

# INST123 — Databases for All

## Assignment 4-See ELMS for the deadline

### Questions

*This assignment has two questions, each of which is worth 25 points. Be sure to complete both questions.*

**BEFORE YOU START:** Create the *meat\_poultry\_egg\_inspect* and *nyc\_yellow\_taxi\_trips\_2016\_06\_01* tables if you haven't done so yet, using the table creation script and importing the data from the two related csv files linked on the Assignment 4 page. Please contact the INST123 instruction team if you encounter issues with the setup process.

**Q.1) (25 points)** In this question you will turn data in the **activities** column of the *meat\_poultry\_egg\_inspect* table into useful information. You will answer three questions:

- How many companies process poultry?
- How many companies process meat?
- How many companies have export certification?

Part 1: Create three new columns called **poultry\_processing** and **meat\_processing** and **export\_certification**. Each should be data type **boolean**.

Part 2: Using **UPDATE**, set the **poultry\_processing** column value to **TRUE** on any row where the **activities** column contains the text "Poultry Processing". Do the same update on the **meat\_processing** column, and then on the **export\_certification** columns, but this time look for the text "Meat Processing" and "Certification - Export" in the **activities** column, respectively.

Part 3: Use the data from the new, updated columns to count how many companies have had each type of inspection.

Hint: You may need to use wild card characters to get the information from the **activities** column.

Your final result set should replicate the following.

	# Poultry Processing bigint	# Meat Processing bigint	# Export Certificate bigint
1	3787	4832	874

Something else to try (not for points): Count the companies that have all three activities: "Poultry Processing", "Meat Processing", and "Certification - Export". Your result set should replicate the following.

	# All Food Inspections bigint
1	156

## Q.2) (25 points)

Part1: Using the New York City taxi data in the ***nyc\_yellow\_taxi\_trips\_2016\_06\_01*** table, calculate the length of each ride using the pickup and drop-off timestamps. Filter the results to display trips that were at least 1 hour long, but shorter than 2.5 hours. Sort the query results from the highest cost to the lowest.

Part 2: (give this answer as a comment in your SQL file). Do you notice anything about the most expensive or cheapest trips that you might want to ask city officials about? (Hint: Do some of the costs seem excessively high or low?)

Your result set should replicate the following.

	trip_id [PK] bigint	pickup timestamp with time zone	dropoff timestamp with time zone	length_of_ride interval	trip_cost numeric (9,2)
1	205125	2016-06-01 16:12:24-04	2016-06-01 17:57:47-04	01:45:23	497.30
2	244873	2016-06-01 18:25:44-04	2016-06-01 19:53:00-04	01:27:16	354.85
3	329398	2016-06-01 22:02:26-04	2016-06-01 23:26:01-04	01:23:35	312.95
4	199609	2016-06-01 15:51:51-04	2016-06-01 17:44:36-04	01:52:45	302.84
5	332231	2016-06-01 22:09:29-04	2016-06-01 23:35:46-04	01:26:17	300.30
6	57916	2016-06-01 08:11:58-04	2016-06-01 09:33:08-04	01:21:10	299.34
7	6434	2016-06-01 00:43:50-04	2016-06-01 01:48:49-04	01:04:59	278.80
8	331145	2016-06-01 22:06:42-04	2016-06-01 23:13:56-04	01:07:14	253.56
9	57061	2016-06-01 08:09:27-04	2016-06-01 09:16:57-04	01:07:30	226.01
10	249828	2016-06-01 18:38:57-04	2016-06-01 20:23:53-04	01:44:56	215.84
11	160861	2016-06-01 13:48:00-04	2016-06-01 16:09:13-04	02:21:13	213.88
12	5855	2016-06-01 00:39:23-04	2016-06-01 01:48:32-04	01:09:09	209.84
13	45745	2016-06-01 07:36:32-04	2016-06-01 08:40:07-04	01:03:35	205.30
14	208208	2016-06-01 16:24:29-04	2016-06-01 18:17:14-04	01:52:45	203.50
15	56608	2016-06-01 08:08:12-04	2016-06-01 09:15:37-04	01:07:25	191.79
16	205817	2016-06-01 16:15:04-04	2016-06-01 17:33:06-04	01:18:02	188.84
17	185535	2016-06-01 15:05:43-04	2016-06-01 16:30:46-04	01:25:03	178.84
18	236152	2016-06-01 18:02:11-04	2016-06-01 19:49:22-04	01:47:11	175.84
19	315516	2016-06-01 21:27:30-04	2016-06-01 22:28:32-04	01:01:02	175.56
20	179452	2016-06-01 14:47:00-04	2016-06-01 17:01:42-04	02:14:42	174.84
21	299293	2016-06-01 20:47:01-04	2016-06-01 21:47:57-04	01:00:56	174.83

4826 rows returned (This is just a partial display of all result rows.)