

Homework 5  
1.264, Fall 2013  
Database implementation for aircraft parts manufacturer  
Due: Monday, October 21

## A. Overview

**This is a two week homework; it is twice as long as usual. You should complete the first half of it during by October 14.**

In this homework you will implement, in SQL Server Express 2012, the database that you designed in homework 4. You may use the homework 4 solutions as the guide for your database design. You may also use your homework 4 response; correct any errors in your data model before you begin building the SQL Server database. The dataset is on the course Web site.

## B. Database implementation overview

The (logical) data model and (physical) database can be the same. If you make some changes between the data model and database implementation, explain them. The SQL Server database that you create must:

- Contain relationships among tables, to allow queries across all the data to be made conveniently.
- Contain referential integrity, to ensure correctness when records are added, deleted or modified.
- Contain appropriate keys, both primary and foreign.
- Be appropriately structured to support the operations required. The entities and attributes must be based on the data model from homework 4.
- Support appropriate cascading delete/update rules for entities with foreign keys.

You must load your database with the data from the dataset provided on the course Web site. You will need to write a set of SQL queries to create and load the tables in your database from the original datasets provided to you. The dataset contains 3 tables, but they are not in normal form, nor do they have relationships among them.

Do not build any data entry forms or menus. We will create these in the following homework. Also, you don't need to create indexes for performance in this homework

A set of suggested steps to develop the database are given below. You are free to approach it in other ways. The steps generally mirror those used to build the data model in homework 3.

To make the downloaded database usable in SQL Server Management Studio:

- Put the downloaded database into Program Files\Microsoft SQL Server\MSSQL11SQLEXPRESS\MSSQL\DATA (or similar).
- Set yourself as the owner for the database that you download. Right click on the database in the explorer, select Properties, select Files, and set the owner (using the browse button on the right).

- See part G below for more details on how to attach the database.

## C. Database implementation

Notes before you start:

- Constructing a database in SQL Server is different than how Visual Paradigm builds the data model. While Visual Paradigm automatically inserted the foreign keys in child tables, SQL Server and other databases don't do that. You must build tables with all attributes, including foreign key attributes, and then set the relationships.
- You must REFRESH after adding each table or relationship in the SQL Server client. Otherwise the new table or relationship may not appear. You must SAVE each new table after each modification before continuing.
- SQL Server allows NULLs in columns by default. It's ok to accept the default in this homework, to save some time, but it's a bad idea to do this in general.

### 1. MIDO

- a. Create the MIDO table by doing a `SELECT DISTINCT column1, column2, ...` into MIDO from PMAList. Use the client (MSE) for all the steps in this homework; type the SQL into the query window.
- b. Set the primary key. In the object explorer (left part of client), expand Tables, then right click on the MIDO table and select 'Design'. In the window, select the attribute(s) to be the primary key, and right click and select 'Set Primary Key'. You can select more than one attribute; use the shift or control key. Save the table after setting the primary key.
- c. If SQL Server doesn't let you set the primary key, there are duplicates in your primary key column. Usually this means that you've forgotten the 'DISTINCT' keyword. We give you a SQL query to look for duplicates at the end of this homework to help you discover any errors.
- d. (If you prefer, you may write a `CREATE TABLE` statement that defines the primary and foreign keys, and then write an `INSERT` statement to load the data. This avoids having to use the client for this.)

### 2. PMAHolder.

- a. Create the PMAHolder table by doing the appropriate `SELECT DISTINCT` from PMAList.
- b. Set the primary key for this table, and for each table as you build it. For composite keys, highlight all the attributes and then hit the key icon.

### 3. Manufacturer

- a. Create the Manufacturer table by doing the appropriate `SELECT DISTINCT` from PMAList.
- b. Set the primary key for this table.

- c. Insert the manufacturer type. McDonnell Douglas is a jet manufacturer; the others are balloon manufacturers.
    - i. In the client, right click on Manufacturer table, select Design, and add the Type attribute to the table (be sure to select an appropriate 'Data Type', nvarchar(50) should work).
    - ii. In the client, right click on Manufacturer table, select 'Edit Top 200 Rows' (this is OK because table has only a few rows), and type the data into the Type column.
4. OriginalPart
- a. Create the OriginalPart table by doing the appropriate SELECT DISTINCT from PMAList. Set the primary key for this table. To create a composite key, highlight all of its columns and then hit the key icon.
5. ReplacementPart
- a. Create the ReplacementPart table by doing the appropriate SELECT DISTINCT from PMAList.
  - b. Set the primary key for this table. To create a composite key, highlight all of its columns and then hit the key icon.
6. ApprovalMeans. Build this in a similar fashion.
7. Create relationships among tables defined so far.
- a. Create relationships as you create each table, to make sure that you've created the keys and attributes correctly. Create the relationship between MIDO and ReplPart, for example. Using the client, right-click ReplPart table and select "Design" Then right-click the MIDO attribute and select 'Relationships'. Select "Add" button then click on the "... " after Tables and Column Specification; using the dropdown boxes, select the MIDO column as the primary key in the MIDO table to define the foreign key MIDO in ReplPart. Hit 'Close' when done. Remember to save the table after making this modification.
  - b. See the final step in the homework for how to resolve problems that may occur. Also, create Diagrams in the client (right click on your database name, choose 'Database Diagrams', 'New Database Diagram', add the tables you want in the diagram, and the diagram will show you all the relationships). Do this frequently as you build the database. The key symbol is the '1' end of the 1-to-many relationship. (If you receive an error while trying to create a diagram, do the following: Right-click the AircraftsParts Database > Select "Properties" > Select the "Files" page on the left > for Owner: type your current SQL Server login ID Select "OK". Then try again to create Diagram)
  - c. Create relationships for ReplPart and ApprovalMeans, PMAHolder and ReplPart, Mfr and OrigPart, OrigPart and ReplPart here also. These are all the relationships you have among the tables you've built so far. The relationship between OrigPart and ReplPart involves a composite foreign key; select both columns in both tables in the relationships wizard. The columns must be in the

same order in both tables; drag and drop them up or down in 'Modify' as needed.

- d. See the final step in the homework for how to resolve problems that may occur. Also, create Diagrams in the client (right click on your database name, choose 'Database Diagrams', 'New Database Diagram', add the tables you want in the diagram, and the diagram will show you all the relationships). Do this frequently as you build the database to make sure you're building your database to match the data model. The key symbol is the '1' end of the 1-to-many relationship.
8. Create the AircraftModel table.
    - a. This may take a little effort. You may SELECT DISTINCT from the Model1, Mfr attributes of the PMAList table into a temporary AircraftTemp table WHERE Model1 is not null.
    - b. You will then need to INSERT INTO AircraftTemp SELECT DISTINCT... from Model2 through Model5, Mfr into the AircraftTemp table. This is 4 separate queries in the simplest approach; you may instead use a SQL UNION if you wish.
    - c. Last, you will need to SELECT DISTINCT from the AircraftTemp table to get your final AircraftModel table. This has a composite primary key.
    - d. Create the relationship from Mfr to AircraftModel.
  9. Create the ReplPartAircraft table by using SELECT DISTINCT from PMAList.
    - a. You will need to do this for Model1 through Model5, using a similar approach to the AircraftModel table above. You will need a WHERE clause to ensure that the model attribute is not null. The composite primary key in this table has 4 attributes. Last, create the relationships from this table to others. You are building the associative table in the many-to-many relationship between aircraft models and replacement parts.
  10. Create the Mode table. Expand the AircraftParts dataset by clicking the '+' > Right-click the 'Tables' folder and click 'New Table...'. In the properties window (at left) modify the table name from the Default to an appropriate name. There are 4 modes (Highway, Rail, Air, Water); you can enter them by hand. These 4 entries must match the names you use in step 12.
  11. Put a primary key on the Carrier table. No other changes are needed.
  12. Create the CarrierMode table. This is the associative table between Carrier and Mode that lists all the modes a carrier provides. This may take a little effort.
    - a. First, create 4 attributes in addition to the single Mode attribute in the Carrier table; call them HighwayMode, RailMode, WaterMode and AirMode.
    - b. Use wildcard matching to convert, for example, '1\_\_' to a 'Highway' in HighwayMode. And '\_1\_\_' to a 'Rail' in RailMode. Etc. Use the LIKE keyword with the wildcard.

- c. Now, you have 4 mode columns in the Carrier table, with the mode values in them for the carriers indicating the modes they offer.
- d. You can now create the CarrierMode table. This may also take a little effort. You may SELECT DISTINCT from the HighwayMode attribute of the Carrier table into the CarrierMode table. You will then need to INSERT INTO CarrierMode SELECT DISTINCT... for RailMode, WaterMode and AirMode into the CarrierMode table. (This is similar to what you did with AircraftModel, for example.) This table has a composite primary key. You should name the column Mode, not HighwayMode, in this table.
- e. Delete the original Mode column if you wish, or keep it to check correctness while you do the homework and delete it at the end. Be sure to specify the composite primary key to the CarrierMode table.

13. Create the Customer table.

- a. You may use the Airline table given to you, or create a table with your own customers.
- b. If you wish to create customer IDs automatically, use:

```
CREATE TABLE Customers (
  CustomerID int identity (1, 1) not null primary key,
  OtherAttribute nvarchar(50)
  ... )
```

14. Create the OrderHeader table.

- a. It will initially contain no data.
- b. Set its primary key. Use the syntax above to create order numbers automatically.
- c. Don't call this table Order, since 'Order' is a SQL reserved word, as in 'ORDER BY'.
- d. Create foreign keys (by creating relationships) for the carrier certificate number (carrier primary key) and mode in the OrderHeader table.
- e. You must make the data types and lengths of foreign keys the same as they are in the parent table (or you won't be able to create the relationships).

15. Create the OrderDetail table.

- a. It will initially contain no data.
- b. Set its composite primary key.
- c. Set a foreign key for replacement part and PMA holder.
- d. Again, make the data types and lengths of foreign keys the same as they are in the parent table.

16. After you've built all the tables, you need to verify the relationships between them. You should build the relationships as you go, and they should all be present now.

- a. Follow the data model from homework 4. Build each relationship shown in that model.

- b. If a relationship fails to build, check the following items:
  - i. Does each table have the correct primary keys?
  - ii. Are the data types and lengths the same?
  - iii. Have you selected the correct foreign key for the relationship?
  - iv. If you cannot set the primary key for a table, use the Find duplicates SQL code below to see if there are duplicates. There shouldn't be any (I've fixed the original data, which did have problems), so this means that you've either built the table incorrectly (perhaps forgetting the DISTINCT keyword) or that you've defined the primary key incorrectly (perhaps not setting it as a composite key). Primary keys cannot have NULLs.
  - v. If you cannot set a relationship:
    - 1. Check that the data types are exactly the same in both tables.
    - 2. If SQL Server says it cannot build the relationship, use the Find unmatched SQL code below to find the records that have no entry in the parent table. This should usually be a good hint on what is wrong.
    - 3. If there is a composite key, have the keys in the same order in both tables.

This completes your database. Please leave the original 'raw' tables and any temporary tables you've created in the database so that we can check your queries. Normally you would delete the 'raw' and temporary tables at this point, of course, and just use the new database you've created. We will delete the 'raw' and temporary tables before using the database in later homeworks.

#### **D. Queries**

Now that you've built the database, implement 5 queries to return data from it. You cannot use the original datasets; use only the tables and relationships you have built.

1. Find the number of replacement parts (PartNumber) that the distributor carries for each manufacturer. Display the manufacturer name and the number of replacement parts. Display in descending order of number of parts.
2. For each MIDO, list the aircraft models for which that MIDO inspects replacement parts. Do this only for McDonnell-Douglas models, not balloons.
3. List the states, if any, that do not have all 4 modes available. Include Puerto Rico, Canadian provinces, etc. Hint: You may use more than one query to implement this: create a view with the first query and use it to implement the second query.
4. List the original aircraft parts that have more than one replacement part, if any. Order by descending number of replacement parts.
5. Find the average price of replacement parts, by PMAHolder. Order by average price in descending order.

#### **E. Assignment**

1. Turn in the queries that you wrote to create the database, saved in the client as a .sql or .txt file. When you save your queries, please number them from 1 to 15, to match the steps in part C of this homework. If you need more than one query for a step, number them 1a, 1b, 1c, etc. Thus, when you write a SQL statement, please comment it as '--9 ReplPartAircraft table', for example. You don't need to write queries for tables you created from scratch, such as OrderHeader, etc. If you use your own sequence of queries, please number them sequentially even if they don't match the homework sequence.
2. Turn in the queries from part D in the assignment. Please number them D1 through D5 and place them in a .txt or .sql file. They can be in the same file as your earlier queries.
3. Turn in the actual database files (the .mdf and .ldf files that SQL Server creates).
4. Please put all these files in a single zip file that you upload to the course Web site. See the TA if you need help creating the zip file.

## F. SQL Code Examples

1. Find duplicates. To find duplicate certificate numbers in the carrier table:

```
SELECT Carrier.CERTNO, Carrier.COMPANY
FROM Carrier
WHERE (Carrier.CERTNO In (SELECT CERTNO FROM Carrier As Tmp GROUP
BY CERTNO HAVING Count(*)>1 ))
ORDER BY Carrier.CERTNO;
```

Modify this for any table in which you wish to check the intended primary key for duplicates.

2. Find unmatched rows between a parent and child table. For example, find rows in order detail for which there is no row in the order table (e.g., we have order detail for order 733 but there is no row 733 in the order header table):

```
SELECT ManufOrderDetail.OrderNbr
FROM ManufOrderDetail LEFT JOIN ManufOrder ON ManufOrderDetail.OrderNbr
= ManufOrder.OrderNbr
WHERE (ManufOrder.OrderNbr Is Null);
```

Modify this to find rows in any child table for which there is no corresponding row in the parent table.

## G. Attaching the homework 5 data

1. Open Microsoft SQL Server Management Studio
2. Log in, preferably using SQL Server authentication and the login that you created in class. If you didn't create a new login, you can login as "sa" with the suggested password of "Kocur10!"
3. Under the "Object Explorer" pane on the left of the screen, right click on the "Databases" folder: click "Attach" from the popup menu.
4. Click the "Add" button beneath the "Databases to attach" pane.

5. Navigate to the .mdf file that you downloaded and click "OK."
  - a. If you can't see the .mdf file, it could be because it is downloaded to a folder that SQL Server does not have permission to access. Move the file somewhere else, such as the folder in which you store your files for this class. After moving the file to a place that can be accessed, try step 5 again.
6. Even though you just told the program where to find the .mdf file, you still need to specify this again in the "AircraftParts database details" pane below (yes, Microsoft should probably fix this).
  - a. Click the "..." button next in the first row (the row whose File Type is "Data") under the "AircraftParts database details" pane and navigate to the same file that you specified in step 5.1.
  - b. Single click the second row of this table (the row whose File Type is "Log") and click the "Remove" button. This will prevent the program from trying to load the non-existent log file and will cause it to create a new one.
7. Click "OK" on the bottom of the popup.
  1. If you get an error, it is probably because you don't have permission to edit the folder to which you moved the .mdf file. If you get the error, continue below; if you don't get the error, continue to step 8.
    - (a) Use Windows Explorer (not Internet Explorer) to navigate to the folder to which you moved the .mdf file.
    - (b) Right click on the folder and click "Properties" from the popup menu.
    - (c) Each version of Windows is slightly different, but you should be able to select a "Security" tab from the popup window.
    - (d) Click the "Edit" button beneath the users or groups pane.
    - (e) On the window that pops up, single click "Users," or "All users" or something similar (again, it may vary with your operating system and installation).
    - (f) Check the "full privileges" checkbox, then click "OK." This should grant the privileges necessary for SQL Server to access the file.
    - (g) Return to SQL Server Management Studio and click the "OK" button again. If this doesn't work, you might have to re-select the file by repeating step 6.a above.
8. The "AircraftParts" database should now be visible under the "Databases" folder in the "Object Explorer" tree. If so, you're good to go!

MIT OpenCourseWare  
<http://ocw.mit.edu>

1.264J / ESD.264J Database, Internet, and Systems Integration Technologies  
Fall 2013

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.