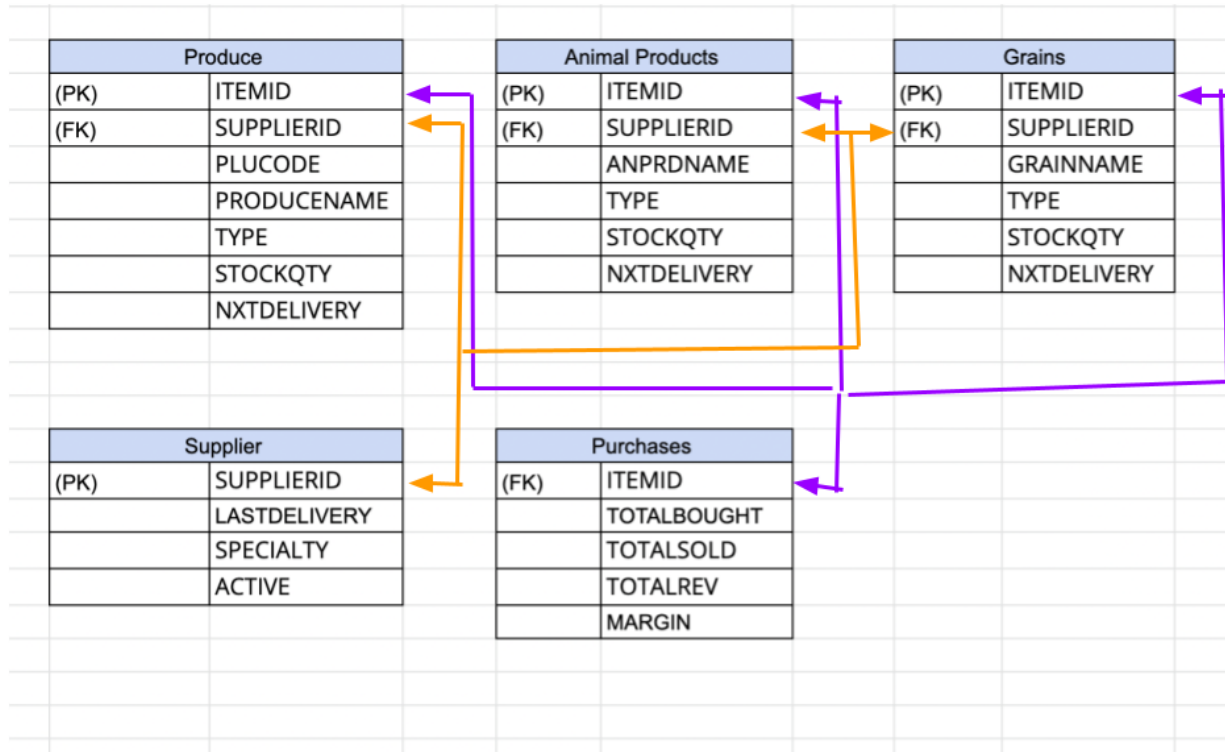


## CS 107: Database Evaluation

### Assumptions on Data Given



The tables above is the Entity Relationship Diagram that was build with Google Sheets and Google Drawings. We know that primary keys (PK) must be “unique” (or different than every row), and foriegn keys (FK) “hold the same value as the primary key of the related table,” (Relationships in SQL: Examples & Explanation, 2020). Based on this definition it is now time to make my assumptions.

1. The SUPPLIERID is a 1:Many relationship since the supplier is one entity providing different types of supplies (Designing a Relational Database: Tutorial & Overview, 2020). This is why the SUPPLIERID appears as a foreign key under Produce, Animal Products, and Grains

2. The tables Produce, Animal Products, and Grains all have an ITEMID that is the primary key. Due to this any changes or updates of the ITEMID, STOCKQTY and NXTDELIVERY will automatically update. ITEMID under Purchases is also set up in this way, but it is a foreign key, which means it is dependant on the other tables. Once an item has been purchased, everything else in the purchase table gets updated, TOTALBOUGHT, TOTALSOLD, TOTALREV and MARGINS.
3. Also, the ITEMID in Produce, Animal Products, Grains, Suppliers, and Purchases range in CHAR(5) and CHAR(10) meaning that it is stored in the correct and proper format. This will make it easier to query in the database.

As, I do not have all the information needed to know what is going with these tables, I am making assumptions based on the information I that I been given. Also, based on the assumptions that I have made I do not think this is a good choice for a business to have this the tables formatted this way. I believe that a one point as the business keeps growing they might need to modify and upgrade the database as this is not a sustainable database.

### **Database/SQL Version**

This is a relational database due to the “tables that are connected, or linked to each other by some common attribute” (Designing a Relational Database: Tutorial & Overview, 2020). This version of SQL is Oracle. The data on the tables has been partitioned into different section, instead of being a one big database. Oracle is know to being the biggest relational databases and these tables due fit in the Oracle description. Also, in Oracle “the data has separate logical structures and physical structures. The separation allows the physical storage of data to be managed without affecting the access to logical storage structures” (Oracle Databases: History &

Overview, 2018). This is where the data type CHAR() comes in as it is the function that keeps everything logically structured and also physically structured.

### **Keys**

As I briefly explained above the primary keys and foreign keys are very important in this databases. Produce, Animal Products, and Grains all have an ITEMID that is the primary key that is unique to each topic. The Purchases ITEMID is a foreign key and thus is dependent on the other tables, but it is still a unique number. The Supplier ITEM ID is a primary key not related to the other tables. This should be related to the other tables, since the other tables have NEXT DELIVERY. It seems like the Supplier primary key should be connected with product, Animal Product, and Grains.

### **Normalization**

First normal form (1NF) reduce redundancy in the database. All tables have a primary key, which Produce, Animal Product, Grain, and Supplier have, which makes them 1NF. These tables are not in Second Normal Form, because they do not meet the criteria. The Produce, Animal Product, Grain, and Supplier all have partial dependencies on the on the ITEMID the primary Key. This automatically disqualified the tables from going to 3NF. The only one that does go to 2NF is it clear of redudnace however it does not go to 3NF, because of the margin dependency on the reveninue.

### References

1. Designing a Relational Database: Tutorial & Overview. (2020, March 17). Retrieved from <https://study.com/academy/lesson/designing-a-relational-database-tutorial-overview.html>.
2. Relationships in SQL: Examples & Explanation. (2020, August 12). Retrieved from <https://study.com/academy/lesson/relationships-in-sql-examples-explanation.html>.
3. Third Normal Form in DBMS with Examples. (2020, March 14). Retrieved from <https://study.com/academy/lesson/third-normal-form-in-dbms-with-examples.html>.
4. First Normal Form in DBMS with Examples. (2020, March 16). Retrieved from <https://study.com/academy/lesson/first-normal-form-in-dbms-with-examples.html>.