

# IT Assignment Coversheet

**Course**: PROG8080 – Database Management

Program Coordinator: David Allison

Professor/Instructor: Mark Morell

Assignment #: Choose from list

Assignment Type:  Individual  Pair  Team

Date Submitted:

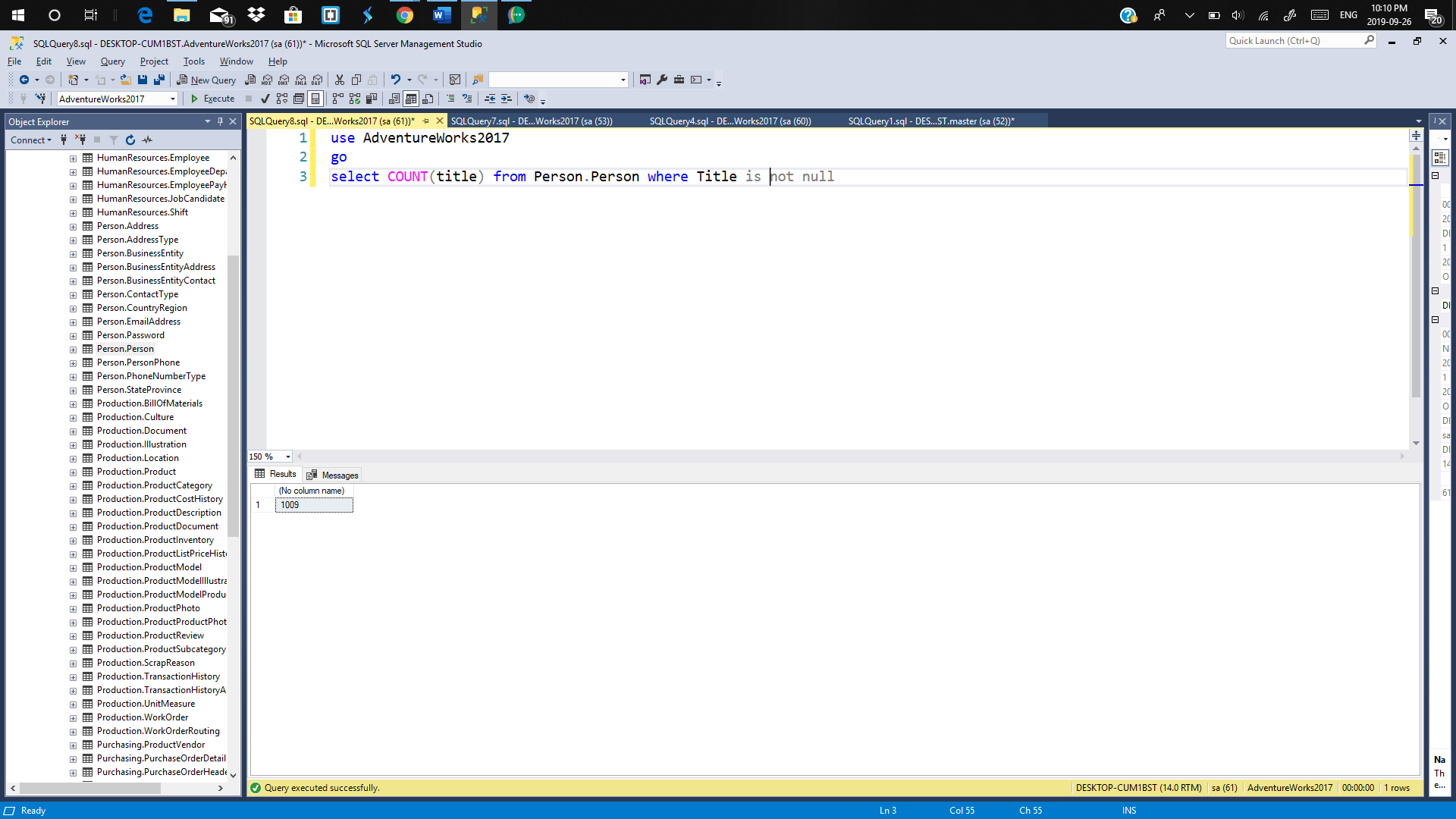
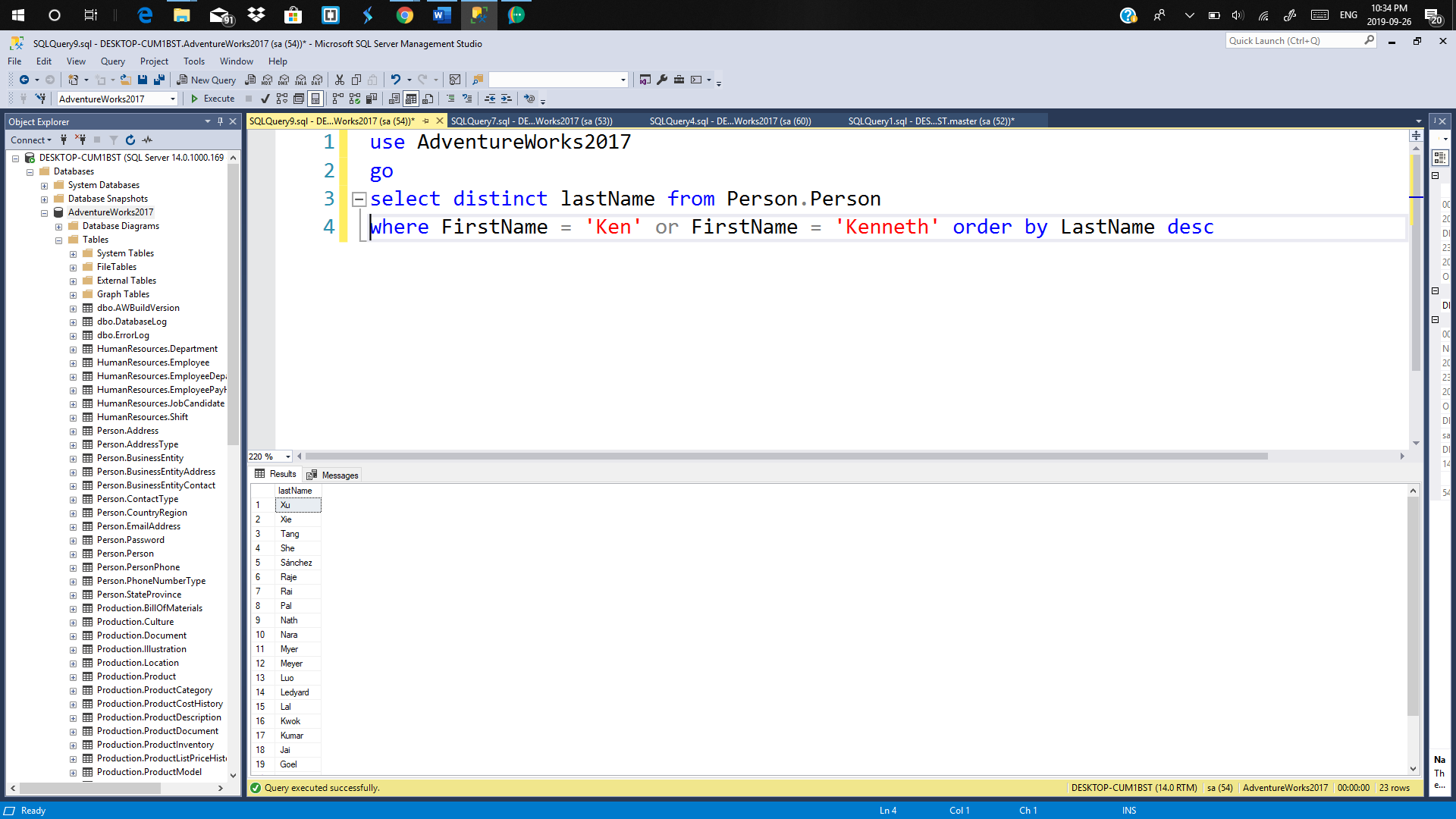
# **Student Information**

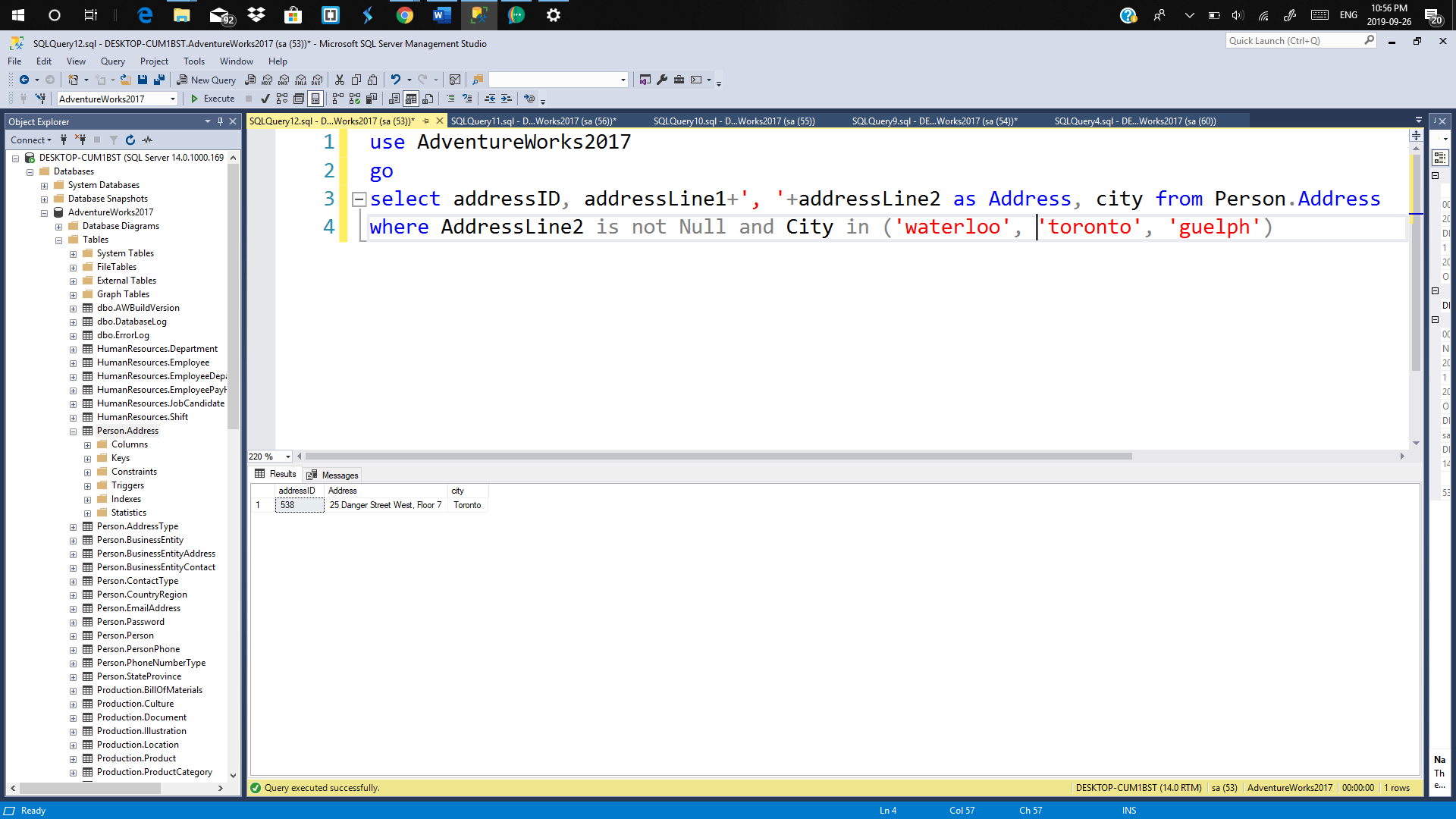
|  |  |
| --- | --- |
| Name | Uploaded (for instructor) |
| Chanchal Pauly Thattil |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 

# IT Standards Marking Sheet

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Programming & SQL Standards - 1% each** | | | |  | |
|  | P1 Meaningful Identifiers | |  | P20 Code Module Size and Focus | |
|  | P2 Prefixes & Hungarian Notation | |  | P21 Single Point of Exit | |
|  | P3 Identifier Case Conventions | |  | P22 Disabled Code & Misleading | |
|  | P4 Header Comments | |  | P23 Each Class in a File Named | |
|  | P5 Method Comments | |  | P24 Class Organization | |
|  | P9 "Magic" Numbers and Strings | |  | P25 Unwise Coding Practice | |
|  | P10 Constant Scope | |  | SQL1 Table Names | |
|  | P11 Indentation | |  | SQL2 Column Names | |
|  | P12 Line Length and Wrapping | |  | SQL3 Keywords & Function Names | |
|  | P13 Blank Lines | |  | SQL4 Header Comments | |
|  | P14 Code Crowding | |  | SQL5 Output Messages | |
|  | P15 Space Around Binary Operators | |  | SQL6 Implementation Comments | |
|  | P16 Space After Delimiters | |  | SQL7 Formatting | |
|  | P17 Curly Brace Alignment | |  | SQL8 Subquery IN and = | |
|  | P19 Global Variables | |  |  | |
|  | |  | | **Late Assignments** | |
| **Days Late** | **Penalty %** |
| **Base Mark:** | |  | | 1 | 5 |
| **Standards Penalties: - %** | | - | | 2 | 10 |
| **Late Penalties: - %** | | - | | 3 | 20 |
| **Final Mark:** | |  | | 4 | 40 |
|  | |  | | 5 | 60 |
| 6 | 80 |
| 7 | 100 |
|  |  |

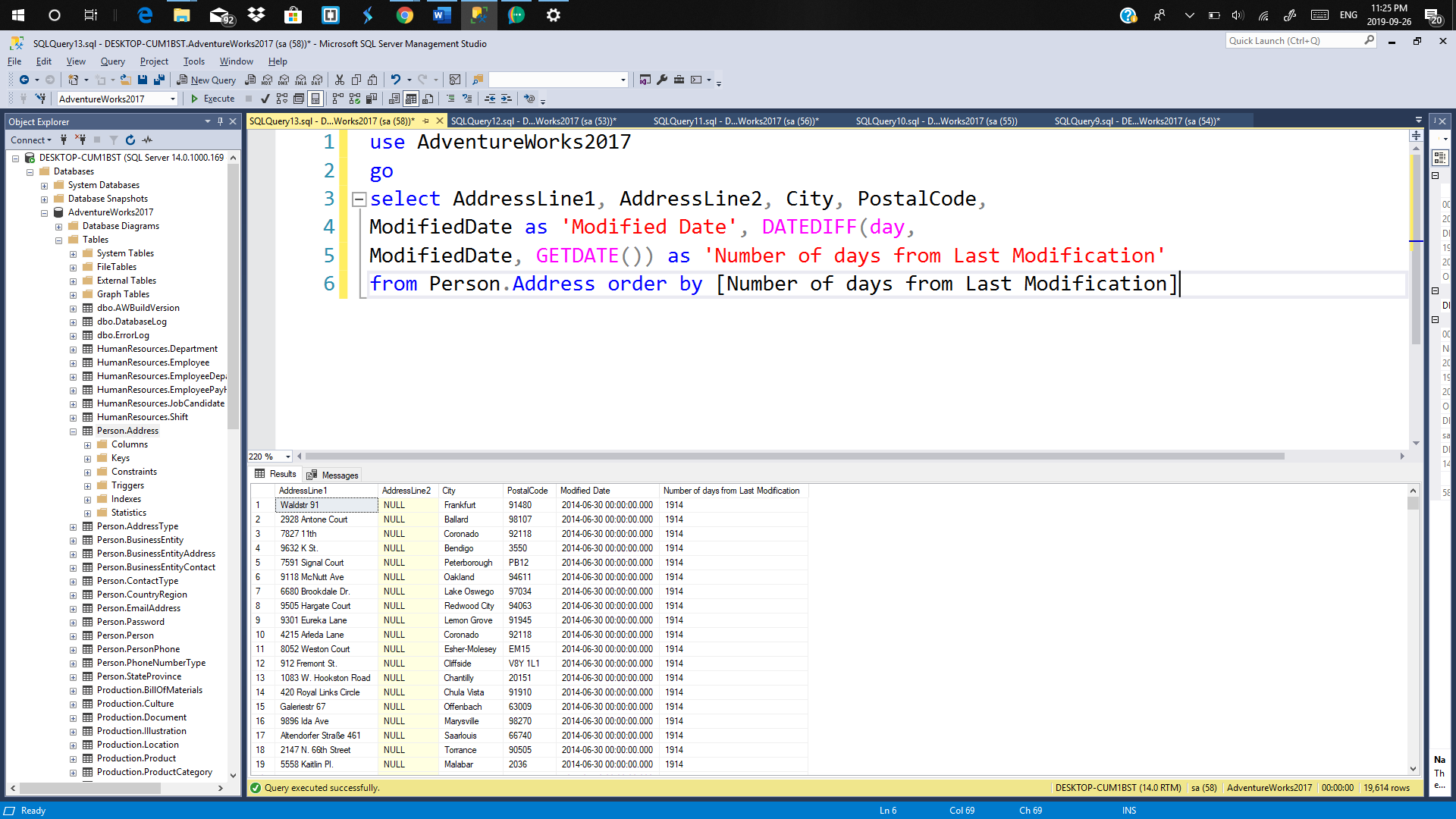
1. Find the number of records of people in the database who have a “Title” associated with their record.
2. Get a list of the unique last names of people in the database who have a first name of either “Ken” or “Kenneth”. Sort the last names in descending order.
3. Retrieve the address ID, a column with the address Line 1 and address line 2 concatenated (and formatted well) and the city for addresses that actually have a second line address and the city is either Waterloo, Toronto or Guelph. Order by city in ascending order and then by the full address in descending order



1. For addresses in the city of Ottawa, get the following information as separate columns:

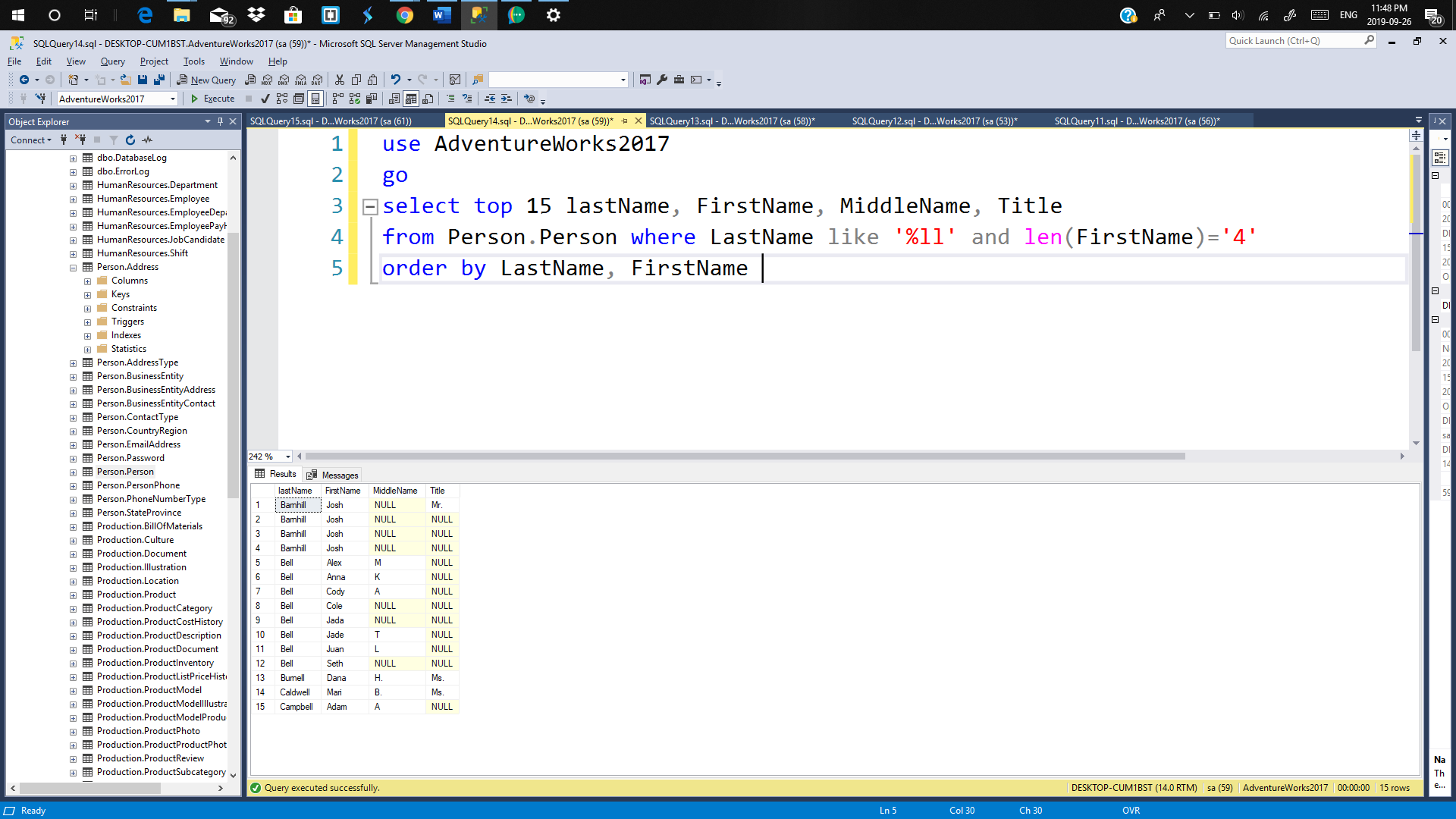
* Address Line 1
* Address Line 2
* City
* Postal Code
* Date the Record was Last Modified
* How many days ago the record was last modified

Sort the data in a way that makes sense and include a comment as to why you chose the sort order. Also be sure that the column headers all have meaningful names.

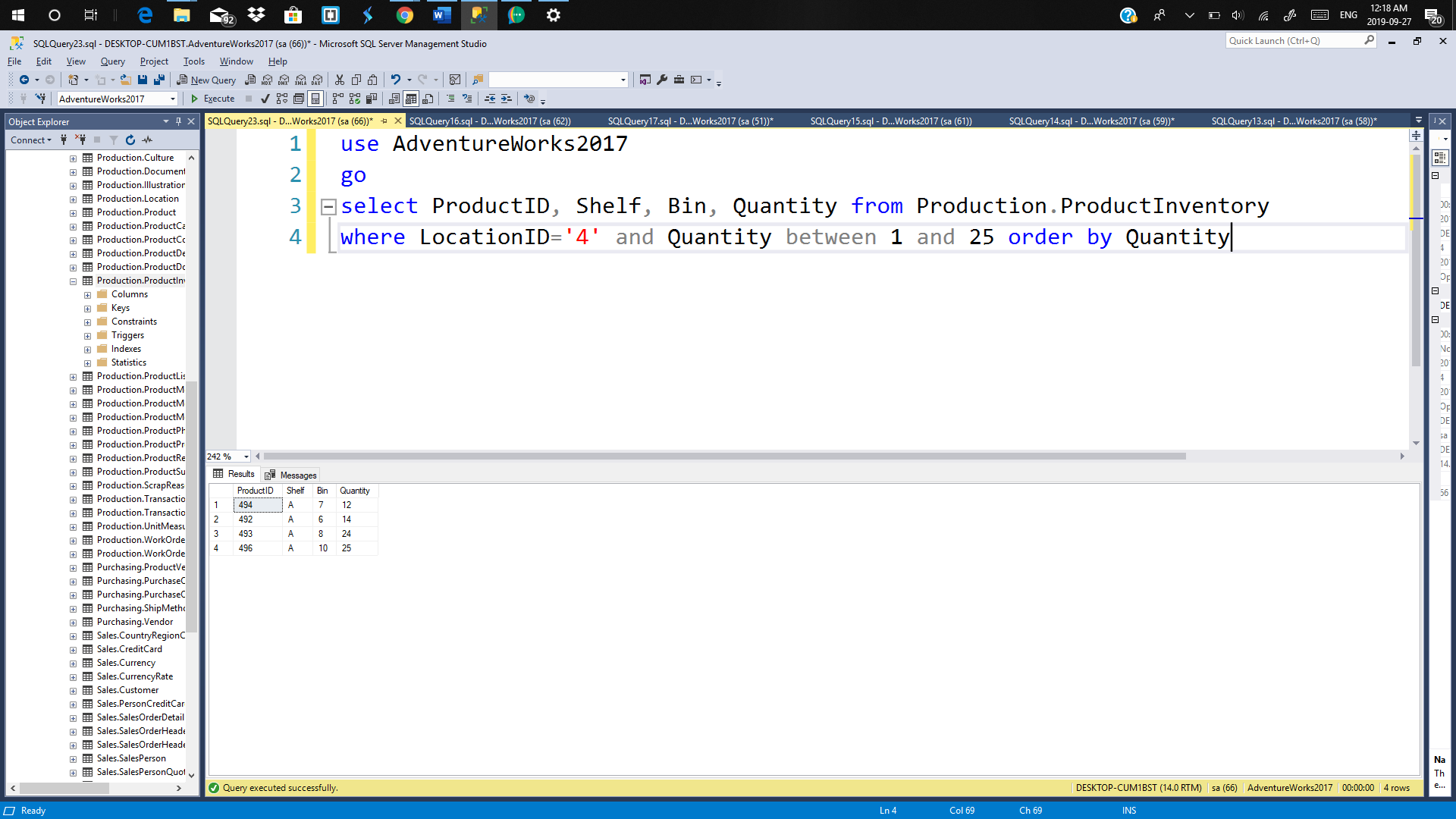


For sorting is done on basis of number of days from last modification, so that last modified rows will be appeared on the top of the table.

1. We’d like to get the Last Names, First Names, Middle Name and Title for people with last names that end with “ll” (i.e. 2 L’s) and a first name that has exactly four characters. Get only the first 15 records when the results are sorted by last name and then first name.



1. We’d like to see product inventory that we have in-stock but is low in supply and location ID of 4. Write a query to retrieve the product ID, shelf, bin and quantity where the quantity is greater than 0 but 25 or less. Sort the results as you see fit and include a comment describing why you sorted how you did.



It is sorted on the basis of quantity, so that items with less in quantity will be appeared on the top of the table result.

1. Expand upon the query in Question #6 (so keep the conditions of being at location ID 4 with a quantity greater than 0 but 25 or less) by joining your information with the Product table so that you can include the following columns in the result:

* Product Name
* Product Number
* Colour
* Days to Manufacture
* Shelf
* Bin
* Quantity

Sort these results in descending order by quantity and then by product #.

