

Data Warehouse Schemas

This assignment provides experience with transaction database and data warehouse database. We are not going to create a data warehouse. Instead, we reinforce our learning by exploring the SQL server sample database AdventureWorks.

Deliveries:

First create a folder and save it with the following format:

dw_hw1_<firstname>_<lastname>_<studentid>.

The folder contains:

- Deliveries for each task.
- A README.txt that explain the contents.

Then Zip this folder and submit it via Canvas

Task 1: Create the Sample Databases

Create both **AdventureWorks** and **AdventureWorksDW** according to the instruction on this page <https://github.com/microsoft/sql-server-samples/tree/master/samples/databases/adventure-works>.

The easiest way is to Install from a backup file onto your database.

After installation, there should be two databases **AdventureWorks** and **AdventureWorksDW** (maybe suffixed with year)

Using queries and diagram functions to become familiar with the tables and contents in the two databases. Note that removing a table from a diagram does not change the database, however explicitly removing relationships change the database.

Building the DW schema including, but not limited to, star/snowflake for each fact table in the **AdventureWorksDW** database. You can create the schemas within SQL Server database and then use the "Copy Diagram to Clipboard" context menu to copy and paste it in a drawing program on your laptop, finally save each one (100% zoom) as a clear image ("Print screen" may not work well when the schema size is large)

Deliveries:

Images for all the DW schemas including, but not limited to, star/snowflake.

Tips:

1. Use the table name to identify whether a table is a fact or a dimension table. Not all tables are useful.
2. Use foreign keys to identify the relationships among these tables. If you think two tables should be connected via a foreign key while this connection does not exist in the database, just add it in your schema.
3. A fact table may act as a dimensional table of another fact table.
4. You may dump out the create-table scripts to help you identify relationships.
5. PK: primary key, FK: foreign key, AK: alternative key (may safe ignore), IDENTITY: identity column.

Task 2: (Reverse Engineering) make a bus matrix

In general, a bus matrix is created after a business process is chosen. Here we do reverse engineering to practice making a bus matrix. One may assume each fact table is a business process.

Use the bus_matrix_template.xlsx template file to finish the task.

Deliveries:

An Excel file with bus matrix inside.

Tasks 3: Answer Questions in the *Questions.docx*

For single- or multiple- choice questions, simply highlight (bold or color) the answer. For example:

Question: what is this course about?

- Statistics
- C++ programming
- **Data Warehousing**

Deliveries:

The word file with answers.