**Major Assignment \_01**

**Database Modeling and Query**

Prepared for

Ms. Hongyu Guo

Course Instructor

Prepared by

Md Towhidul Islam

CDBM-603- Enterprise Data Architecture

Sask atchewan Polytechnic

March 26, 2025

Table of Contents

[Section 1: Normalization 3](#_Toc193898746)

[1. Bringing the table into First Normal Form (1NF) 3](#_Toc193898747)

[2. Bringing the table into Second Normal Form (2NF) 3](#_Toc193898748)

[3. Bringing the table into Third Normal Form (3NF) 5](#_Toc193898749)

[Answer to the four questions after normalizing the database 6](#_Toc193898750)

[Question 1 6](#_Toc193898751)

[Question 2 7](#_Toc193898752)

[Question 3 7](#_Toc193898753)

[Question 4 8](#_Toc193898754)

[Section 2: Entity Relationship Diagram (ERD) 9](#_Toc193898755)

[Section 3: Queries 10](#_Toc193898756)

[Query 1 10](#_Toc193898757)

[Query 2 10](#_Toc193898758)

[Query 3 11](#_Toc193898759)

[Query 4 11](#_Toc193898760)

[Query 5 12](#_Toc193898761)

[Query 6 12](#_Toc193898762)

[Query 7 13](#_Toc193898763)

[Query 8 13](#_Toc193898764)

[Query 9 14](#_Toc193898765)

[Query 10 14](#_Toc193898766)

# Section 1: Normalization

Column names in our table of the Employees excel sheet:

'EmpId', 'SIN', 'Last', 'First', 'Street', 'City', 'Prov', 'Postal', 'Job Code', 'Position', 'Payrate', 'Inc. Tax', 'Birth Date', 'Hire Date', 'Pay Week End Date', 'Days Available', 'Hours', 'OT', 'Person Hours Worked', 'Supervisor', 'Supervisor Cell#'

## Bringing the table into First Normal Form (1NF)

To bring the table into **First Normal Form (1NF)**, the goal is to ensure that:

1. **Atomicity**: Each column must contain only atomic (indivisible) values. No arrays, lists, or composite fields are allowed.
2. **Uniqueness**: Each row must have a unique identifier (primary key).
3. **No Repeating Groups**: Remove any repeating groups of columns or data.

**Conversion Process from 0NF to 1NF:**

In our 0NF table the columns ‘Pay Week End Date', 'Days Available', 'Hours', 'OT', 'Person Hours Worked' don’t contain atomic value. So we are separating those.

Now, each row must have a unique identifier. So our 'EmpId' & 'Pay Week End Date' together (composite key) form a unique key.

There are no repeating groups.

## Bringing the table into Second Normal Form (2NF)

**To bring the table into Second Normal Form (2NF), we must ensure:**

1. The table is already in **1NF**.
2. **No partial dependency** exists: All non-key attributes must depend on the entire primary key, not just part of it.

**Conversion Process from 1NF to 2NF:**

Our table is already in the 1NF.

As our primary in 1NF is 'EmpId' & 'Pay Week End Date' together so we have to remove all partial key dependency. Means those non key attributes (columns) aren’t fully dependent on the primary key ('EmpId' & 'Pay Week End Date' together) we have separate those (scraping the repeating values) to make full key dependency. They still need to be related to each other, so we have to create a mapping (lookup) table in that case.

We can see only 'Hours' and 'OT' columns are fully dependent on primary key ('EmpId' & 'Pay Week End Date' together to as a composite key). So we separate those in a new table.

Then we can see 'Days Available' and 'Person Hours Worked' columns are dependent only on ‘Pay Week End Date’ column. So we moved those in a separate table.

Then we see the remaining columns ('SIN', 'Last', 'First', 'Street', 'City', 'Prov', 'Postal', 'Job Code', 'Position', 'Payrate', 'Inc. Tax', 'Birth Date', 'Hire Date', 'Supervisor', 'Supervisor Cell#') are dependent only on 'EmpId' column. So, we moved those in a separate table.

Also we removed any duplicate records from those tables.

**Set notation for those conversion:**

**After 1NF original table schema is below.**

R('EmpId', 'SIN', 'Last', 'First', 'Street', 'City', 'Prov', 'Postal', 'Job Code', 'Position', 'Payrate', 'Inc. Tax', 'Birth Date', 'Hire Date', 'Pay Week End Date', 'Days Available', 'Hours', 'OT', 'Person Hours Worked', 'Supervisor', 'Supervisor Cell#').

Here 'EmpId' and 'Pay Week End Date' together form a primary key.

**But we can see the following partial dependency.**

'EmpId' 'SIN', 'Last', 'First', 'Street', 'City', 'Prov', 'Postal', 'Job Code', 'Position', 'Payrate', 'Inc. Tax', 'Birth Date', 'Hire Date', 'Supervisor', 'Supervisor Cell#'

'Pay Week End Date' 'Days Available', 'Person Hours Worked'

**In 2NF, the schema is decomposed into:**

R1 ('EmpId', 'SIN', 'Last', 'First', 'Street', 'City', 'Prov', 'Postal', 'Job Code', 'Position', 'Payrate', 'Inc. Tax', 'Birth Date', 'Hire Date', 'Supervisor', 'Supervisor Cell#')

R2 ('Pay Week End Date', 'Days Available', 'Person Hours Worked')

R3 ('EmpId', 'Pay Week End Date', 'Hours', 'OT')

## Bringing the table into Third Normal Form (3NF)

To bring the table into **Third Normal Form (3NF)**, we need to ensure:

1. The table is already in **2NF**.
2. **No transitive dependency** exists: Non-key attributes should depend **only on the primary key**, not on other non-key attributes.
3. Also mapping (lookup) relationships among the tables using PK and FK has to be established.

**Conversion Process from 2NF to 3NF (Refer to Employees-2NF sheet of excel file):**

Our table is already in the 2NF.

We have 3 tables in our 2NF. These are PaymentDetails, PayWeekEndDates and Employees.

In PaymentDetails we don’t see any transitive dependency. So this table will remain the same.

In our PayWeekEndDates table we also don’t see any transitive dependency. But we see ‘Person\_Hours\_Worked’ column contains cumulative sum of hours of any given date. We can delete this column as this can be calculated from PaymentDetails table.

In our Employees table we see some transitive dependency. As ‘Position’ and ‘Payrate’ depend on ‘Job\_Code’. So we make a new table called Jobs. Also, ‘Supervisor’ and ‘Supervisor\_Cell#’ is separated into new table and attributed by ‘SupervisorID’ to avoid redundancy.

We then create a mapping table ‘EmpJobSupervisor’ to connect three tables (Employees, Supervisors and Jobs) together.

**Conversion Process from 2NF to 3NF (Refer to Committee sheet of excel file):**

Our table is already in 2NF.

To handle the membership of employees in multiple committees we create a new table EmployeesCommittees. This connects both the Committees and Employees table.

**Set notation for those conversion:**

**After 1NF original table schema is below.**

R1 ('EmpId', 'SIN', 'Last', 'First', 'Street', 'City', 'Prov', 'Postal', 'Job Code', 'Position', 'Payrate', 'Inc. Tax', 'Birth Date', 'Hire Date', 'Supervisor', 'Supervisor Cell#')

R2 ('Pay Week End Date', 'Days Available', 'Person Hours Worked')

R3 ('EmpId', 'Pay Week End Date', 'Hours', 'OT')

**But we can see the following transitive dependency.**

‘Job\_Code’ ‘Position’ and ‘Payrate’

‘Supervisor’ ‘Supervisor Cell#’

**In 3NF, the schema is decomposed into:**

Employees (‘EmpId’, ‘SIN’, ‘Last’, ‘First’, ‘Street’, ‘City’, ‘Prov’, ‘Postal’, ‘Birth\_Date’, ‘Hire\_Date’, ‘Inc.Tax’)

Supervisors (‘SupervisorId’, ‘Supervisor’, ‘Supervisor Cell#’)

Jobs (‘Job\_Code’, ‘Position’, ‘Payrate’)

PayWeekEndDates (‘Pay\_Week\_End\_Date’, ‘Days\_Available’)

PaymentDetails (‘EmpId’, ‘Pay\_Week\_End\_Date’, ‘Hours’, ‘OT’)

EmpJobSupervisor (‘EmpId’, ‘Job\_Code’, ‘SupervisorID’)

Committees (‘CommitteeId’, ‘Committee\_Name’, ‘Meeting\_Nights’)

**\* For detail table with data check the attached excel file \***

## Answer to the four questions after normalizing the database

### Question 1

**There is a concern that the spreadsheet will get out of hand, as the number of pay periods increases; how does your 3NF solve this issue?**

**Answer:**

Microsoft Excel has a limitation in number of columns and rows. In Excel 2007 and later (including Excel for Office 365) maximum number of rows can be 1,048,576. So if pay period increase it will increase uses of number of rows in excel sheet. Ultimately at one point there will be no room to enter the data in the excel sheet. On the other hand there is no row limit in database. So it can handle unlimited data until storage permits.

Also after applying 3rd normal form we can avoid updating of redundant data in each record. In ‘PaymentDetails‘ table of our 3NF we can see there are only 4 columns. So for each pay period record only those 4 columns needs need to be updated.

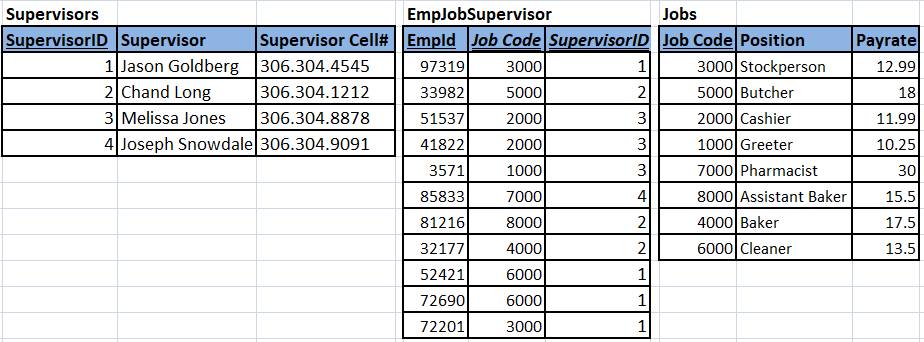
### Question 2

Supervisors often manage several departments; for example, Chad Long is managing butchers and bakers. How did you address this problem?

**Answer:**

To solve this problem we took help of mapping (lookup) table in our 3NF.

We created a separate table named ‘Supervisor’ which contains ‘SupervisorID’. Also for the jobs we created another tabe named ‘Jobs’ where each job is denoted by corresponding Job\_Code. Then we created another mapping (lookup) table named ‘EmpJobSupervisor’ that connects those two tables together with one to many relationship and ‘Employees’ table with 1 to 1 relationship. Below are the snippets of those 3 tables for reference.

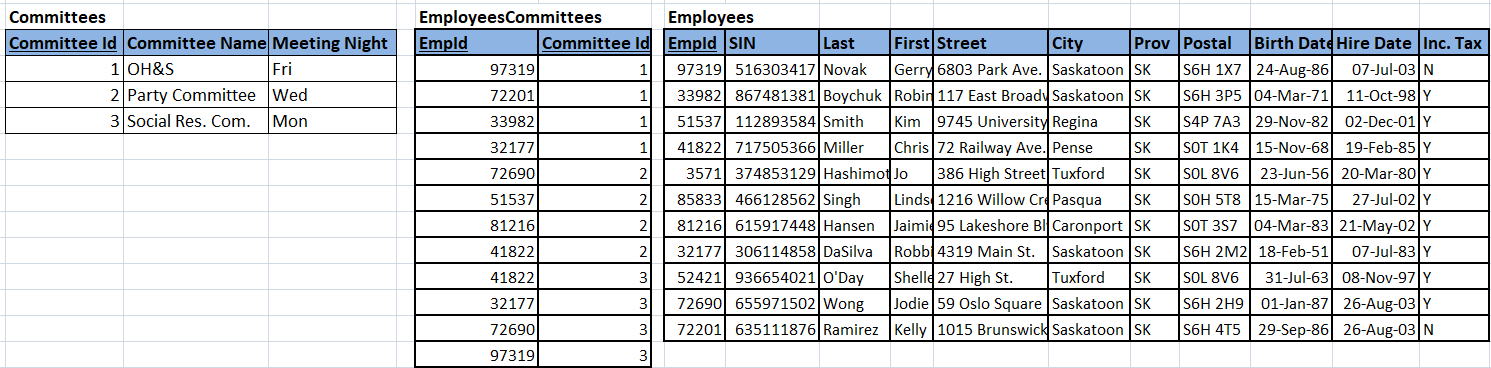


### Question 3

Employees often serve on multiple committees at once. How did you address this?

**Answer:**

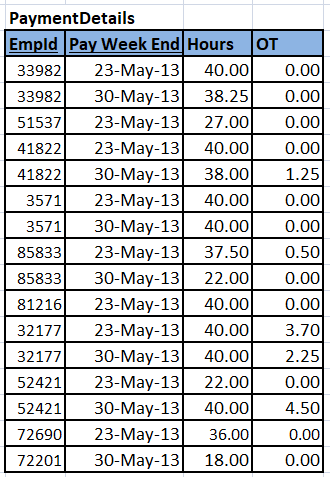
To address this issue we created another mapping (lookup) table ‘EmployeesCommittees’. This table connects both ‘Employees’ and ‘Committees’ tables in 1 to many relationship. Below are the snippets of those 3 tables.



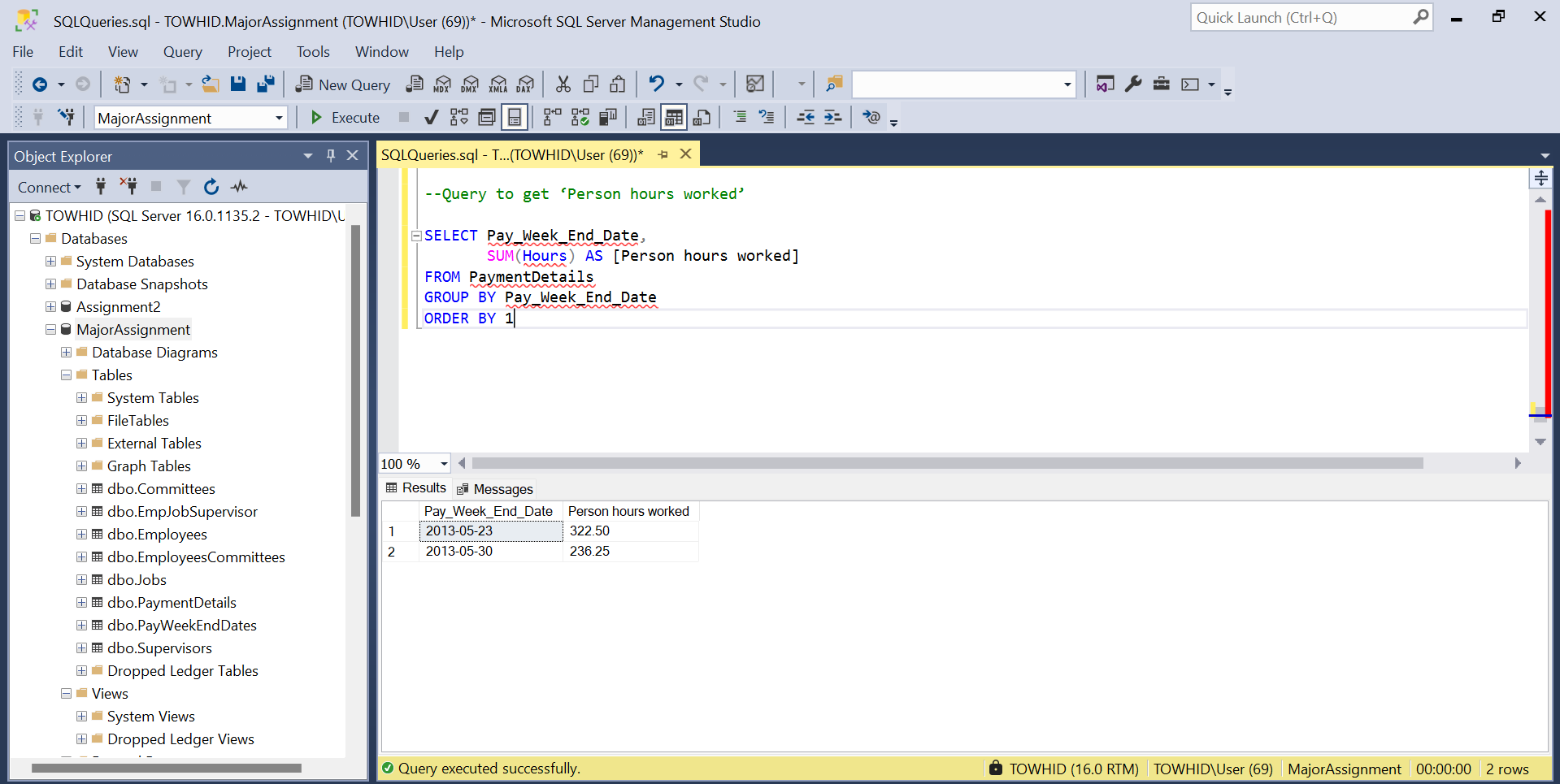
### Question 4

‘Person hours worked’ is calculated (it represents information, not data) for all hours worked by all employees for that week. Is there a better way to handle this rather than record it in the database? Implement your solution in the queries section.

**Answer:**

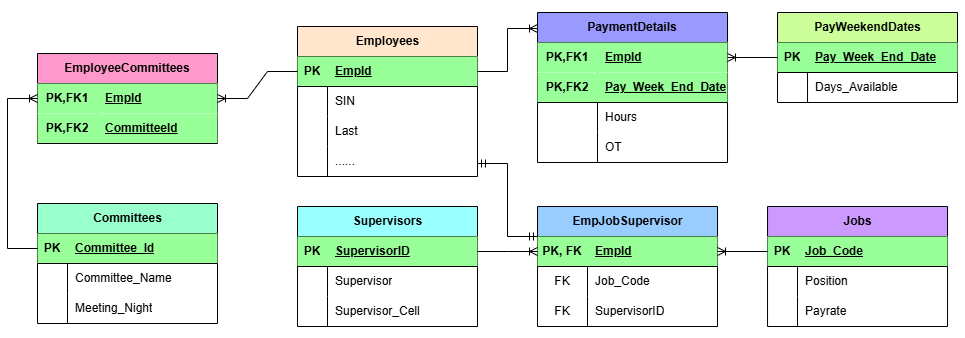
To handle this ‘Person hours worked’ information better we have deleted that column from the table. As we can easily get that information by querying our ‘PaymentDetails’ table. That table is shown below.

Using query we can extract ‘Person hours worked’ information. Below is the screenshot of the query and result.



# Section 2: Entity Relationship Diagram (ERD)

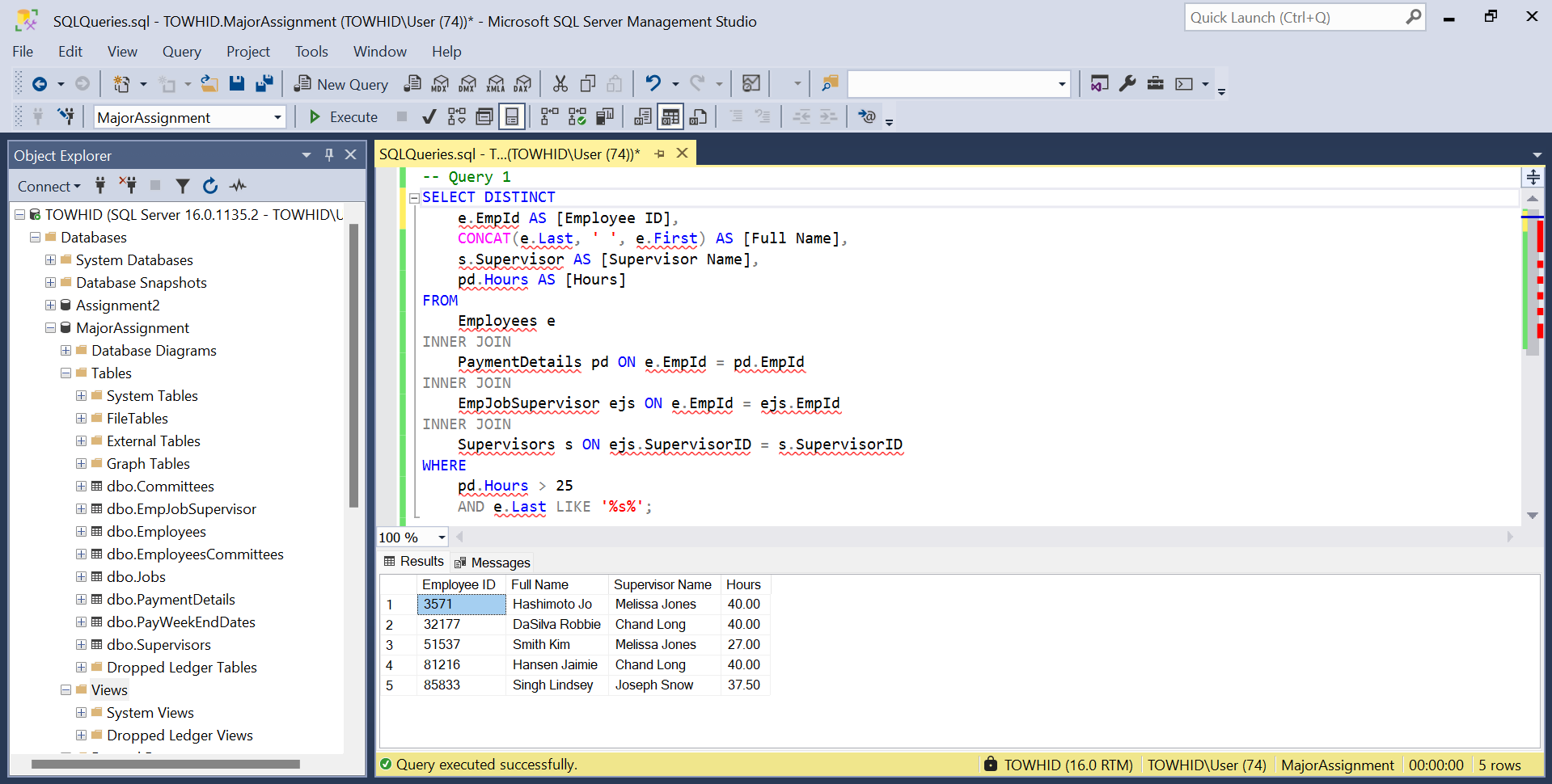
Below is the full physical ERD diagram of our database. This ERD diagram depicts Primary Key (PK), Foreign Key (FK), 1-to-1, 1-to-many, and many-to-many relationships using Crowfoot notations.



# Section 3: Queries

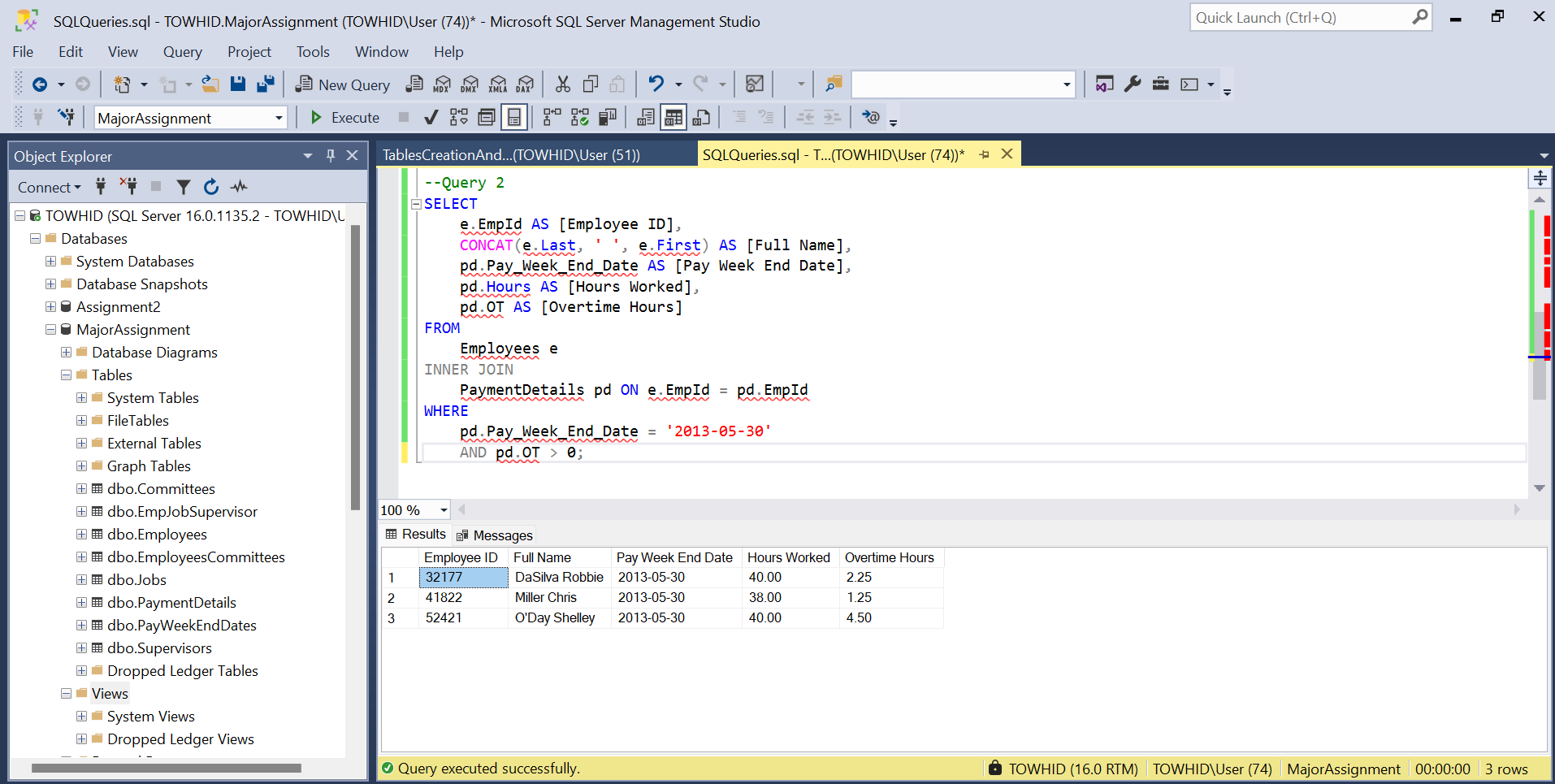
## Query 1

* + 1. Who worked more than 25 hours in any pay period and the employee’s last name includes the character 's'? Please print the employee's [employee ID], [Full name] (Last name + First Name) as ‘Full Name’, [Supervisor Name], [Hours].



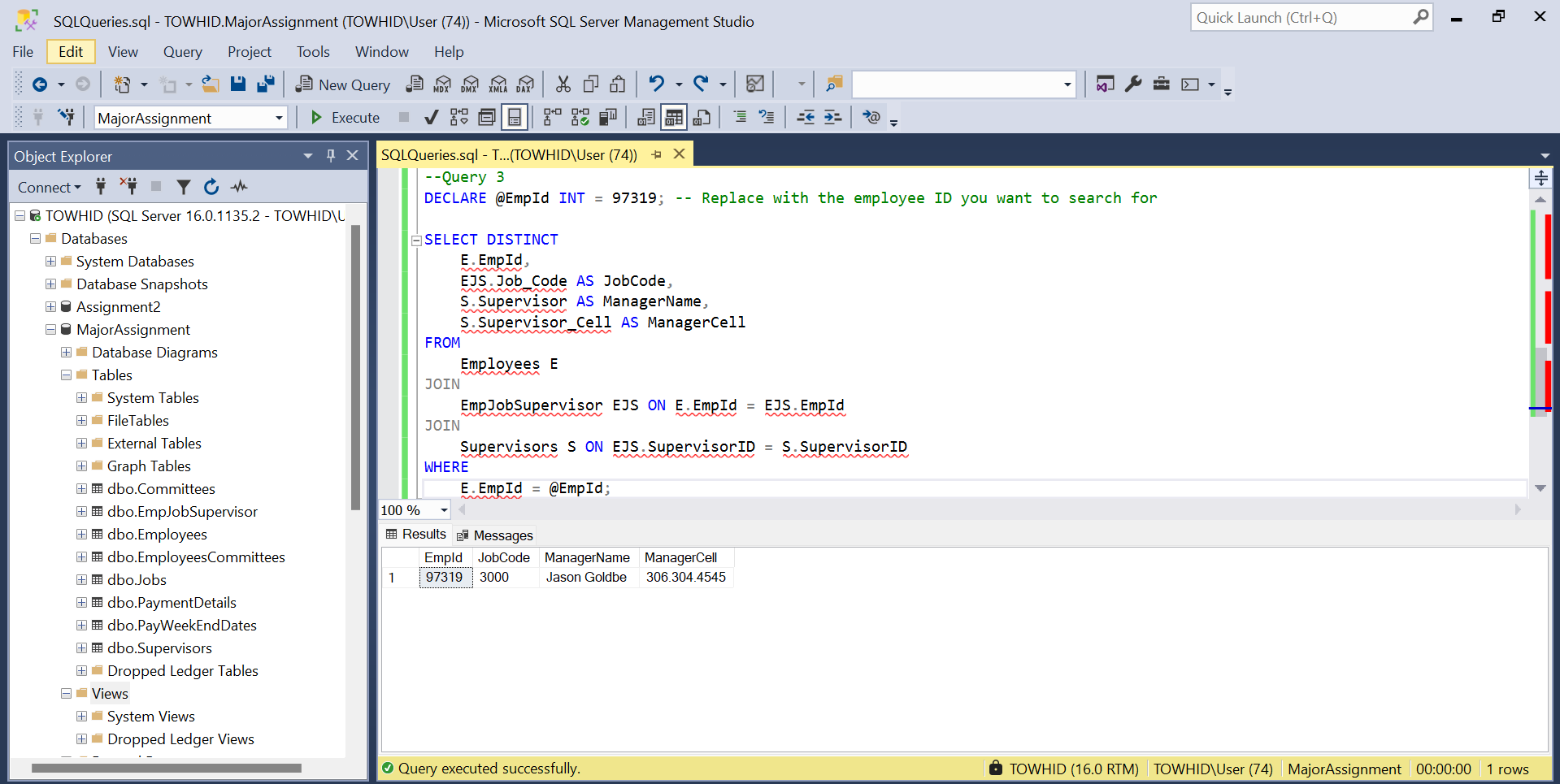
## Query 2

* + 1. Who worked overtime hours in the 30-May-13 pay period.



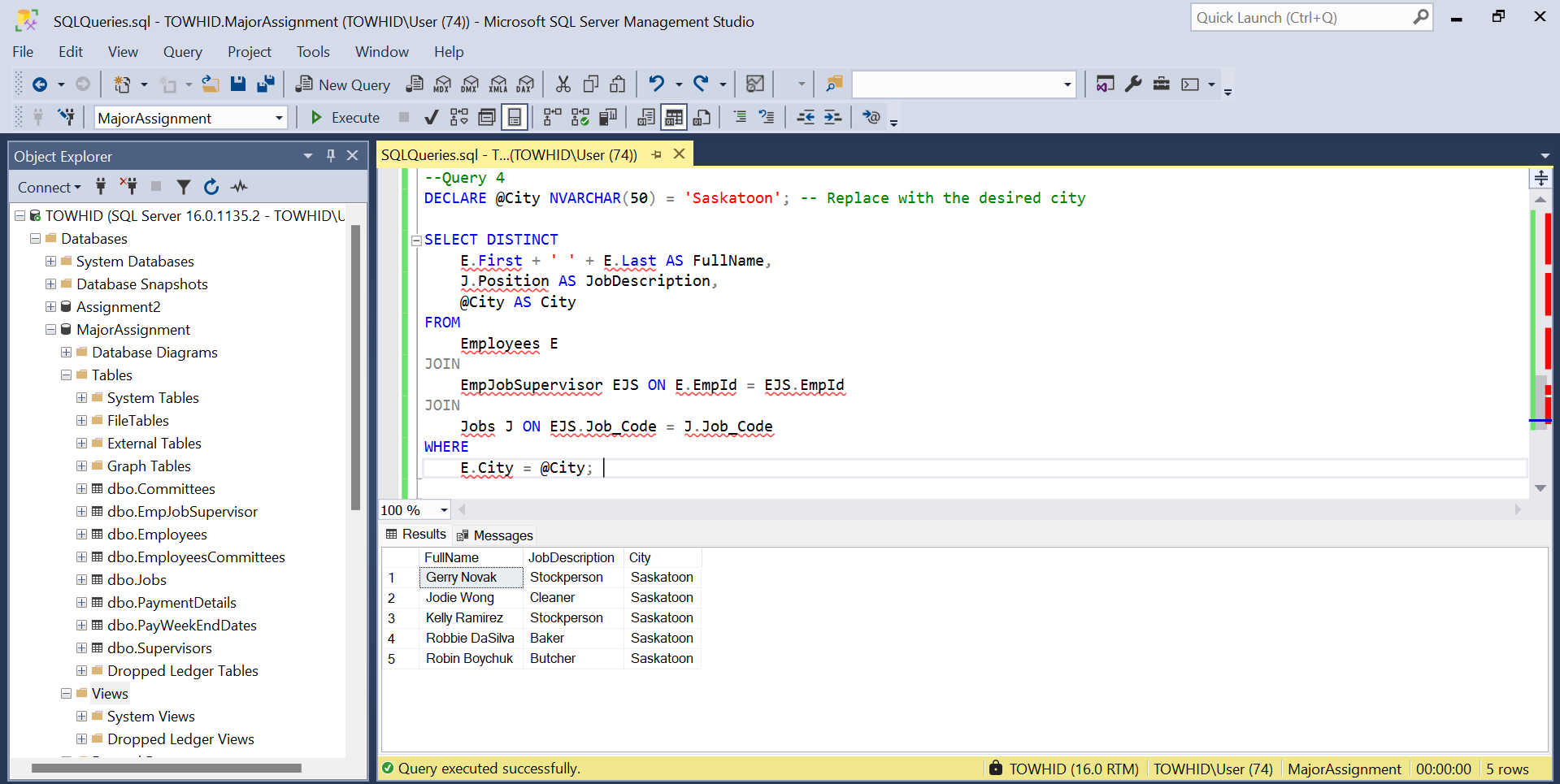
## Query 3

* + 1. I want to enter an employee id and find their job code, their manager’s name and their manager’s cell number. (No duplicate records)



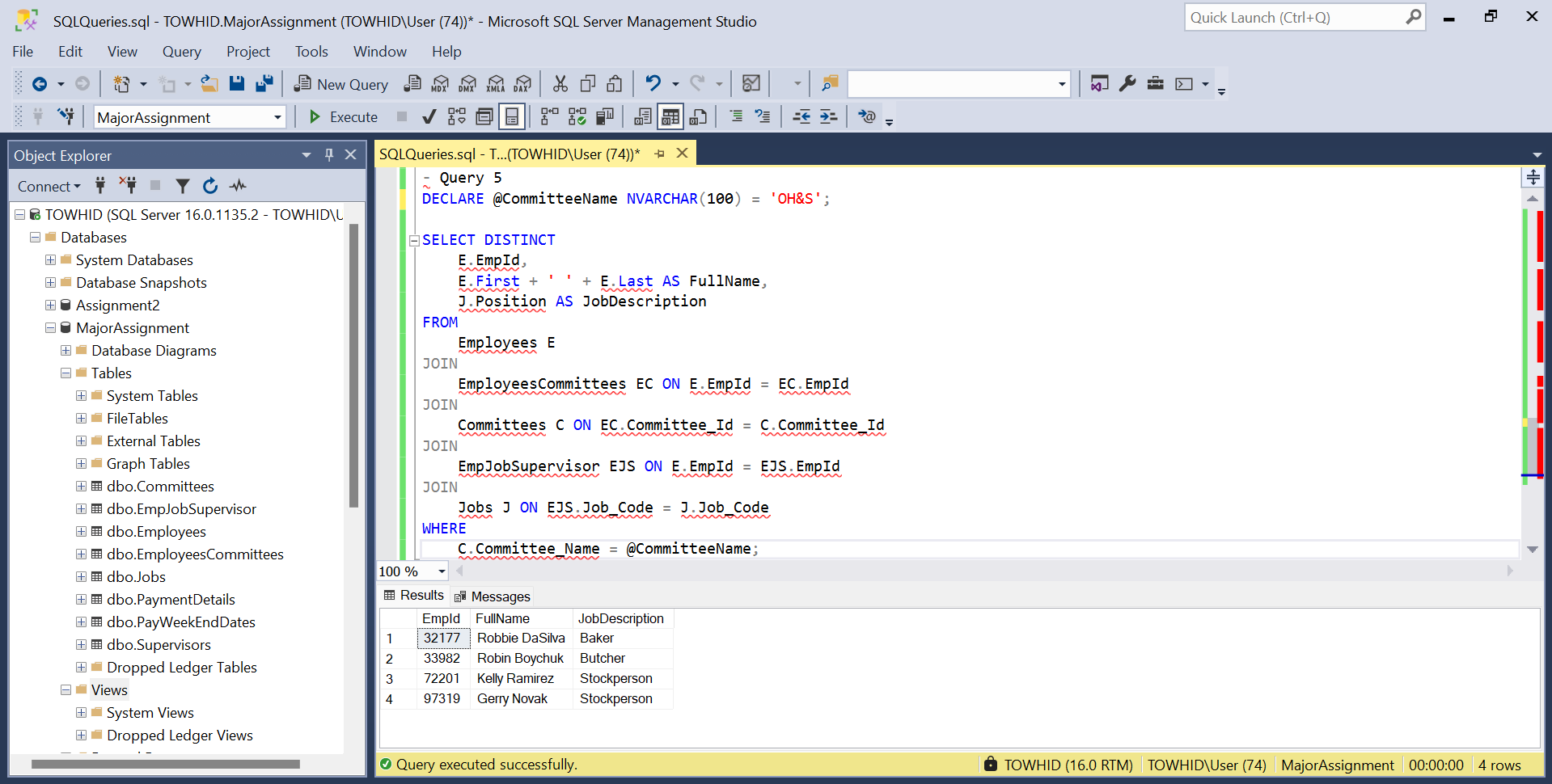
## Query 4

* + 1. I’d like to know the skill sets we have by city. I want to enter a city and get a list of the people who live there and their job descriptions.



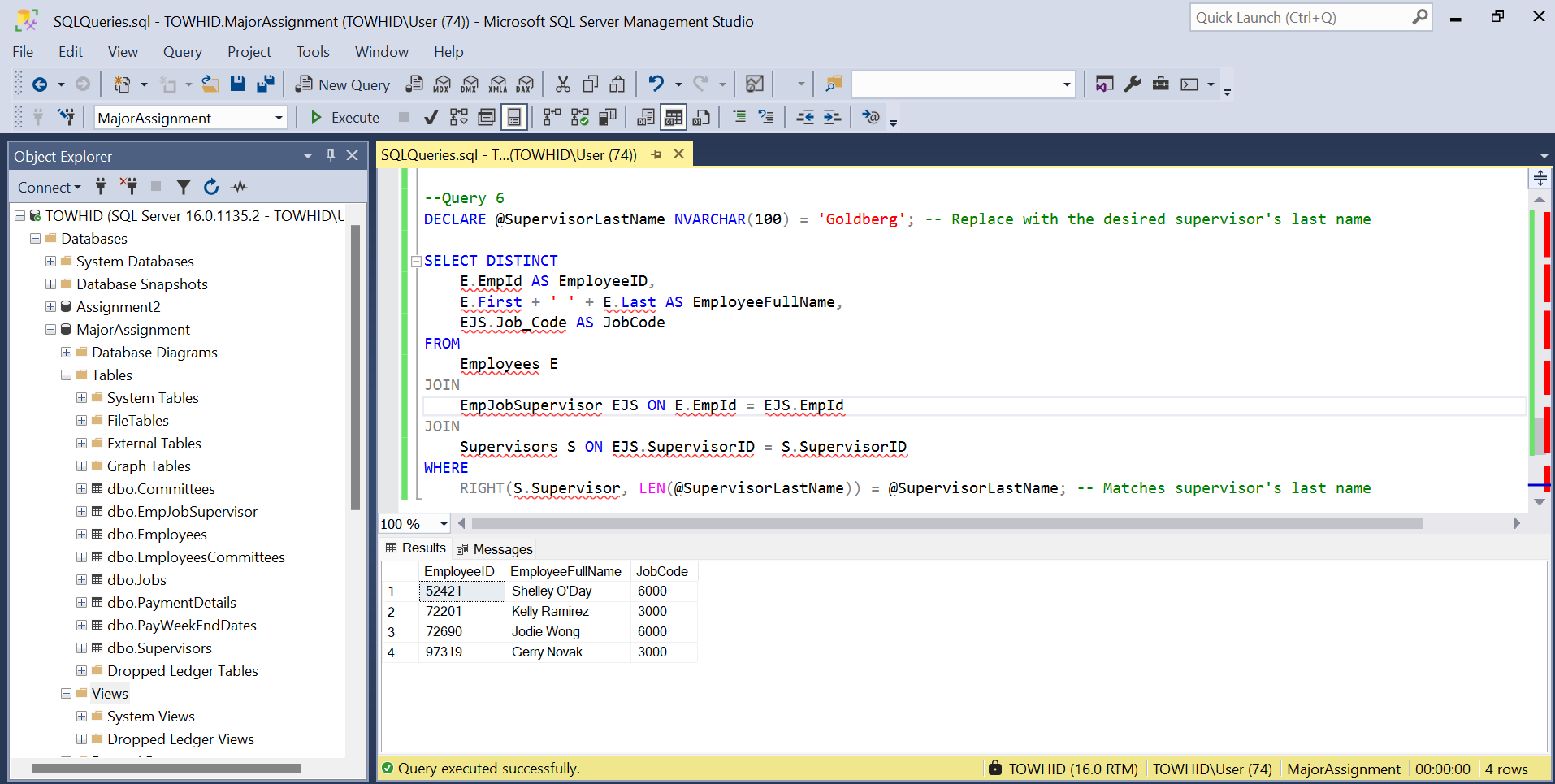
## Query 5

* + 1. Given a committee name 'OH&S', I’d like a list of people who work on it and their job description.



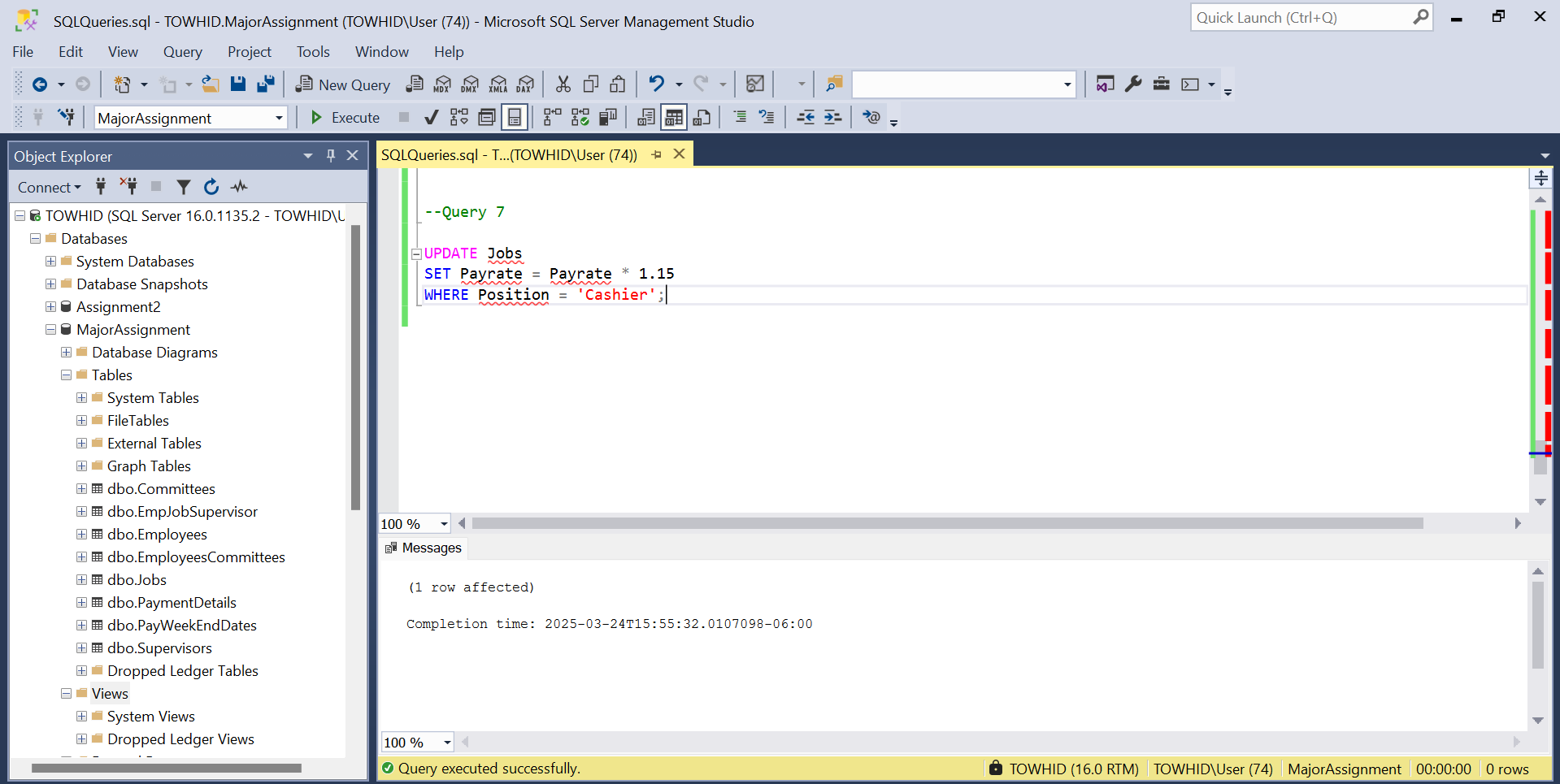
## Query 6

* + 1. Given a supervisor’s last name ‘Goldberg’ , provide a list of employees being supervised and their job code.



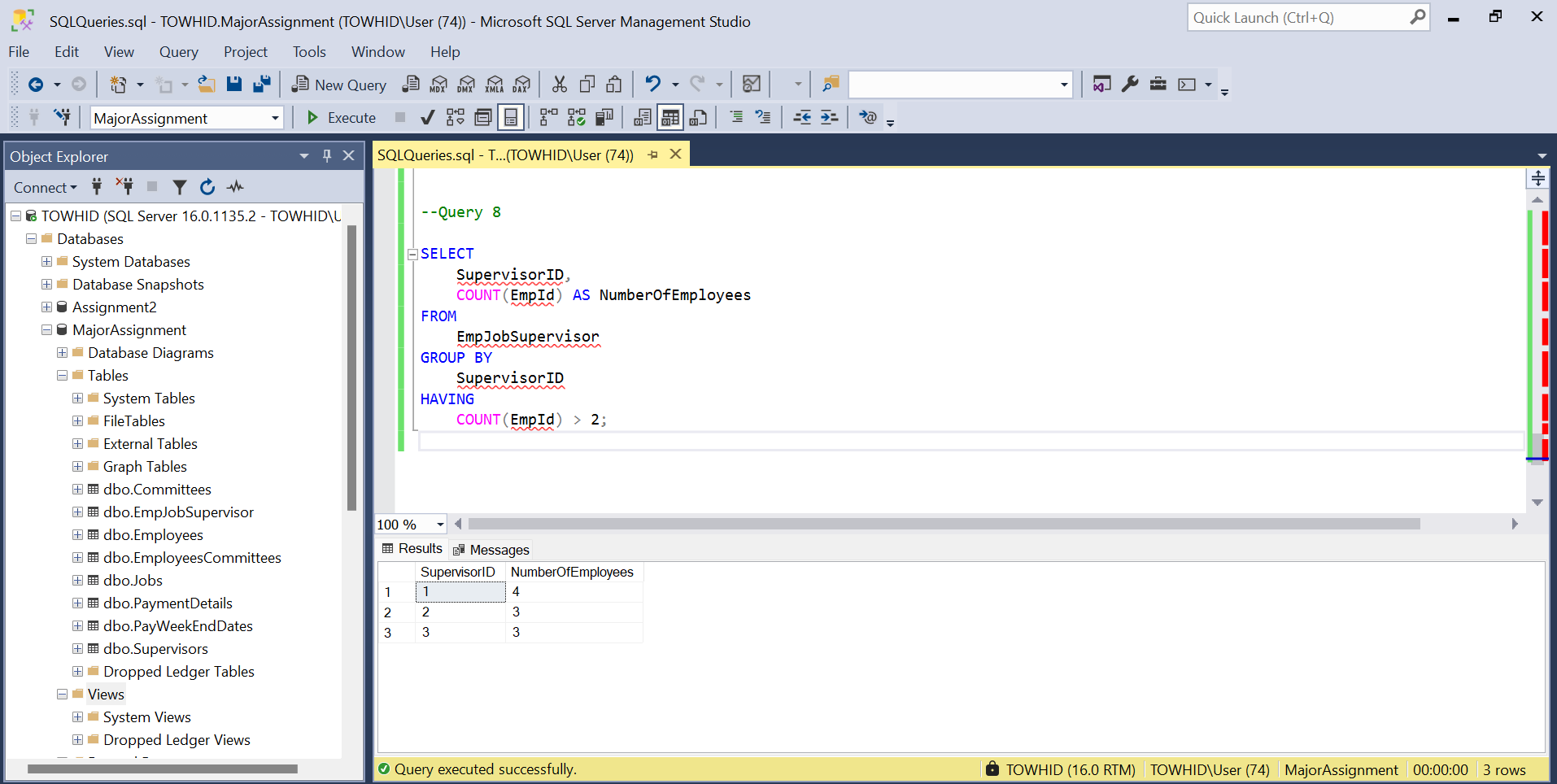
## Query 7

* + 1. Increase the pay rate by 15% for employees with the position title ‘Cashier’.



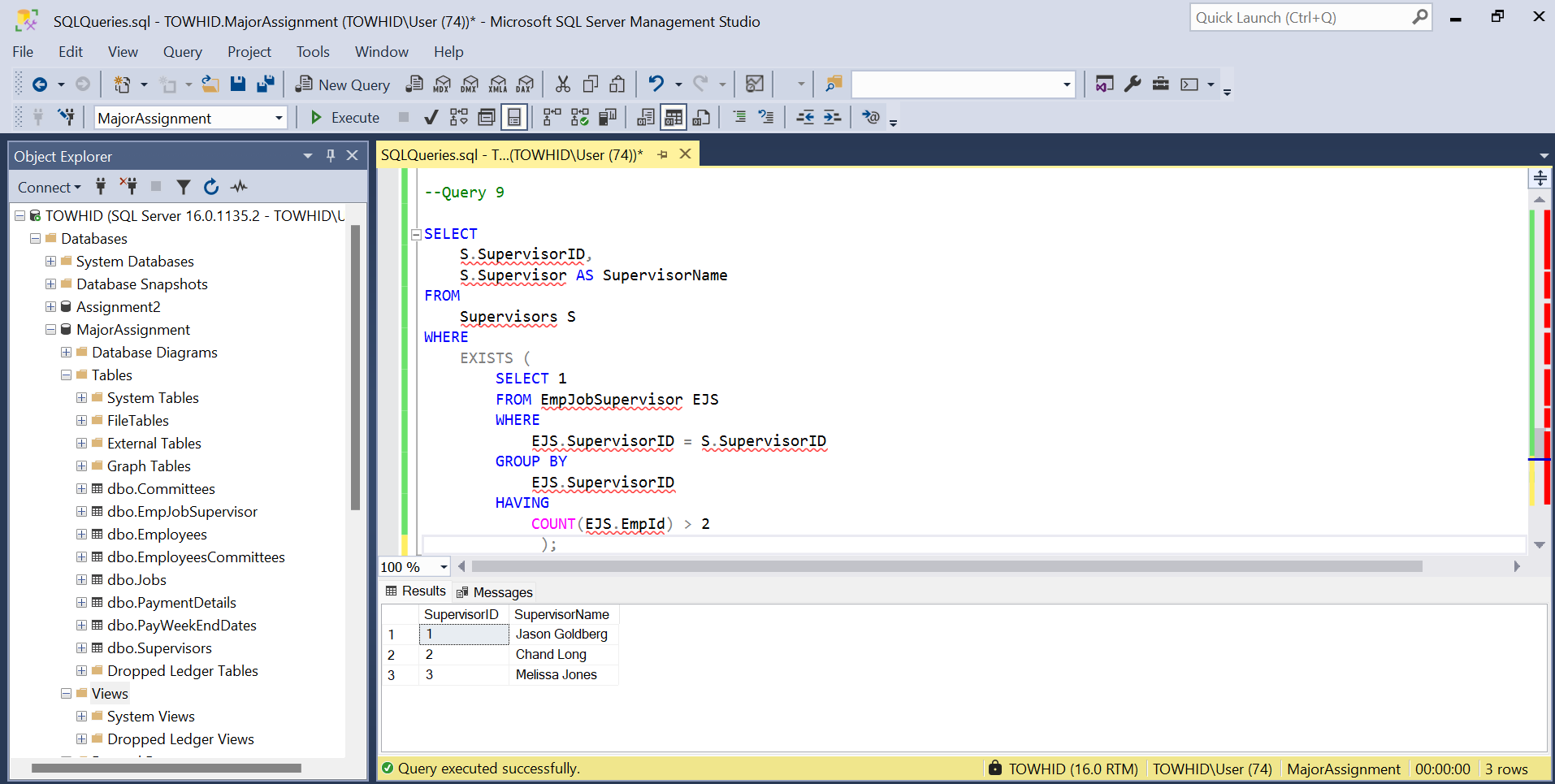
## Query 8

* + 1. List the number of employees under each supervisor. Only include the supervisor with more than 2 employees. Print out [SupervisorID],[Number of Employees].



## Query 9

* + 1. List the supervisor’s ID and name, who supervises more than 2 employees. (hint: using EXISTS clause)



## Query 10

* + 1. Create a view to show the employee ID, employee’s full name, committee ID, meeting night, committee name, supervisor id and supervisor’s name, where the condition is the meeting night is ‘Fri’. No duplicate records.

