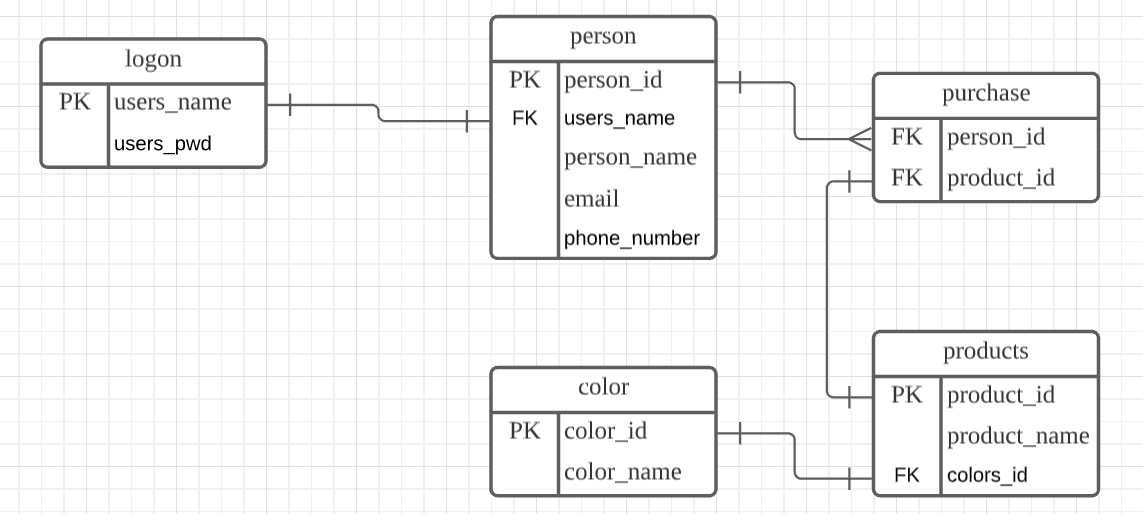
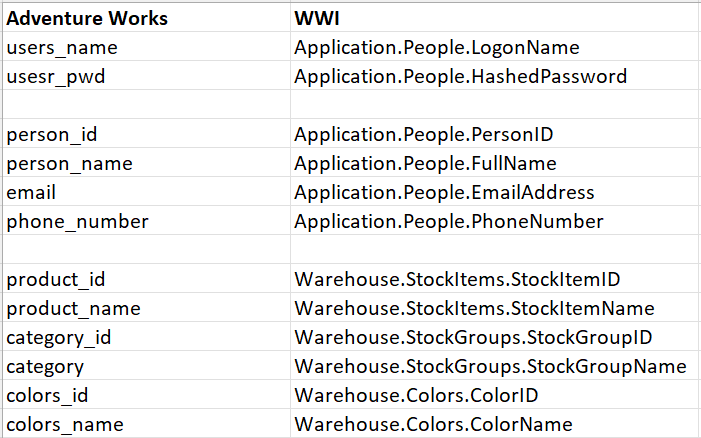
**Write a short essay talking about a scenario: Good news everyone! We (Wide World Importers) just brought out a small company called “Adventure works”! Now that bike shop is our sub-company. The first thing of all works pending would be to merge the user logon information, person information (including emails, phone numbers) and products (of course, add category, colors) to WWI database. Include screenshot, mapping and query.**

The entities and relationships of Adventure works are as below:

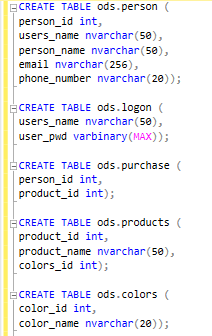


We assume that user\_name used to log is an email address. The passwords are stored as encrypted data. One person only has one chance to sign up. One person can buy multiple bikes. Color has its id and name which follow the color code. We also assume that WWI follows the same color code. Since this is a bike company, all products are in bike\_category when they are merged in WWI.

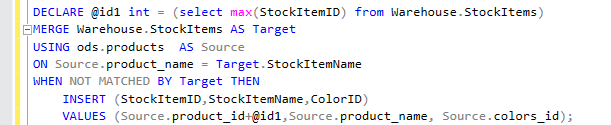
And this is the mapping of Adventure\_works columns and WWI:



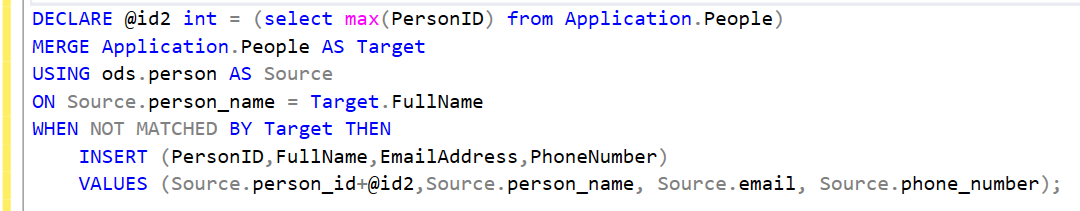
Creating tables:

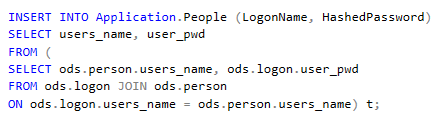


If the person is not in WWI, insert it as a new row. (we can assume that all are new haha) since the bike shop is new to WWI, all products in the bike shop are new to WWI. That is to say, all the products have to be inserted. The WWI and bike shop both follow the same color code, so if their color\_ids are the same, the colors are the same.

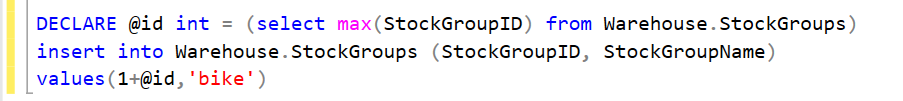


When we insert data into Application.People table, we have to judge if the names are the same because we don’t have to specific information, like SSN, to see if the same name represents different people. In this case, we will insert all the information, no matter whether they are the same or not.





Add a category called ‘bike’.



Since WWI and bike shop follow the same color code, we insert the color\_id and color\_name into WWI if it does not exist in it.

