**BUS 440 Database Management**

**Use Case:**

This assignment will instruct you to design and create a database for a small business, *Ranier Brothers Solutions*. The business performs small home-building and remodeling services. The company does electrical, interior, and exterior work for clients. The company is currently involved with three projects: Highland House, Baker Remodel, and Hew Remodel. They purchase various products and pieces of equipment, such as wire, siding, doors, shingles, electrical panels, etc. from a supplier (often at a discount). Some of their trusted suppliers are NW Electric, EB Supplies, Contractor, Inc., and Interior, Inc.

To help manage the projects, the company owner tracks each project, the homeowner’s name and phone number, the project category on a project equipment list on an Excel spreadsheet. He has hired you to design and build a small database so he can better track his projects, the equipment used on them, and the associated costs from supplier quotes.

**Instructions:**

1. Using the data in Figure 1, design a database. To do so, look at the data and the use case. What are some business rules?
   1. A project purchases many items
   2. One supplier supplies many items
   3. A project has one project owner
   4. A owner has one phone number
   5. One to many items belong to each category.
   6. A discount can apply to many items
   7. A unit price can apply to one or more items
2. Write out the relational schema for all of the data and the functional dependencies.

PROJECT(**ProjectName**, OwnerContact, Phone, Category, Quantity, **ItemDescription**, UnitPrice, ExtendedPrice, **Supplier**, Discount)

**ProjectName** -> (OwnerContact)

**OwnerContact** -> (Phone)

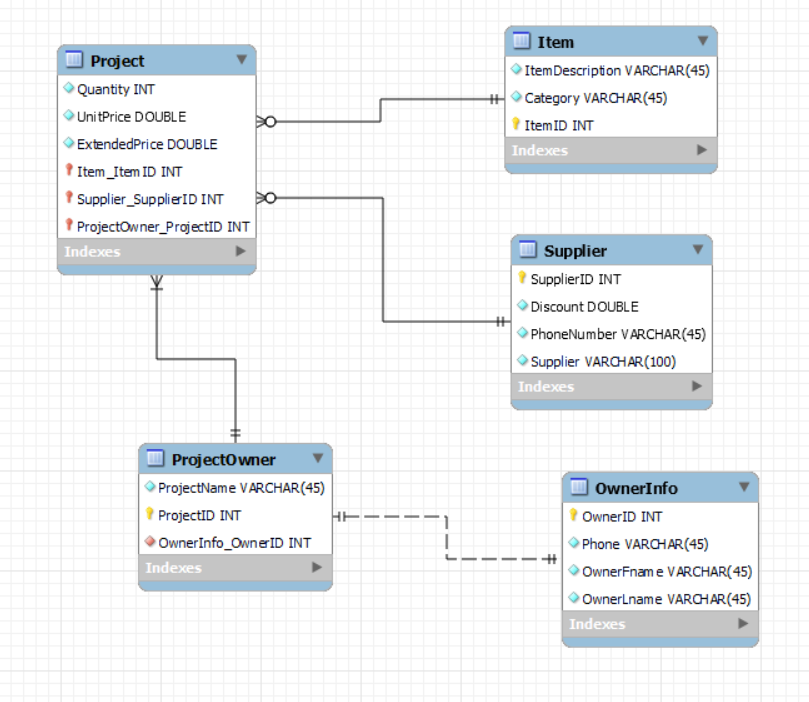
**ItemDescription** -> (Category)

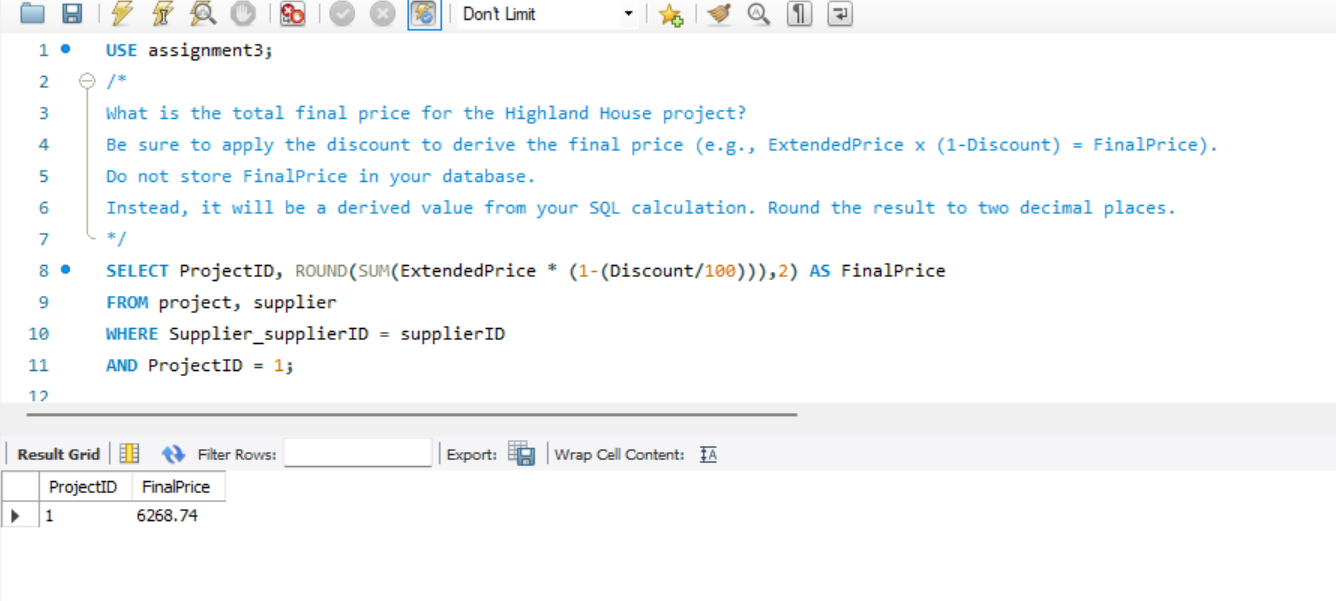
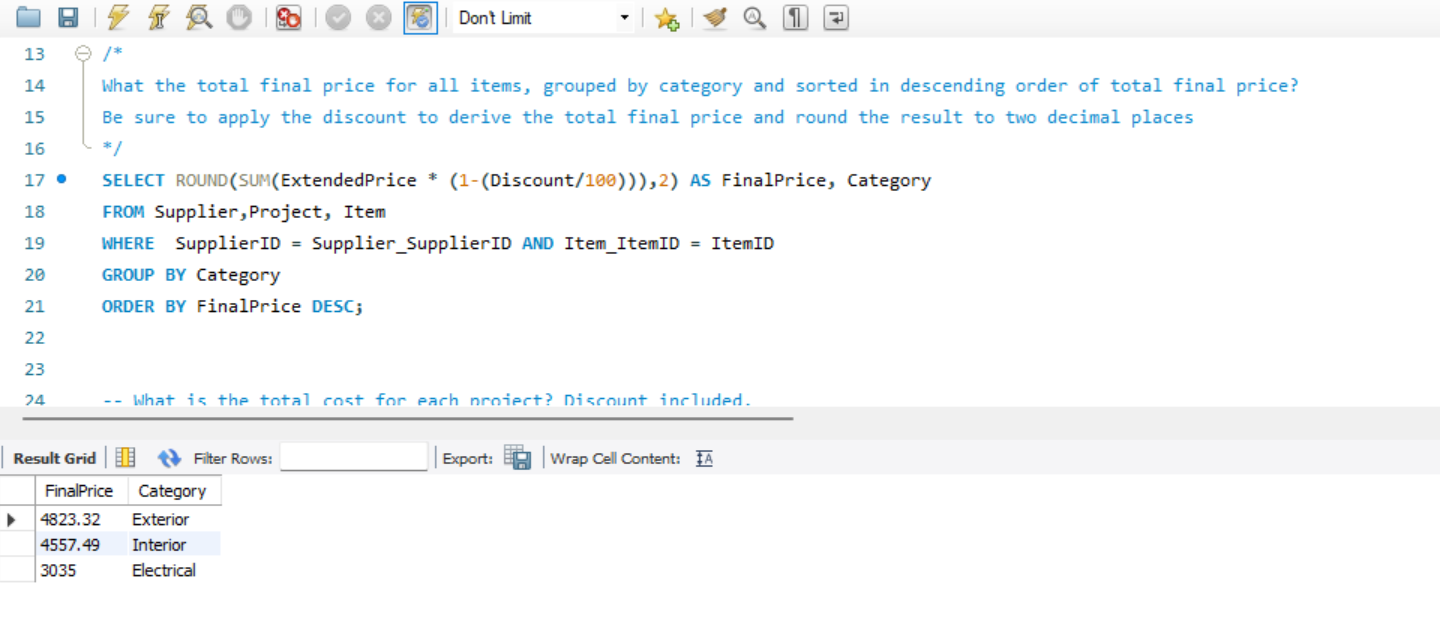
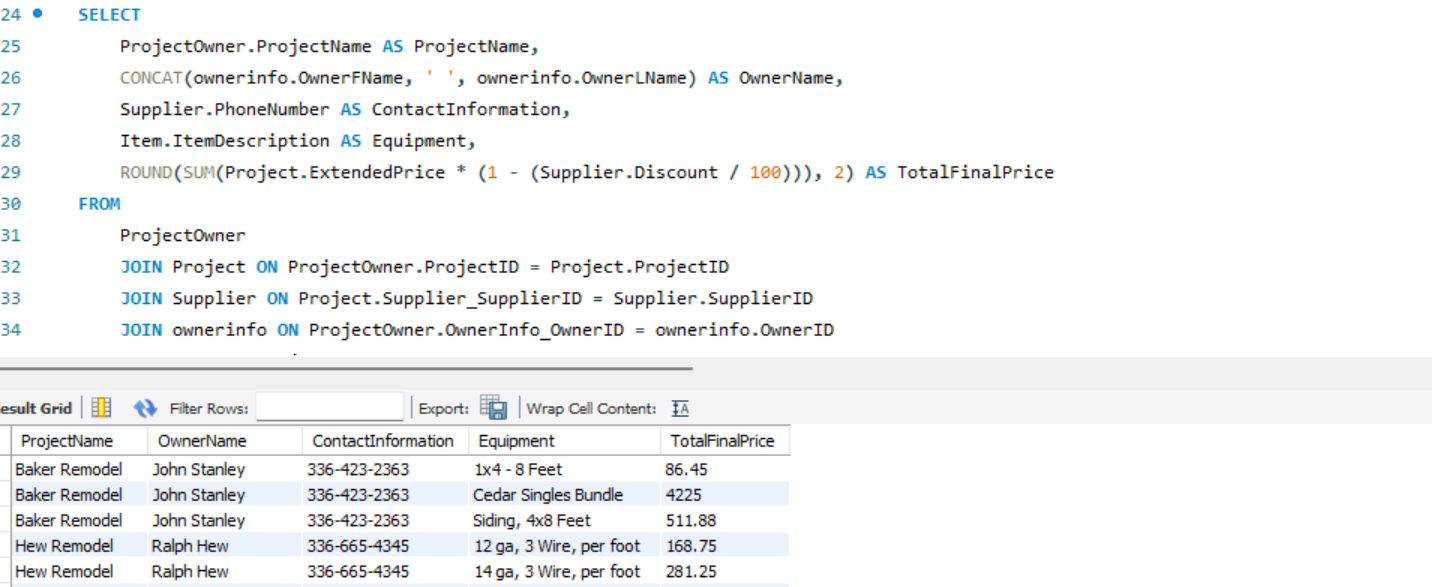
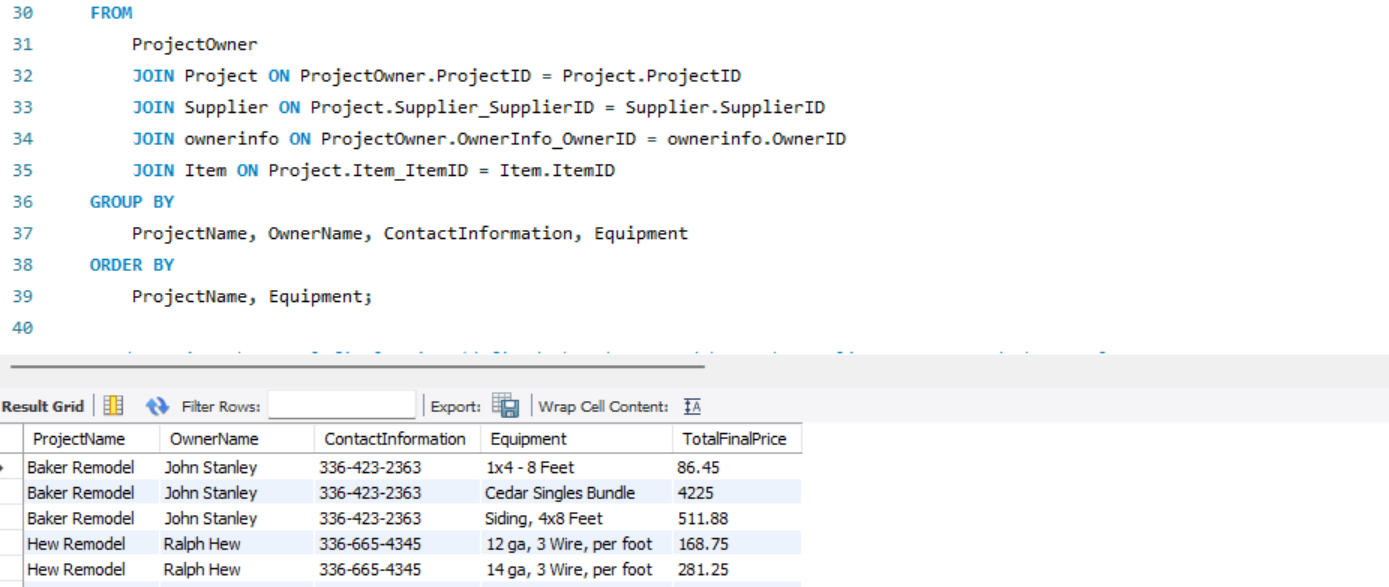
**Supplier** -> (Discount)

Quantity, UnitPrice -> (ExtendedPrice)

**ProjectName**, **Supplier**, **ItemDescription** -> (Quantity, UnitPrice)

1. Normalize to 3NF. Show your work for each step.
   1. Normalize to 1NF. Be sure to use proper primary keys.
      1. PROJECT(**ProjectName**, **Supplier** , **ItemDescription**, Quantity, UnitPrice, OwnerContact, Phone, Category, ExtendedPrice, Discount)
   2. Normalize to 2NF.
      1. PROJECT(***ProjectName***, ***Supplier***, ***ItemDescription***, Quantity, UnitPrice, Phone, ExtendedPrice)
      2. PROJOWNER(**ProjectName**, OwnerContact)
      3. SUPPLIER(**Supplier**, Discount)
      4. ITEM(**ItemDescription**, Category)
   3. Normalize to 3NF. Assign foreign keys, as appropriate.
      1. PROJECT(***ProjectName***, ***Supplier*** , ***ItemDescription***, Quantity, UnitPrice, ExtendedPrice)
      2. PROJOWNER(**ProjectName**, *OwnerContact*)
      3. SUPPLIER(**Supplier**, Discount)
      4. ITEM(**ItemDescription**, Category)
      5. OWNERINFO(**OwnerContact**, Phone)
2. Names are usually not appropriate, so add a code as a surrogate PK for each table as an improvement. Add at least one other improvement.
   * 1. PROJECT(***ProjectID***, *SupplierID*, *ItemID*, Quantity, UnitPrice, ExtendedPrice)
     2. PROJOWNER(**ProjectID**, ProjectName, *OwnerID*)
     3. SUPPLIER(**SupplierID**, Supplier, SupplierPhone, Discount)
     4. ITEM(**ItemID**, ItemDescription, Category)
     5. OWNERINFO(**OwnerID**, OwnerFname, OwnerLname, Phone)
3. Create an EER in MySQL Workbench to show your final design (model) for the database, including all entities, relationships, and columns. Save the EER as a MySQL Workbench model (.mwb) file.



1. Build the database and each database table in MySQL from your design. To do so, synchronize your model to forward-engineer your design to your build via MySQL Workbench.
2. Populate the tables with the data from Figure 1. You may import the data from another source, key the data into the MySQL database tables, or use SQL statements to insert the data into tables.
3. Create and run the following SQL programs. Save them in a script named *RanierBrotherSolutions.sql*. Be sure to validate your results using an alternate method. Screenshot your code and answer in this document for each question. Use comments to clarify each question. Add each team member name at
   1. What is the total final price for the Highland House project? Be sure to apply the discount to derive the final price (e.g., ExtendedPrice x (1-Discount) = FinalPrice). Do not store FinalPrice in your database. Instead, it will be a derived value from your SQL calculation. Round the result to two decimal places. – Alan Tung
      1. 
   2. What is the total final price for all items, grouped by category and sorted in descending order of total final price? Be sure to apply the discount to derive the total final price and round the result to two decimal places. -- Alan Tung
      1. 
   3. List all of the company projects (project name), the owner name and contact information for each project, the equipment (item description) needed for that project, and the total final price (which includes the discount, defined above). Round the result to two decimal places and sort first by project name, and then by item description. (To sort by two fields, just add a comma between both in the ORDER BY clause (e.g., ORDER BY projectname, itemdescription). – Sean Thornton
      1. 
      2. 
   4. List the total final price (defined above) grouped by each supplier name. Round the result to two decimal places. Sort the result in descending order by total final price.

– Sean Thornton

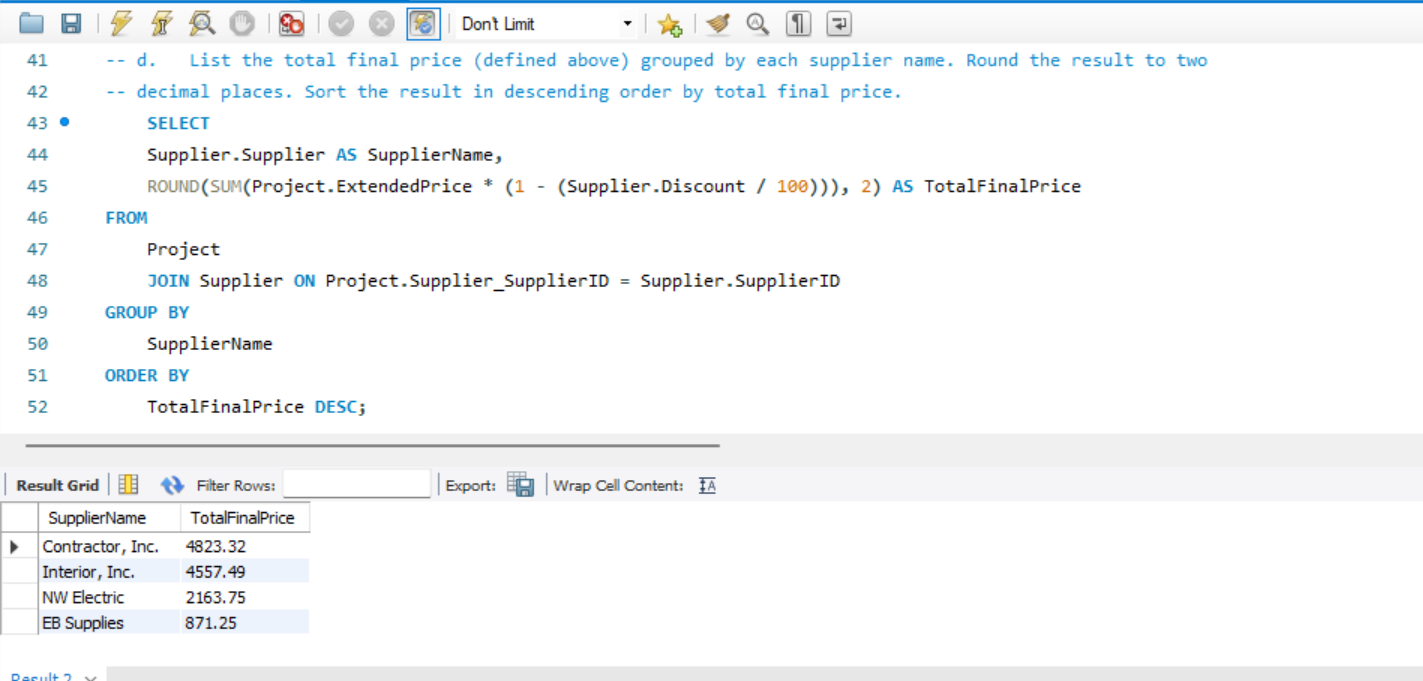
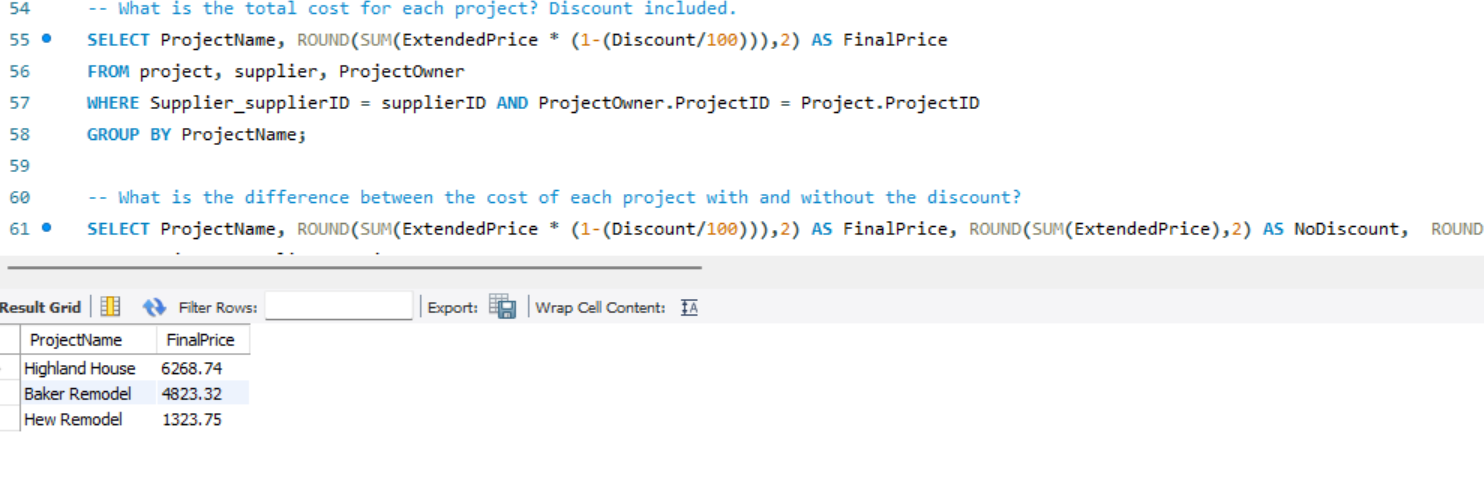
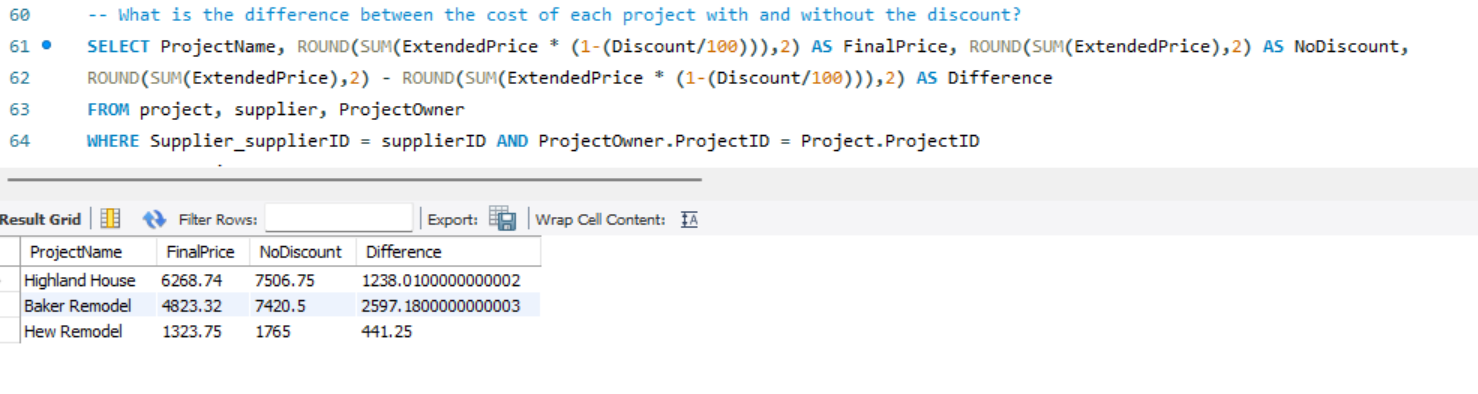
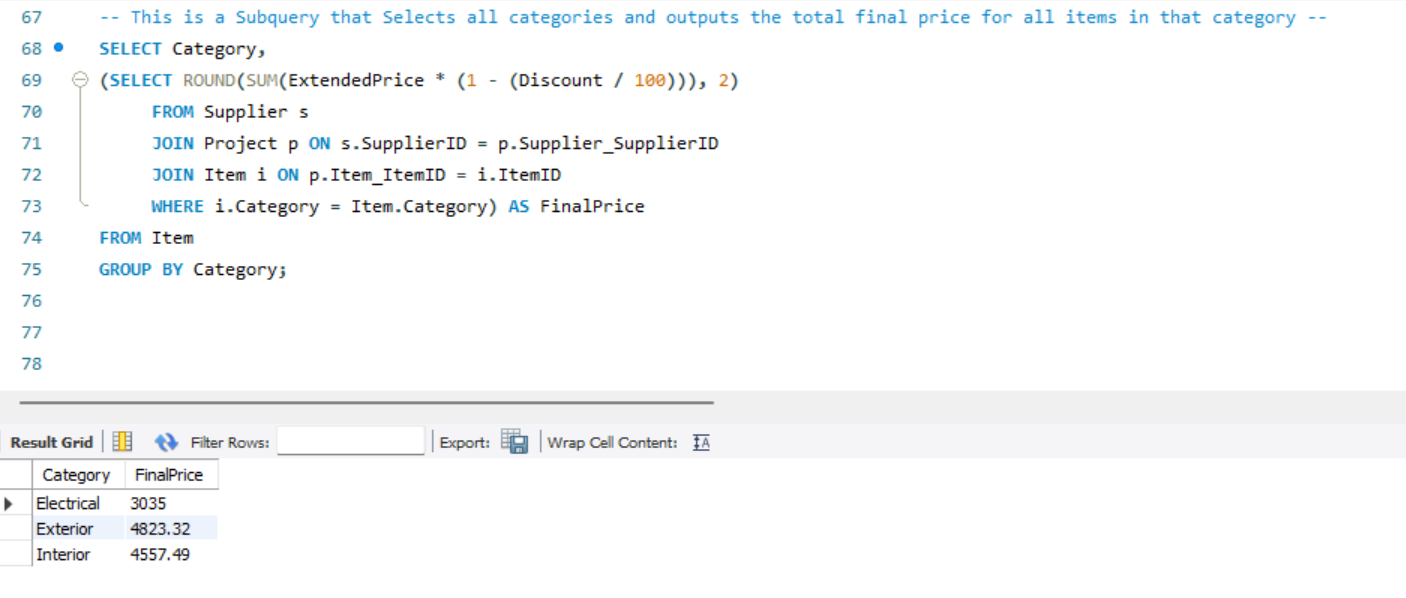
* + 1. 
  1. Add *three* other queries of your choosing but unique to your team. One of them must use a subquery. – Alan Tung & Justin Sanchez
     1. 
     2. 
     3. 

Figure 1: Project Equipment List

