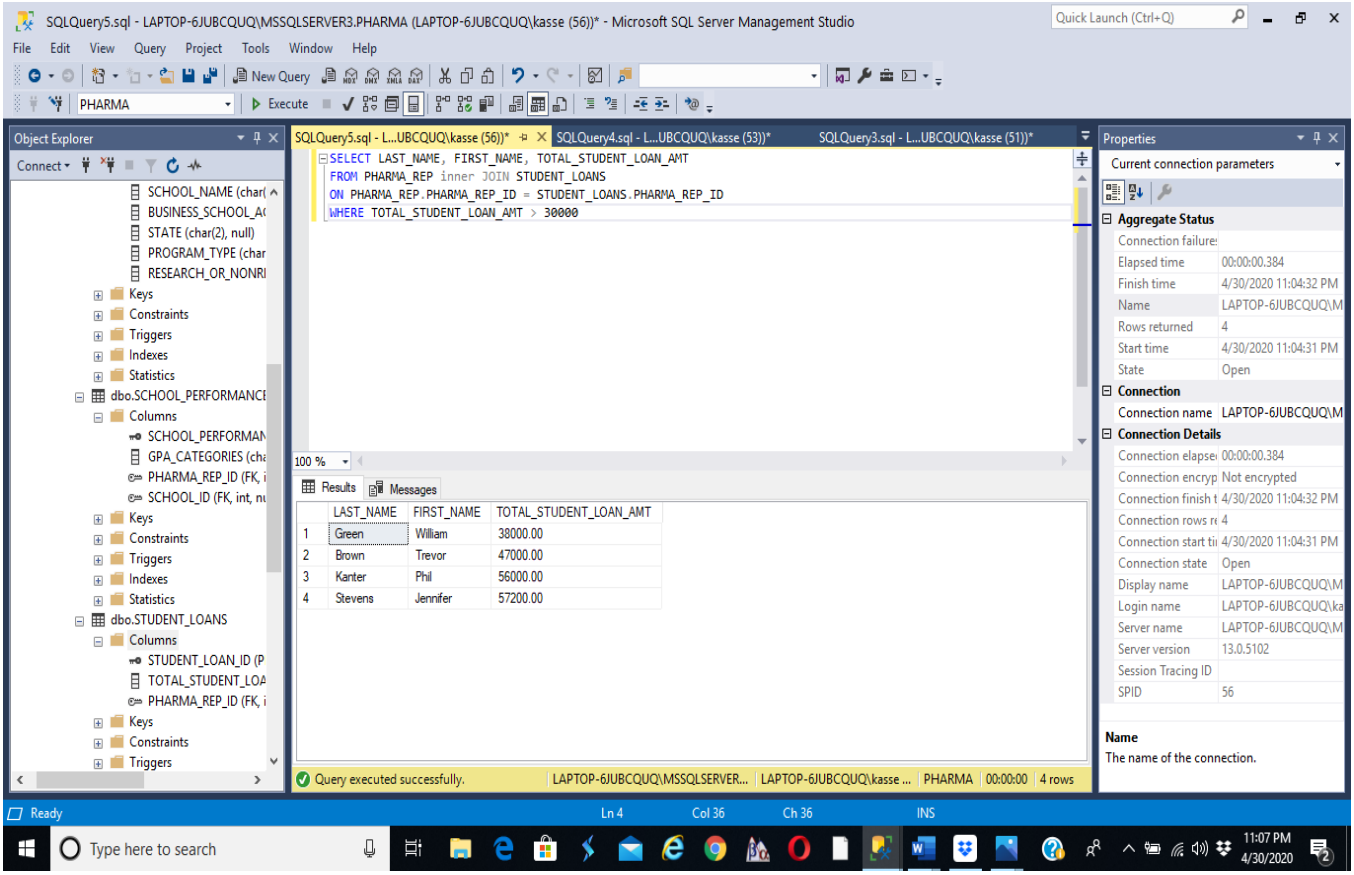


Student Loans Table



INNER JOIN with two tables

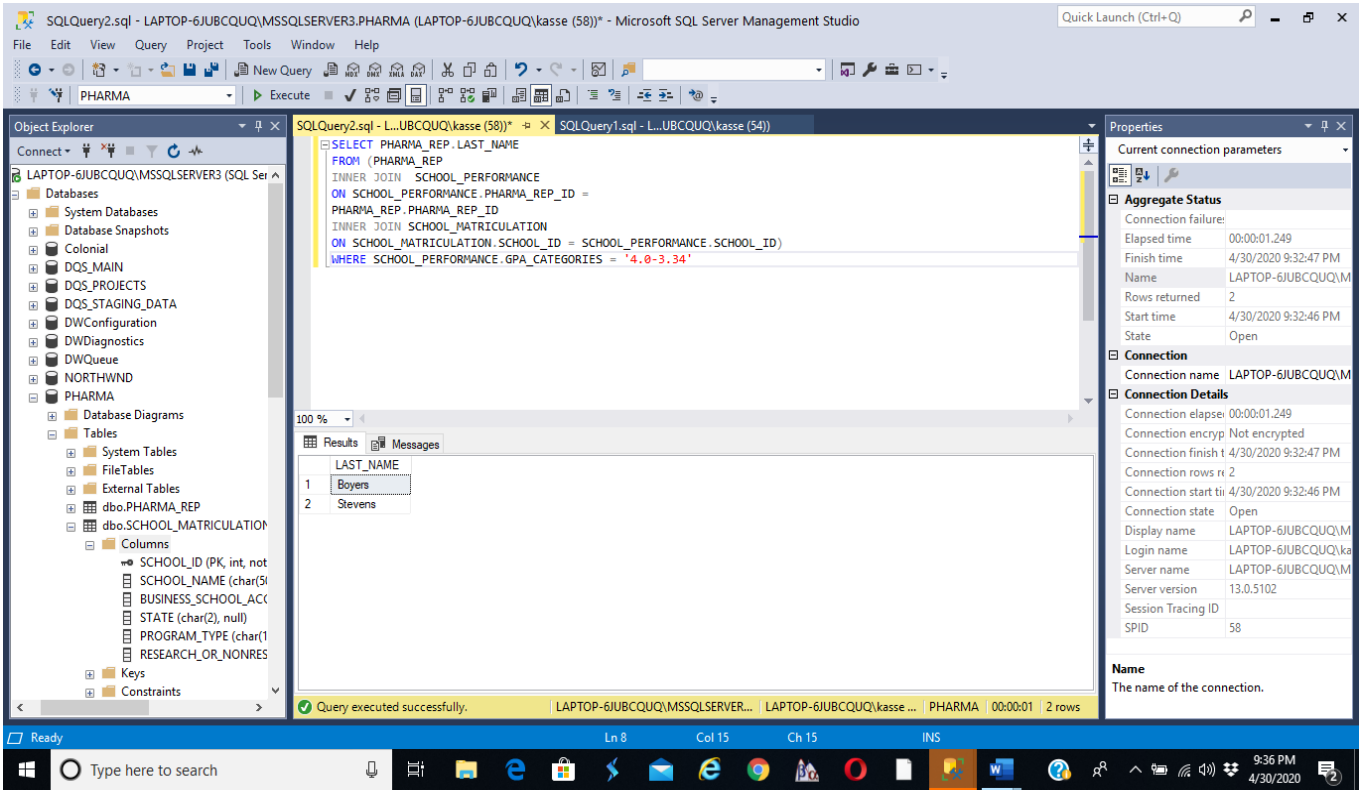
```
SELECT LAST_NAME, FIRST_NAME, TOTAL_STUDENT_LOAN_AMT
FROM PHARMA_REP inner JOIN STUDENT_LOANS
ON PHARMA_REP.PHARMA_REP_ID = STUDENT_LOANS.PHARMA_REP_ID
WHERE TOTAL_STUDENT_LOAN_AMT > 30000
```



This query is an inner join with two tables which include the pharma rep table and the student loans table. The purpose of this query is to list the first and last names of all the pharmaceutical representatives with student loans over $30,000. This is an inner join table that joins the tables together through the Pharma Rep ID. This table can be used in a corporation to determine loan payments if the reps are going to have their student loans taken out of their paycheck if they are over a certain amount, such as $30,000.

INNER JOIN With INTERSECTION TABLE

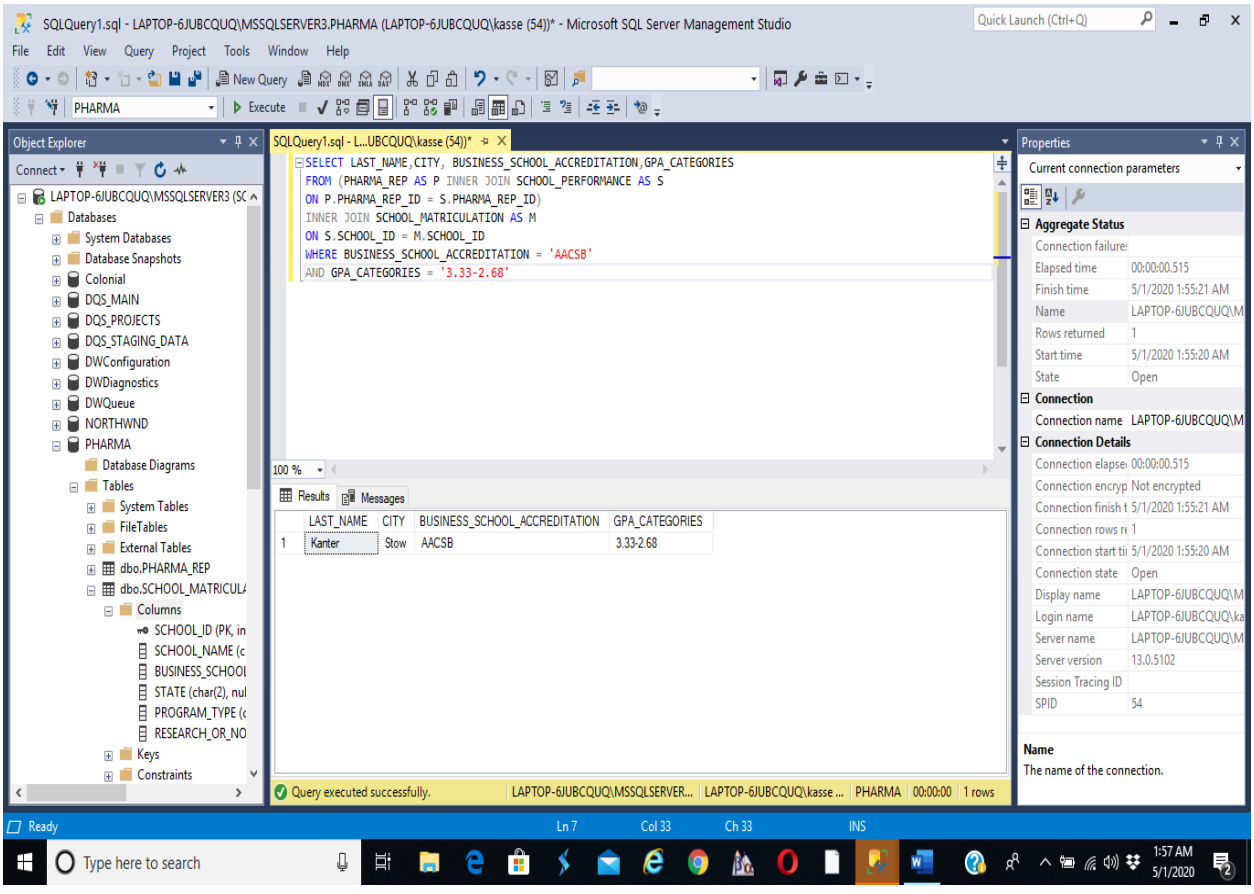
```
SELECT PHARMA_REP.LAST_NAME
FROM (PHARMA_REP
INNER JOIN  SCHOOL_PERFORMANCE
ON SCHOOL_PERFORMANCE.PHARMA_REP_ID =
PHARMA_REP.PHARMA_REP_ID
INNER JOIN SCHOOL_MATRICULATION
ON SCHOOL_MATRICULATION.SCHOOL_ID = SCHOOL_PERFORMANCE.SCHOOL_ID)
WHERE SCHOOL_PERFORMANCE.GPA_CATEGORIES = '4.0-3.34'
```



This is an inner join table with an intersection table. The purpose of this table is to list the last names of the reps with a 4.0 – 3.34 grade point average. This might be important to employers who want to know which employees have an A average for promotion purposes. This table was joined on the pharma rep id and the school id. The intersection tables can also be used to do compound searches with the and clause. I have also included a query with an and clause as well.

INNER JOIN WITH INTERSECTION TABLE

```
SELECT LAST_NAME,CITY, BUSINESS_SCHOOL_ACCREDITATION,GPA_CATEGORIES
FROM (PHARMA_REP AS P INNER JOIN SCHOOL_PERFORMANCE AS S
ON P.PHARMA_REP_ID = S.PHARMA_REP_ID)
INNER JOIN SCHOOL_MATRICULATION AS M
ON S.SCHOOL_ID = M.SCHOOL_ID
WHERE BUSINESS_SCHOOL_ACCREDITATION = 'AACSB'
AND GPA_CATEGORIES = '3.33-2.68'
```

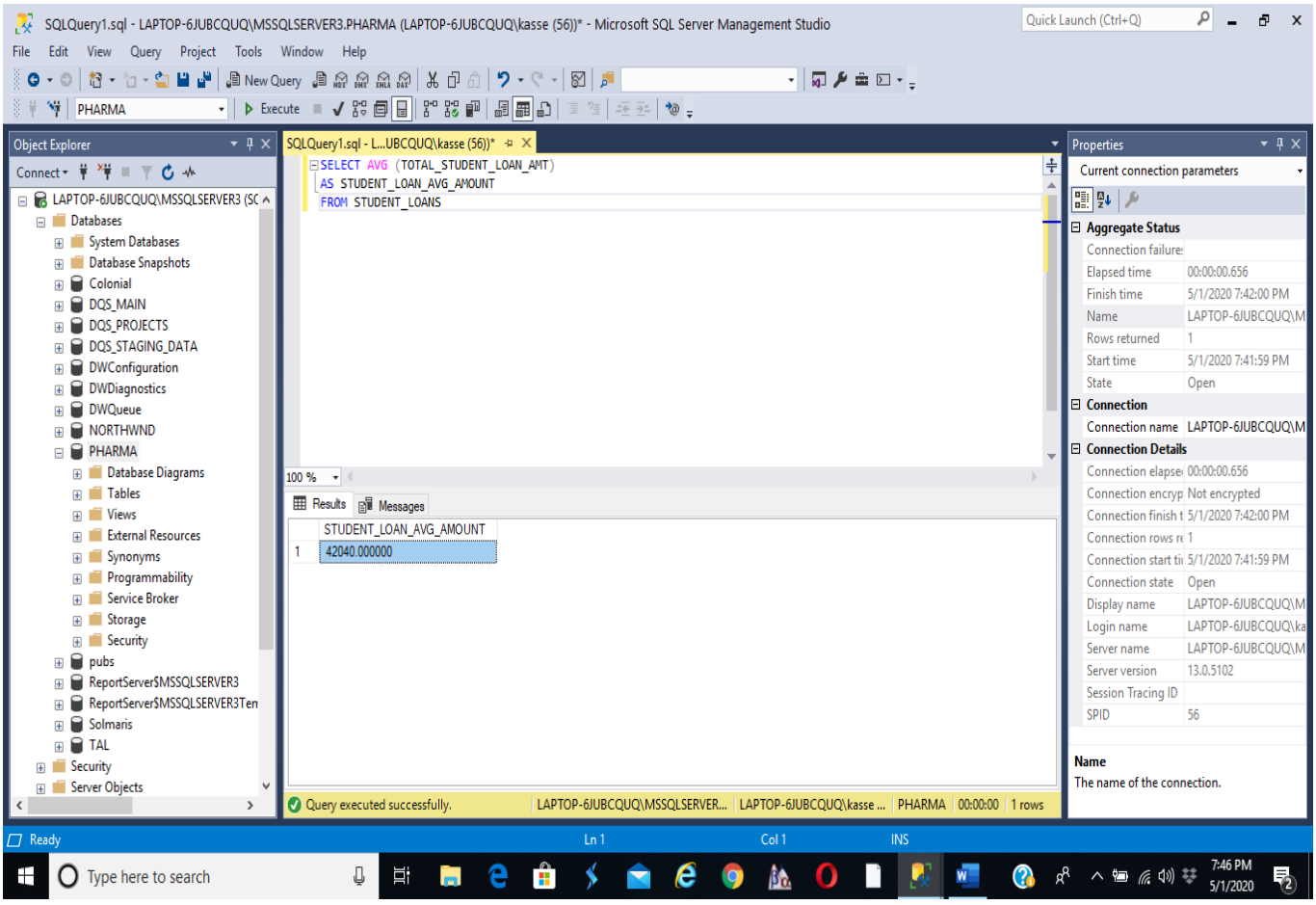


This is an intersection table inner join query that uses an alias to make the query easier to construct. This query lists last name, city, business school accreditation and GPA for the salespeople with a 3.33-2.68 grade point average. This query might be important to determine those salespeople with a B average but who attended an AACSB school which is considered to have an extremely high level of rigor by some standards. An employer might use this information to consider employees for promotions who might not be considered if only those employees with an A average were being given consideration, for

example. The city where the salesperson lives might also be important to see if it is nearby an open territory where there is a promotion opportunity, for example.

Aggregate function queries

```
SELECT AVG (TOTAL_STUDENT_LOAN_AMT)
AS STUDENT_LOAN_AVG_AMOUNT
FROM STUDENT_LOANS
```

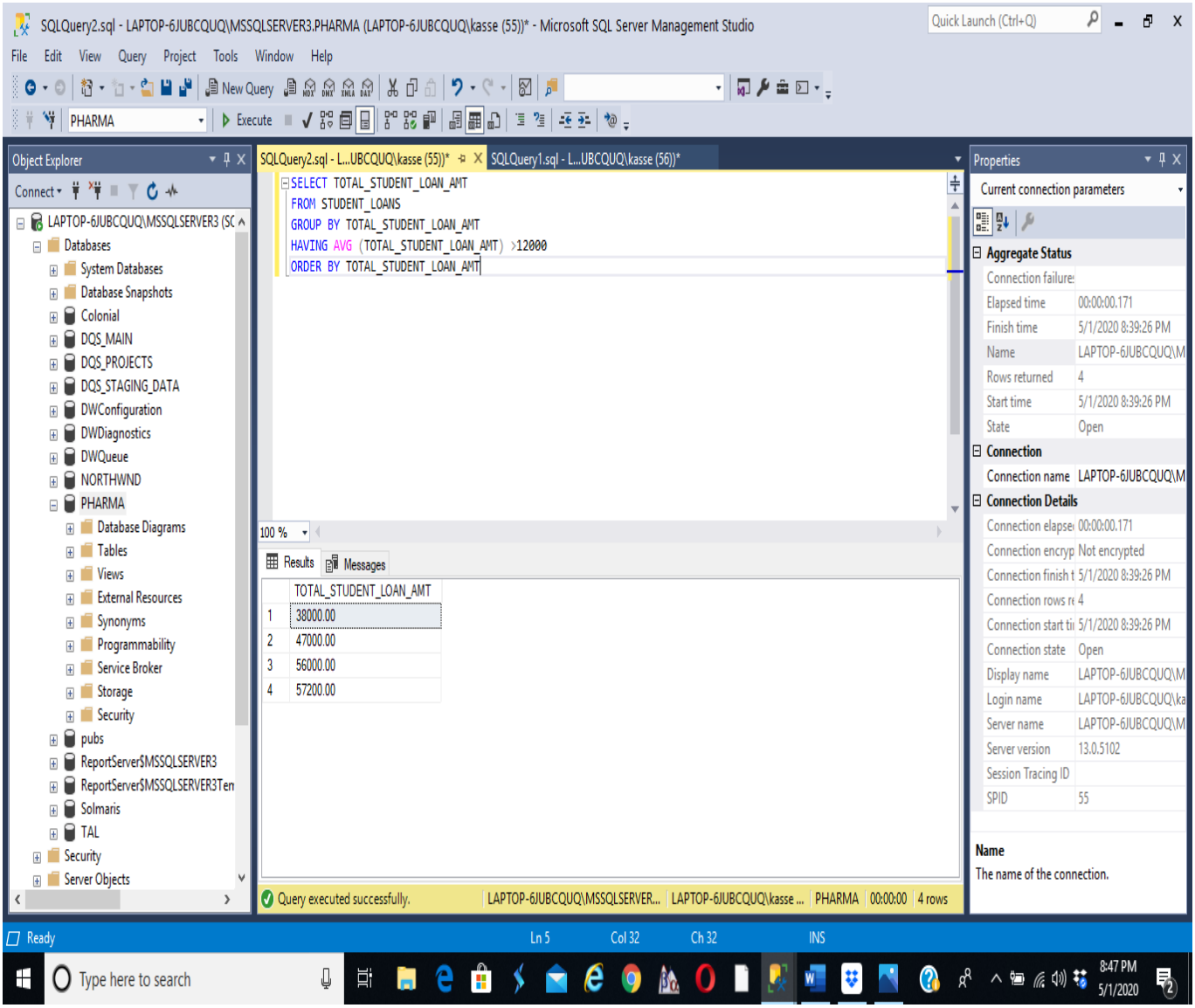


This was an aggregate function query which provided the average amount of all total student loan amounts in the database. What was great about this query is that you were able to name the result with a label of your choice. This information can be used in the corporate world for employers to get an idea of the average amount of debt employees are taking on to earn their degrees. This information could also be broken down by degree level as well to get an idea of the average amount of debt by degree level if a relationship were created between student loans and school matriculation. Employers

might also want to use this information to design scholarships for students going into pharmaceutical sales to defray the cost of schooling based on scholarship amounts that reflected the average amount of debt incurred by employees to earn their degrees.

```
AGGREGATE function with A HAVING clause

SELECT TOTAL_STUDENT_LOAN_AMT
FROM STUDENT_LOANS
GROUP BY TOTAL_STUDENT_LOAN_AMT
HAVING AVG (TOTAL_STUDENT_LOAN_AMT) >12000
ORDER BY TOTAL_STUDENT_LOAN_AMT
```

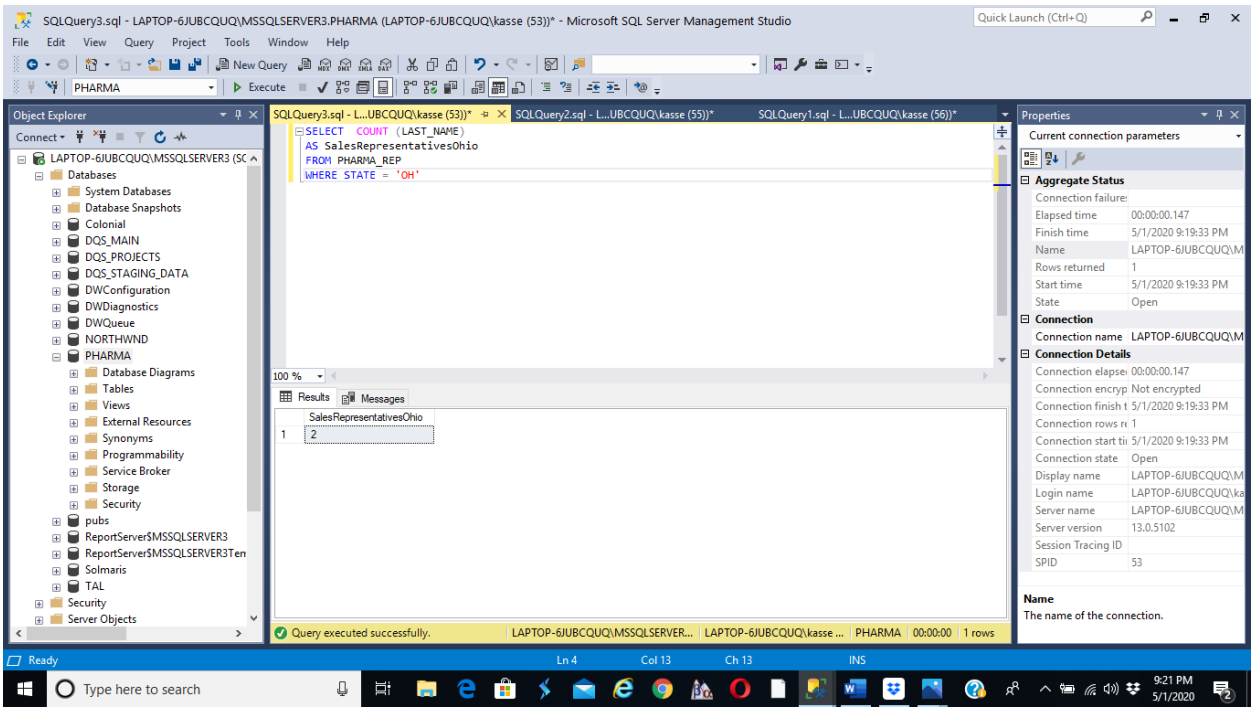


AGGREGATE Function with a HAVING clause

This is a HAVING clause with an aggregate function that lists student loan amounts from smallest to largest for student loan amounts greater than $12,000. This query can be used by employers to rank order Student Loans over a certain amount. In a large database, a query such as this would show a range of amounts.

Aggregate function with Count function

```
SELECT  COUNT (LAST_NAME)
AS SalesRepresentativesOhio
FROM PHARMA_REP
WHERE STATE = 'OH'
```



This query was used to demonstrate how the count function can be used to count the number of sales representatives in each state. This could be done on a state by state basis and then sales in each state could be compared if a sales table were also created, which would enable the ability to determine sales by territory versus GPA, for example. This query as it exists would be used to determine how many reps are in each state, which would enable employers to compare states with similar sales records to determine if more sales representatives should be hired in each state.