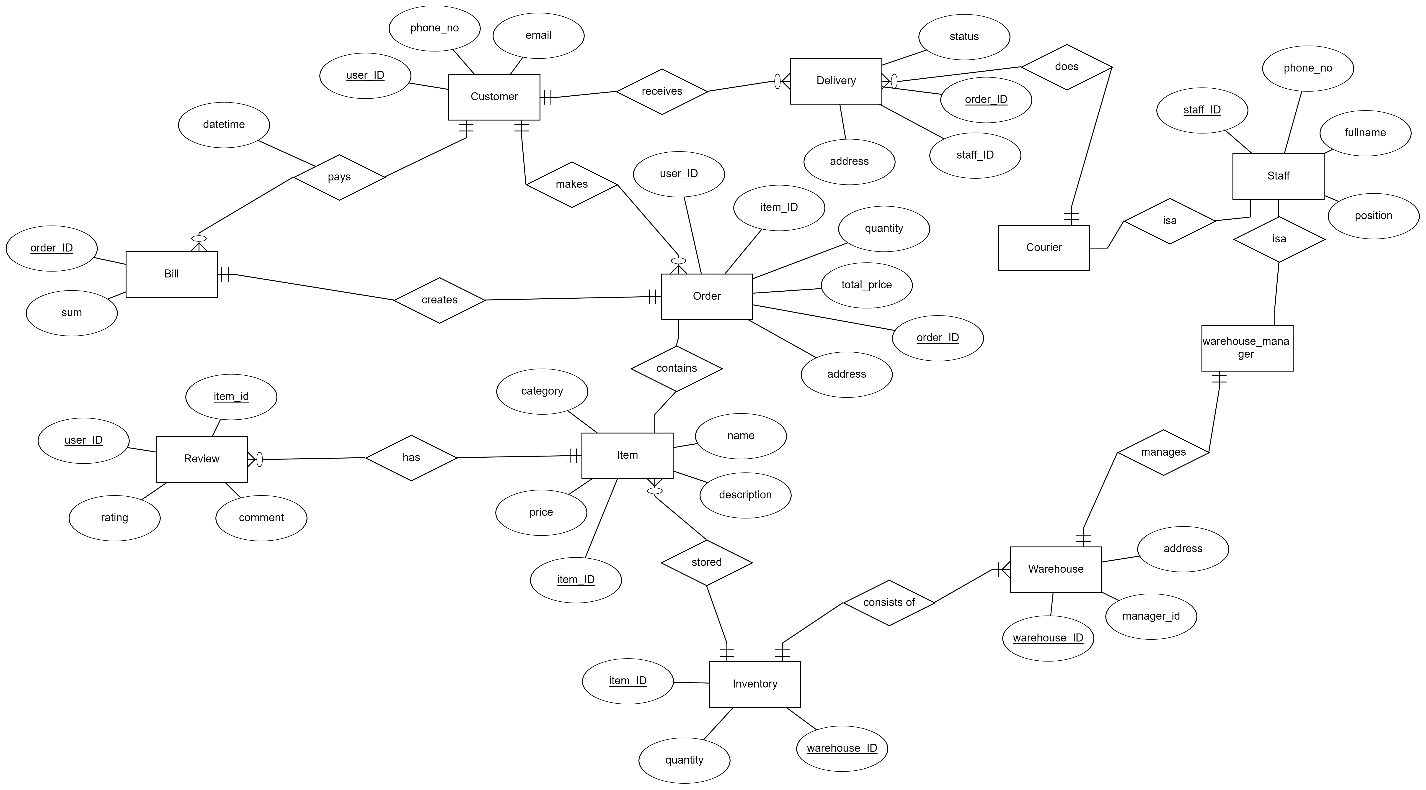
REPORT

Introduction:

This project is a simple design of an e-commerce system, in this example of a furniture store. But the design can also be suitable for selling books, parachutes or even hoods. The system includes tools for inventory, personnel management, contains a basic transaction system. The system is in the third normal form, which means minimal load on the hardware: no redundancy, improved data integrity, and efficient storage and retrieval of data.

ER diagram for the system (also included on the github page):



On the diagram you can see entities, its relationships and attributes.

To elaborate on the normalization of the structure:

As I mentioned above, structure follows 3NF, which means it follows 1NF and 2NF too. But what does it mean?

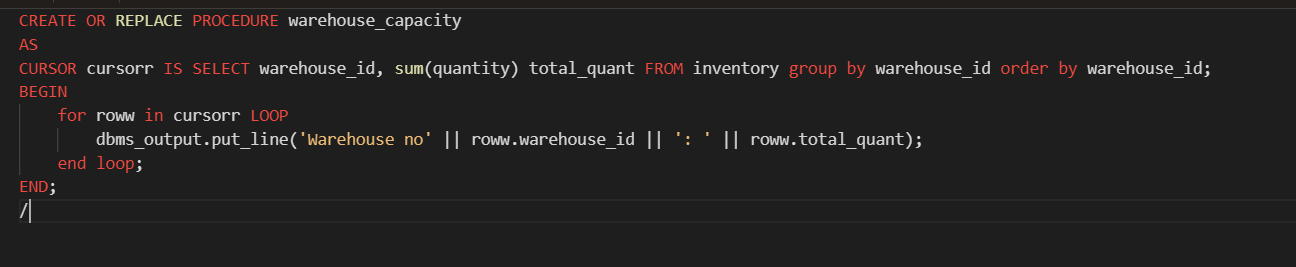
First Normal Form (1NF) requires that each table cell should contain a single value, and each record must be unique. You can examine the structure and see that that it follows this form. Although staff and customers can only have one phone number, but who cares anyway.

Second Normal Form (2NF) requires that structure is in 1NF already, and has no partial dependencies. That is, all non-key attributes are fully dependent on a primary key. Also check, moving on…

Finally, Third Normal Form (3NF) says that no table should have transitive partial dependency. In other words, a table satisfies 3NF, if and only if for any non-trivial X depends on Y either X is a superkey or each attribute in Y is contained in a key.

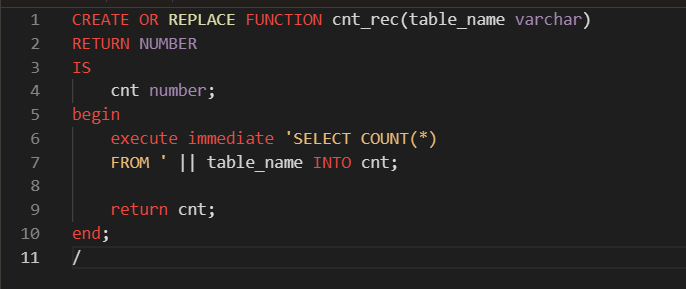
Now to coding part:

Procedures and Functions are available on .txt file.

First task was to create a procedure which does group by information. Code:

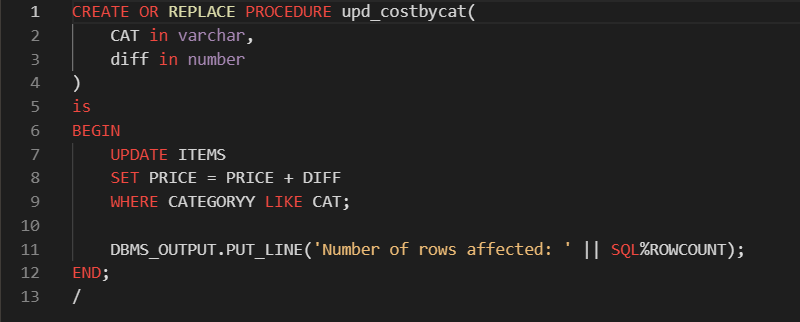
This procedure returns number of items stored in each warehouse. First, I wrote query that does what I need. And then used cursor to save the query output, after which used loop to iterate through every row displaying id and total quantity.

Function which counts number of records:



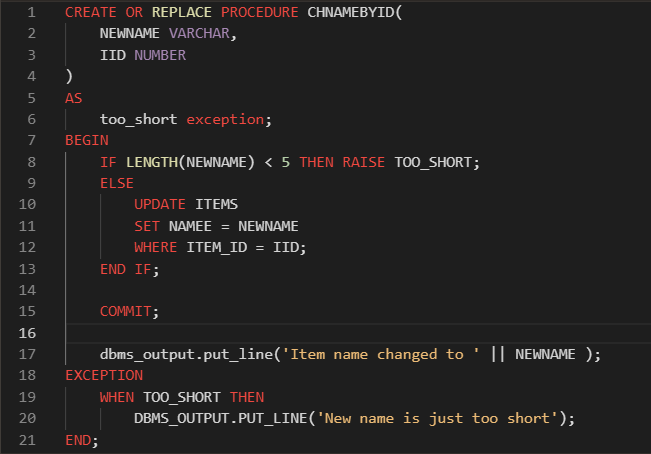
As the name suggests this function counts number of rows given the name of the table. I used ‘execute immediate’ keywords so that query will be dynamic, giving output depending on user input.

Procedure which uses SQL%ROWCOUNT to determine the number of rows affected:



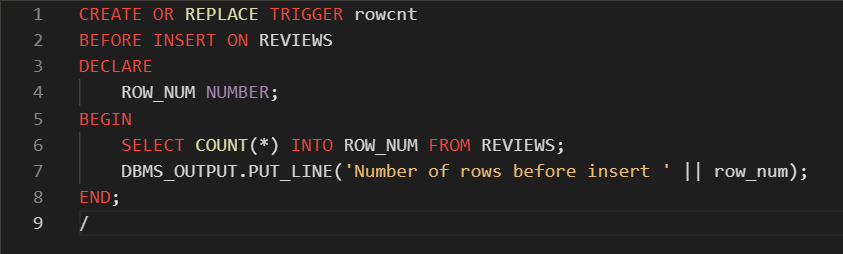
This procedure is used to increment prices of items based on the category. Procedure takes category name, and value of increment. SQL%ROWCOUNT is automatically created, so I didn’t need to initialize or give value to it. Easy as that

Add user-defined exception which disallows to enter title of item (e.g. book) to be less than 5 characters:



Procedure called CHNAMEBYID changes name of the item given its id and new name. But if new name is too short (less than 5 symbols) it will raise an exception.

Trigger before insert on any entity which will show the current number of rows in the table:



This one is really simple. We create a trigger that executes COUNT(\*) and displays it before insertion on given table.