Terms are a basic introduction to most fields one may study. Overall, there are five simple terms that could be classified as essential to any aspiring database engineer: Database, table, record, field, and value. At the very top, we have the term Database. This describes the element to be crafted containing all other four terms. Essentially, a database is a storage of many sets of data, or tables, to provide organization and information to the end user. Shops may use databases to store records of clients, employees, retail items, and many other things associated with their daily business. Within the database, we have tables. Tables provide a means of organizing all this data into easy-to-access chunks. If the same business would need a database constructed, they would likely have tables associated with each aspect they were dealing with. Through third normal form in the database, tables may exist detailing client relations, customer transactions, and even personal employee information. These tables consolidate a group of datapoints like employee phone numbers, addresses and so forth into one organized structure, and relate to other tables in the database. For instance, the Customer table may relate to the Product table by featuring fields for a SaleID to give persistence to this particular sale and make sure the store is able to refer to this particular sale in the future. They would also feature a ClientID field to keep track of exactly who purchased the product, and possibly even an EmployeeID field to determine who was at the register to sell this item to the customer. With this information, you may be able to infer what a Field is. Each table has a number of rows and columns to it. A field defines each column of the table. For the example listed above, the table consists of many columns like SaleID and EmployeeID. These would be the SaleID field and so on. A record in the database is simple the other axis of the table. If a field is the measure of what goes from top to bottom, a record is what goes from left to right. For example, there may be many fields as described above to make up the table. But the database would be useless without entries. The record is created as each transaction goes through, and consists of data about the primary key of the table (in this case the SaleID). For a record you would have a number of values starting with the SaleID, consisting of EmployeeID, ClientID and so on. Then at the smallest chunk of our database, we have values. Values describe each bit of information located in each cross-section of rows and columns. The value for SaleID would need to be a unique identifier to distinguish it form other sales, as is the point of this field. One SaleID value may be 0, and another SaleID value may be 1 going up and up with each transaction. These values are simply the single bits of information people or programs can gleam from a field in a database. With this information, it has been demonstrated that in its most general form, a database encompasses the entire collection of data consisting with many tables. Tables organize data consisting of many Fields. These many fields form one or more Records. These Fields and Records consist of many values.

All these terms would mean nothing as useless data floating throughout the cosmos, however. There must be a means to organize the tables to show meaning to the dataset you’ve created. Elsewise you’ve just made an incredibly long list of datapoints that are impossible to search through, and the database constructed is useless. These organizations come in the form of Relations. Relations help to keep a database efficient while reducing errors, duplications, and inconsistencies present throughout the dataset. All these relationships come together to from a Relational Database. In such a database tables could be linked together, duplicated data could be minimized by using different keys and connections, and updated information wouldn’t cause catastrophic failures throughout the database. In the Above example, the use of ID tags are prevalent instead of using English names. Well, when diving further into such a table, it may be that the English name is nowhere in sight. Employees are listed via number, and because of this it may be difficult to read at first. This is because there is a relation between tables. There exists another table in which these English names are present. For a more concrete example, the EmployeeID may exist as a foreign key within the Sales table, but it may be the primary key inside the Employee table. There may be fields for EmployeeID, First name, Last Name, Phone Number and so on. Thus, since the EmployeeID attribute is present as a primary key with a record associating it, there is context behind its appearance in the Sales table in which it is used not as a primary key. Its this context that defines a relation, and the same could be for many different attriburtes in normalized databases. These all work together to create a Relational Database, in which the contents of one table give context to the fields of another.

Now that there’s a conceptual model of the database to be created, let’s look at Microsoft Access as a form of database production. As described in Chapter 1 of the *Access 2019 Bible*, there are 6 main objects to access to make a database: Tables, Queries, Forms, Reports, Macros, and Modules (Access 2019 Bible, 2019). A table is a collection of values in Records and Fields. A query is an instruction that searches your database to return the requested value(s). A form is a simple User Interface relating to your data. Forms can be used to display data within the database, and to provide easy access for users to gather data from within your database through the use of these UI elements. Reports are a means of displaying data acquired through queries. Macros perform automated tasks such as cleanup, or data reporting without input from the user or database engineer. A Module is a simple library of programming utilizing the Visual Basic for Application language. Now that terms are properly defined, the *Access 2019 Bible* lists a five-step process to creating the fundamentals of databases anywhere. This differs from and contradicts other multi-step processes, such as the ones found throughout the internet such as Tutorials for Microsoft Access (LuvCite, 2013). Due to this, the content of this paragraph will cover strictly what the *Access 2019 Bible* covers in their specific 5-step process. Step 1 introduces conceptualization. It is important that you ask yourself questions about how the industry is changing. What are commonly used tools in the field today? What are common storage solutions for the data you’re working with? How will you process this data? During this step, and all others subsequently, it is essential that you keep regular contact with your client to ensure your work adheres to their vision. Step 2 involves formatting your reports. Make sure you thoroughly understand exactly how your report will look, since that is what your client will see. Step 3 details the design and layout of your data. This step focuses on consistency and error minimization. Make sure your reports are uniform to represent the same data, and that relationships are clear. Step 4 will walk through designing tables. What data will you need to display in your table? How many fields will you need? How will this accumulate in the record? Make sure you make space for all necessary elements in a clear and concise format. Pay particular attention to how various keys will fit in. Are your primary keys listed as foreign keys in a valid format? Are your numbered relationships appropriate given the types of data you are representing? The final step will be about designing your form. Now that the data is made, it’s time to design what it will look like to your client. The three types of objects are as follows: data entry fields to let the user input data, special controls to give the user control over that data, and graphical objects just as color and shapes to help the data stand out, and be easily recognizable. With that designed, you should have a valuable database constructed that holds the data specified by the user, and is easily accessed by the user to facilitate operation.

# **References**

(2019). In M. A. Kusleika, *Access 2019 Bible.* John Wiley & Sons.

LuvCite. (2013, November 18). *DESIGNING AN ACCESS DATABASE: SOME SIMPLE STEPS FOR YOUR NEW MS ACCESS DATABASE*. Retrieved from Access Database Tutorial: https://accessdatabasetutorial.com/desiging-an-access-database-simple-steps-ms-access-database/