

# CSE4110 – Database System

*Project2. Normalization and Query Processing*



Spring 2021

# Automobile company

*“You are a DBA in this company”*

**Goal :** The goal of this project is to provide a realistic experience in the physical design, query processing implementation and maintenance of a small relational database you made in Project 1.

## **Application description(Same as Project 1) :**

The application is an automobile company, such as General Motors, Ford, Toyota, or Volkswagen (or maybe a company from yesteryear like Studebaker, Hudson, Nash, or Packard). In our hypothetical company, it has been decided to redesign a major part of the database that underlies company operations. Unfortunately, the manager assigned to solicit database design proposals is not very computer literate and is unable to provide a very detailed specification at the technical level. Fortunately, you are able to do that.

The company needs to keep quite a bit of data, but we shall focus on the following aspects of corporate operations.

## **Here are a few points to consider :**

- vehicles: Each vehicle as a vehicle identification number (VIN). Lots of stuff is encoded in real VINs (they are well described on Wikipedia), but you can just make them up if you want.
- brands: Each company may have several brands (for example, GM has Chevrolet, Pontiac, Buick, Cadillac, GMC, Saturn, Hummer, Saab, Daewoo, Holden, Vauxhall, and Opel and Volkswagen has Volkswagen, Audi, Lamborghini, Bentley, Bugatti,

Skoda, and SEAT)

- models: Each brand offers several models (for example, Buick's models are the Enclave, LaCrosse, and Lucerne, and Mercury's models are the Mariner, Milan, Sable, and Grand Marquis). Each model may come in a variety of body styles (4-door, wagon, etc.)
- options: We'll stick to color, and maybe engine and transmission.
- suppliers: suppliers supply certain parts for certain models
- company-owned manufacturing plants: Some plants supply certain parts for certain models; others do final assembly of actual cars.
- customers: In reality, lots of demographic data are gathered. We'll stick to name, address, phone, gender, and annual income for individual buyers. The customer may also be a company (e.g. Hertz, Avis, or other companies that maintain corporate fleets, but we'll skip that).
- We'll skip data on corporate finance, pending bailouts, bankruptcy status etc. Not that these data are unimportant, but we need to keep the project within bounds.

## Project Requirements :

### 1. BCNF Decomposition

- **Decompose your Relation Schema into BCNF form if they are not.**
- Same process learned in lecture

### 2. Physical Schema Diagram

- After creating an decomposed logical schema diagram, also creating Physical Schema diagram
- Create the "physical" schema diagram in Erwin Data Modeler
- Be sure to identify data types, domain, constraints, relationship type, allowing nulls.

### 3. Queries

The queries listed below are those that your client wants turned in. We now called these query category as TYPE. (total 13 difference query types)

- (TYPE 1) Show the sales trends for a particular brand over the past k years. (1)
    - (TYPE 1-1) Then break these data out by gender of the buyer. (2)
    - (TYPE 1-1-1) Then by income range. (3)
  - (TYPE 2) Show sales trends for various brands over the past k months. (4)
    - (TYPE 2-1) Then break these data out by gender of the buyer. (5)
    - (TYPE 2-1-1) Then by income range. (6)
  - (TYPE 3) Find that transmissions made by supplier (company name) between two given dates are defective. (7)
    - (TYPE 3-1) Find the VIN of each car containing such a transmission and the customer to which it was sold. (8)
    - (TYPE 3-2) Find the dealer who sold the VIN and transmission for each vehicle containing these transmissions. (9)
  - (TYPE 4) Find the top k brands by dollar-amount sold by the year. (10)
  - (TYPE 5) Find the top k brands by unit sales by the year. (11)
  - (TYPE 6) In what month(s) do convertibles sell best? (12)
  - (TYPE 7) Find those dealers who keep a vehicle in inventory for the longest average time. (13)
- We should execute these queries within MySQL DBMS what you managed. The customers can check all results from these query execution.
  - We use ODBC with C language API to implement your Database model

### 4. Code implementation

- We will use Visual Studio 2019 and MySQL connector(ODBC)

- Implement C code to execute your queries in MySQL DBMS you built
- The program must use file input from text file which have CRUD queries as text format

e.g. CREATE, INSERT, UPDATE, DELETE

- The program has a user interface to handle each TYPE. Program runs until put quit instruction.

```

----- SELECT QUERY TYPES -----
1. TYPE 1
2. TYPE 2
3. TYPE 3
4. TYPE 4
5. TYPE 5
6. TYPE 6
7. TYPE 7
0. QUIT

```

- After select one of type in menu, user put values of query as stdin and print result of query execution as stdout.

```

----- TYPE 5 -----
** Find the top k brands by unit sales by the year**
Which K? : 3

```

- TYPE 1 & 2 & 3 has submenu for subtypes.

```

----- Subtypes in TYPE 3 -----
1. TYPE 3-1
2. TYPE 3-2

```

- Keep running each type of query until put "0" in stdin. (for all types of query)
- After exit from query menu, back to select menu.

**What to turn in :**

- Decomposed Logical Schema diagram (.png)
  - **student\_id.png** (submitted filename ) e.g. 20219999.png
- “Physical” Scheme diagram ERwin file (.erwin)
  - **student\_id.erwin** (submitted filename ) e.g. 20219999.erwin
- **C code (.c or .cpp)**
  - **student\_id.c** (submitted filename) e.g. 20219999.c
  - Describe your code with comment (if neccessary)
  - It should be able to build and execute in Visual Studio 2019 with MySQL C APL
- Report file (.pdf)
  - **[project2]student\_id.pdf** (submitted filename) e.g. [project2]20219999.pdf
  - Describe the detail explanation about your Physical Schema diagram and ODBC implementation within MySQL that you made.
  - MAKE YOUR OWN DESCRIPTION on physical schema and ODBC C language codes.
  - Describe the detail explanation how you decompose each relationship to BCNF, see testing one of your relationship result by simplified test in lecture notes.
  - Feel free to use any template you made.

#### NOTICE :

- 2020.06.14(Mon) 18:00
- Submit your soft copy with title “**[DBproject2]student\_id**” to [sogang879@gmail.com](mailto:sogang879@gmail.com) (softcopy includes png, erwin, c code and pdf file you wrote)
- Submit your hardcopy to box in front of AS816 before the deadline. (hardcopy includes one E-R model picture and one report you wrote)
- **DON'T COPY ANYTHING FROM YOUR FRIENDS AND WEB SOURCES. IF YOU VIOLATE THIS, YOU WILL GET F FOR THIS COURSE.**