**IS 475/675 Database Design and Implementation Project Part 2 : updated on 12/10/2019**

**What is the purpose of this part of the project?** The objectives of this part of the project are to (1) revise your logical model; and (2) validate that model through the implementation of a prototype database for Replica Toys. These parts of the project require you to:

• Evaluate and **revise** as necessary the logical data model submitted for part 1. An important final deliverable for this project is a complete logical data model in third normal form.

• Design a prototype physical data model that will generate the result tables listed in this document.

• Create a prototype physical data model using the SQL Server database management system. You do not have to include each of the attributes on your logical model on your prototype physical data model, but you do have to include all the attributes that are in the sample data provided in the Excel workbook “ReplicaToysData-f19.xlsx”.

• Import the data provided in the Excel workbook entitled “ReplicaToysData-f19.xlsx”. Put the data in the tables in your database so that the data is not redundant. I recommend using the SQL Server Import/Export utility to import the data into a single table. Then use the INSERT INTO/SELECT FROM statement to parse the data from the single table and input it into each of the tables in your database. A SQL Lab showing how to do this was completed in class on 11/26/2019. The lab is available on WebCampus for that date.

• Write SQL queries to generate the requested result tables.

**What are the deliverables?**

1. **A revised logical data model.** This deliverable is an ERD. If your logical data model (documented with an ERD) turned in for part 1 did not have any problems noted by me, then simply turn in that ERD. If you found you needed to modify the design to address my concerns then turn in a revised ERD. You must have a logical data model in third normal form as one of the deliverables for this part of the project.

**2. A copy of your original logical data model submitted for part 1.**

**3. An ERD for the prototype database.** This deliverable will be an ERD of the database implemented for the prototype. Since the prototype should include only those tables necessary to store the data in the Excel workbook (ReplicaToysData-f19.xlsx) and produce the required queries, this ERD will be a sub-set of the revised logical data model. The prototype ERD should look exactly like the tables you create for the project.

**4. A prototype database.** Create the tables necessary to store the data in the Excel workbook ReplicaToysData-f19.xlsx with minimal data redundancy. Make sure you create appropriate data types for all fields in all tables. Create appropriate constraints. Hint: There are 30 different problem reports and 56 different tests in the sample data set provided in the Excel workbook. There are 40 different people. “People” are not separated by the type of person in the workbook and do not have to be stored in separate entities in your database. You can create a “person” table and store all referenced people in the database.

Document this deliverable as follows: CREATE TABLE statements. Turn in all SQL statements used to create tables.

1

SELECT \* FROM each\_tbl. Turn in the complete contents of all tables in the prototype database. Please do NOT turn in any SQL INSERT statements.

**5. SQL Queries.** Turn in the code and the output for the queries on the next page using the same format as used on the SQL homework assignments (HW#7, 8, 9). Please place the code before the output generated in the printed output for each requested query. Provide documentation/comments for all views and CTEs so that I understand the purpose of the views and/or CTEs in your code. Be sure to include the code for all views used for the queries! I will not have time to chase down the views and ask you to submit them!!

**6. Electronic version of the SQL Queries.** Upload a file (only one file, not separate files for each query!) with just the SQL code for your queries and no output. The file can be a Word document or txt file but make sure that they are not “pictures” of your SQL code. Do not turn in a .pdf file for this deliverable. I may want to test your queries with your database, so I need a copy of the SQL code for the queries that is run-able when copied to SQL Server.

7. **The name of the database where your test tables reside.** I have access to all databases on our server, so I don’t need a login but I need to know the name of the database to look at your database and run your queries online.

2

**What are the required SQL queries?**

The SQL queries are designed to test the quality control part of your database – there are no queries required for the marketing, registration, survey questions or returned toys part of your database.

1. The sample data provided in ReplicaToysData-f19.xlsx maintains referential integrity, but there are always other types of data accuracy/integrity issues that are not validated through a referential integrity constraint. For example, are there any tests that were conducted BEFORE a problem report was created? It should be impossible to have a test conducted prior to when a problem report is created, but it is always possible that a person input the dates incorrectly. Check the dates in the problem report and test tables to see which problem reports or tests might have incorrect dates. Here is the full result table:

2. Summarize the ProblemReport by ProblemTypeID. Be sure to include all the ProblemTypes possible in the ProblemType table, even if there are no problem reports for a given problemtypeID. Provide a count by problem type of the injury reports in the ProblemReport table. Here is the result table:

3

3. Which problem reports have a report date in October of the current year? Include the data as shown in the columns below. The DaysInSystem is the difference in date between the ReportDate and the CompleteDate. If the problem report is not complete, then it is the difference between the ReportDate and the current date. When I ran the query below, the current date was December 10, 2019. Your DaysInSystem should be calculated using the GETDATE() when you execute your query, so the DaysInSystem will be a little different than the ones shown for those problem reports that are not yet complete (have a null value for the CompleteDate). Sort the result table by ReportDate. The full result table is provided below:

4. Modify the report created for question #3 to remove the ProblemType description and include each of the tests that were performed for each problem report. The result table should still include only those problem reports with a report date in October of the current year. Include all of the problem reports displayed in question #3 whether or not a test has been completed for that problem report. The full result table is provided below. This report was also run on 12/10/2019.

4

5. Significantly modify the report for question #4 to remove the detailed information about each test and simply provide a count of the tests that were completed for each problem report in the database. This means you should also remove the condition looking for problem reports in October of the current year – we want to count the tests for all problem reports in the system. Remember that the column DaysInSystem is calculated based on the current date (as determined from the function GETDATE()). When I ran this report, GETDATE() = 12/10/2019, so some of your DaysInSystem will have a different value than mine shown below. The result table is shown below.

5

6. Which problem report(s) that are not completed (have a complete date that is null) has/have the largest count of tests? Do not use the SELECT TOP option to determine the answer. Remember that the column DaysInSystem is calculated based on the current date (as determined from the function GETDATE()). When I ran this report, GETDATE() = 12/10/2019, so your DaysInSystem will have a different value than mine shown below.

7. Time for a new query! Summarize problem reports issued for each model of vehicle in the database. Count the total number of problem reports and tests by model. Count the total number of injury reports for each model. Determine the earliest date that a problem was reported and the most recent date that a problem was reported for a given model. Count the tests for each model. Determine the earliest date that a test was done and the most recent date that a test was done for a given model. Include all models in the database, whether or not there is a problem report outstanding for that model. Result table:

8. Which model(s) has/have the most problem reports associated with it? Do not use the SELECT TOP option to determine the answer because with other data sets, it might not be just one model as it is in our dataset.

6

9. List all the tests for the model(s) that have a problem description with the word “battery” in it and also had an injury description. Sort the result table by report ID. The result table is displayed below.

10. Which models have had no problem reports?

7