

Building Data Models in MS Visio 2010

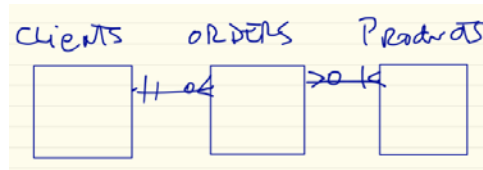
Prepared by Alberto Espinosa, last revised on November 10, 2013

MS Visio is just a drawing tool. I provide below some guidelines to help you build your data models, but it is much more important to understand the conceptual foundations of data models and relational database design because, if you do, the drawing in Visio is then the easy part. So, it is very, very important that you sketch your data model on paper first. There is no need to include all the fields in the sketch. Simply sketching all your entities (i.e., data tables) with lines denoting how they relate to each other may be of great help before you jump into Visio. Again, as you draw develop your data model, think of your model in business terms and ensure that it makes business sense. For example, if you have a model composed of 3 entities or tables (Clients, Orders and Products), the way to explain these in business terms is something like this:

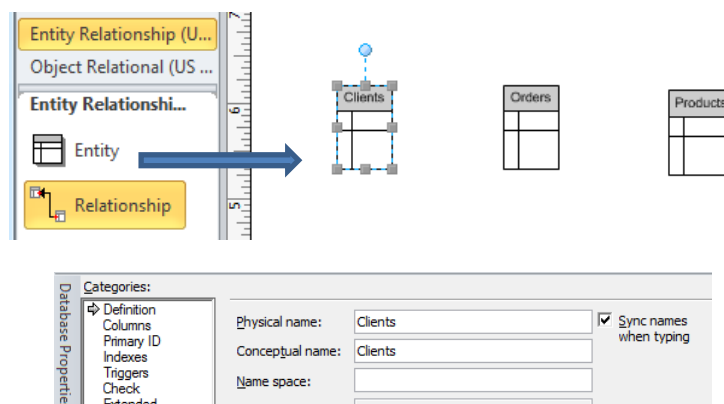
- The Clients table contains one record for every client; the Orders table contains a record for every order placed; and the Products table contains a record for every product sold by the company.
- A client can place many orders and an order must be associated with a single client. An Order can be placed for 1 or many products and a product can appear in many orders (hint: this many-to-many logical relation points to the need for an “intersection” table, which in our class example we called “LineItems”).

To build a data model like this in Visio, here are the steps:

1. Do a quick sketch of your data model on paper. Just draw a box for each entity and link them with lines representing relationships. This is what is called a “conceptual data model”. That is, one that simply shows your main entities and how they relate to each other.

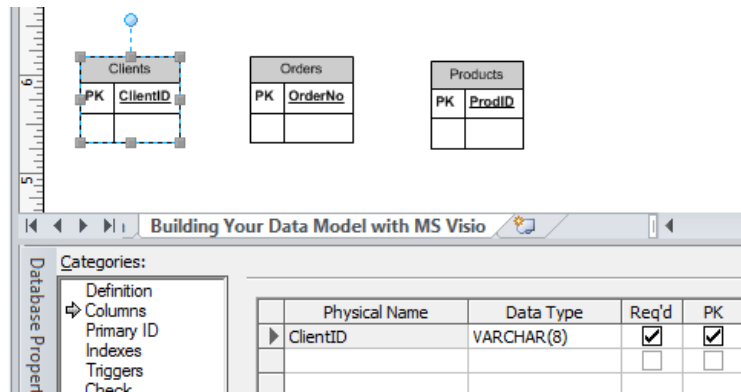


2. Open Visio → File → New → Software and Database → Database Model Diagram
Then Draw all the main entities needed first by dragging the entity symbol to the canvas and then clicking typing the entity name in the Definition section in the Database Properties Window

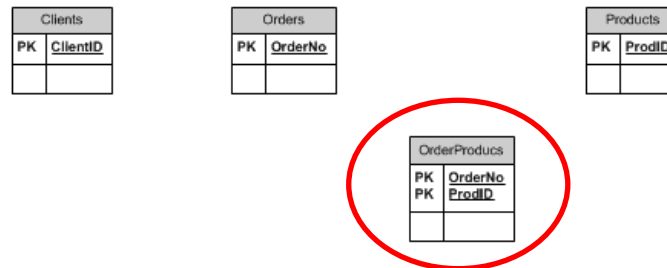


- Then add the PKs for each entity and inspect them to ensure they are correct. To add a field, click on the corresponding entities. Then select the Columns section in the Database Properties Window and type the field name. To make it a PK, check the corresponding box.

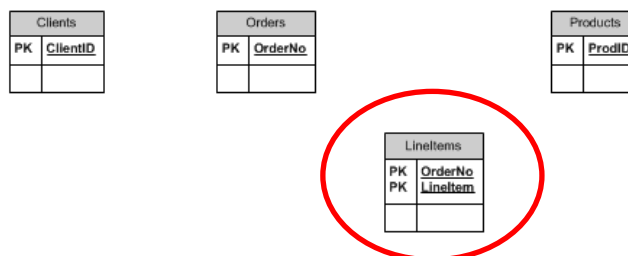
IMPORTANT: if you select the wrong PK everything will fall apart in your model. Again, please review your model to ensure you have the correct PK's.



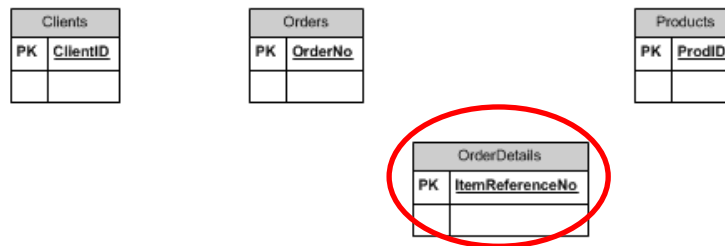
- Print your Visio diagram and draw your relationships by hand once again. If you have any many-to-many relationship, decompose it into two one-to-many relationships using an intersection table. The intersection table usually has a name connecting the two intersecting tables and a composite key with the two PK fields of the intersecting tables:



However, other options are possible. In the class example, we used a different table name and PK designation for the Lineltems table, but it is still technically an intersection table because it is used to break up the many-to-many relationship into two one-to-many relationships.



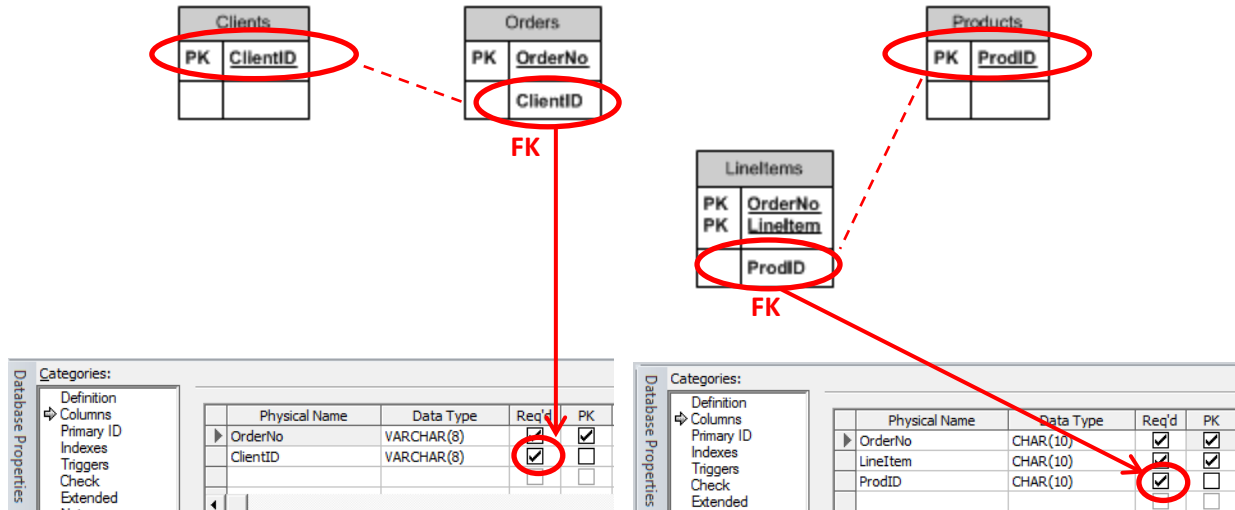
Yet, one more option is possible with a single PK, like ItemReferenceNo, provided that it is unique and not duplicate:



5. For our illustration in this example, we will use the second option with the Lineltems table. Once you are certain that you have all the FKs right, go back to Visio and click on each entity and create the necessary FK fields and check the box to make it a required field (the attribute will be boldfaced when you do this). Please note that you will not see the FK mark next to the FK field until you draw the relationship line (see 7 below).

Remember this is an important conceptual fact: relationship links have two ends, one end of the relationship line points at the field that needs to be validated (i.e., the FK) and the other end points at the field where the valid data resides (i.e., the PK). For example, the ClientID field in the Orders table is the FK side of the relationship line (i.e., in order to place an order you need to be a valid client, so we need to validate what is entered in this field to ensure it is correct), whereas the ClientID field in the Clients table is the PK side of the relationship line (i.e., this table has the list of valid clients). Similarly, the ProdID in the Lineltems table is the FK because the ProdID you enter in this table needs to be validated against the ProdID's (PK) entered in the Products table (see figure below).

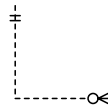
VERY IMPORTANT: there is a glitch in Visio that sometimes makes it draw the wrong cardinality. I've tried several workarounds for this problem and what I describe here appears to be the only one that works. So please pay close attention. Before you draw a relationship line, identify the two tables you need to link and the common fields you need to link. Then clearly identify in which of the two tables has the FK field you need to validate and which one has the PK field where you will do the validation. Then go to the table that contains the FK and check the Req'd box (see figure below). This will make the FK field required and bold face it in Visio. Please ensure that you typed the field names correctly. Again, any mistakes in defining and declaring your PK's and FK's will cause everything else to fall apart. If you do this BEFORE creating the relationship, Visio will display the correct cardinalities.



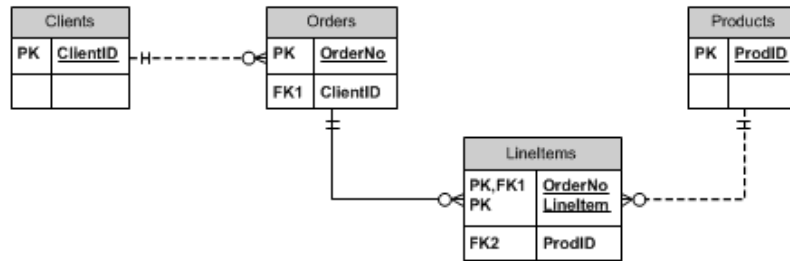
6. Now you can build the relationships. **Important: ensure that you know which end of the relationship line you will connect to each entity or Visio will mess up your model.** But before you connect the relationships, let's ensure that you have the proper modeling notation. We are going to use the "Chen" notation, also known as the "Crow's Feet" notation. Just drag a relationship line to the canvas without connecting it to anything. Your line may look like this, which is another popular notation:



If so, go to Database → Display Options → Relationship tab → check the Crow's feet box → OK. Your relationship line should now look like this:

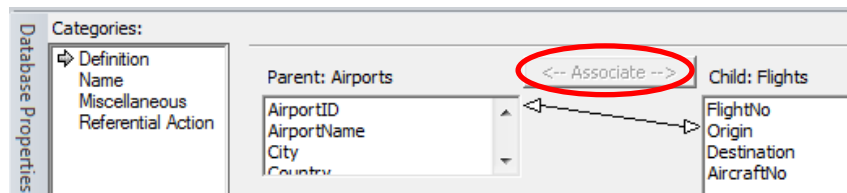
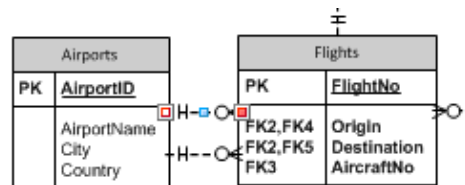


7. Now pick one relationship (e.g., Clients to Orders) and drag the "one" side of the relationship to the table that has the PK (i.e., Clients) and the "many" side of the relationship to the table that has the FK (i.e., Orders). The rationale is based on what we call "cardinality". That is, one client can place many orders, but an order must be associated with a single client. Thus, naturally, the many side of the relationship needs to connect to the table that has the FK. Drag and connect all the necessary relationships in Visio. Note: don't worry about cardinalities just yet. We will adjust this shortly.

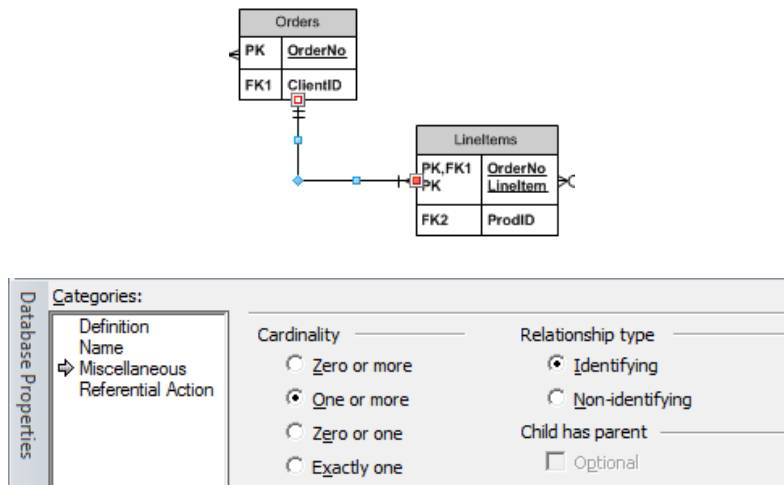


Notice a few things. First, PK's and FK's are automatically associated by Visio when it finds that they both have the same attribute name (e.g., ClientID). Then, not all the cardinalities are exactly how we want them. And finally, some lines are solid and some are dotted. I discuss each of these next.

- On the first issue, click on the relationship line Orders-Lineltems and then select Definition in the Database Properties window. You will notice that Orders and Lineltems are associated through the OrderNo field. This is the correct association, which is what Visio does by default. Most of the time, this is what you need (i.e., PK and FK have the same attribute name), but sometimes you may need to link two attributes with different field names. In this case, all you need to do is disassociate the association created by Visio, then click on the two fields you want to associate and click on the Associate button. Here is an example we covered in class in which we associated the AirportID attribute in the Airports table, with the Origin attribute in the Flights table:

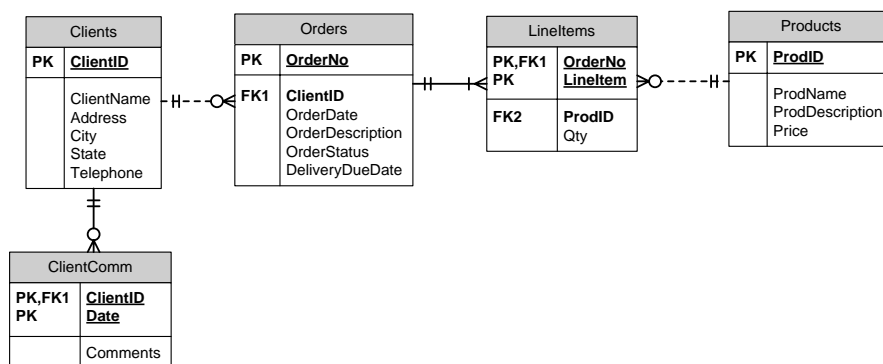


- About fixing cardinalities, in the example, a client can place zero or many orders and an order has to be associated with one and only one client, which is correct. No need to do anything. But, an order can have zero or many line items, which is incorrect (we want one or many because an order has to have at least one line item), and a line item has to be associated with one and only one order, which is correct. Let's correct the cardinality on the Lineltems side of the Orders-Lineltems relationship. Click on the Orders-Lineltems relationship line, then select Miscellaneous on the Database Properties window, then click on the One or more radio button. Your relationship should be correctly displayed now:



10. About the solid and dotted lines. Visio will automatically draw a solid lines for “identifying” relationships and dotted lines for “non-identifying” relationships. What does this mean? This is not very important for us, but it is good information to have. If a FK attribute is inside a PK, then the relationship is said to be “identifying”. If the FK attribute is outside the PK, it is said to be non-identifying. This is important for more advanced data modeling operations, but for us is just an informational issue. If you click on the Miscellaneous section you can click on the Non-Identifying and Identifying buttons back and forth and see how the FK moves to the PK section and vice versa, and how lines get solid and back to dotted. When you are done, please ensure that your attributes and relationships are correctly drawn.

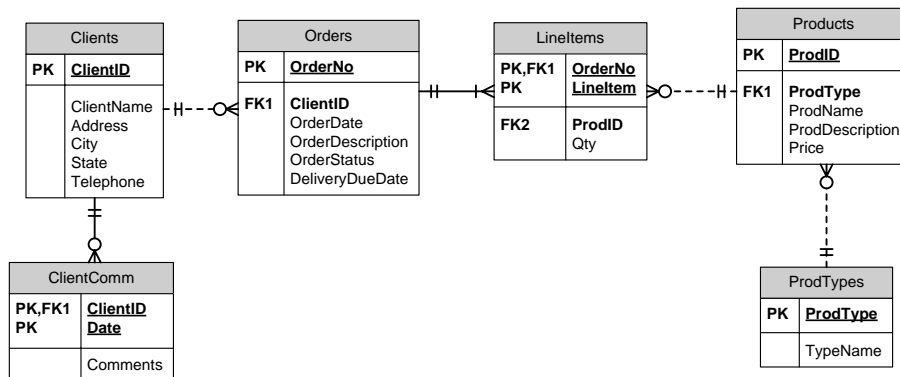
11. Now that you have all the entities, PK’s, FK’s, relationships and cardinalities, you have a good logical data model. It is very important to pause at this time and review your data model. Most problems with data models are due to something not modeled correctly in the logical data model. Once you are sure that everything is correctly modeled, and only then, add the remaining non-key attributes:



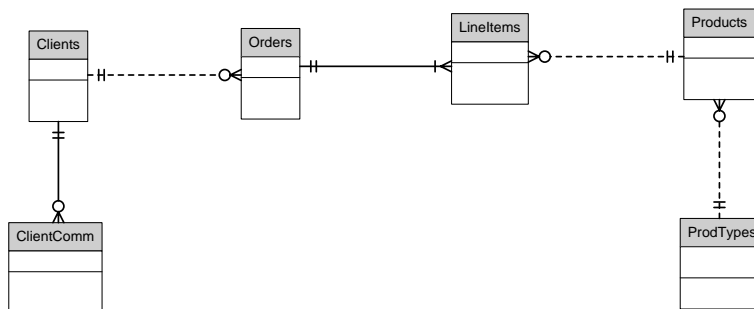
12. Now, let’s analyze this data model. You will notice that I added the ClientComm table to the model. Is this an intersection table? Not really, it does not intersect two tables, only one. This is just a simple data table with a dual PK.

13. Now let’s analyze one more thing. How do you add lookup tables? A lookup table is a convenient way to add standard descriptor fields to categorize your records. For example, look at the Products

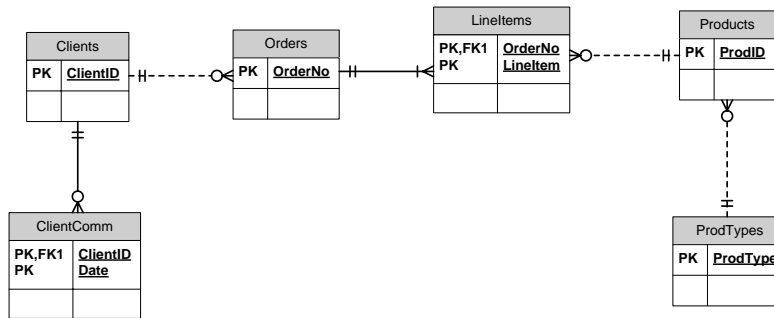
table. How do you know what type of product a given product is? For example, you may sell toys, instruments, books, clothing, etc. But there is nothing on the Products table to help you classify your products. The first simple step to correct this is to create a new field in the Products table for this purpose. Let's call it ProdType. But if this is all we do, then we are leaving it up to the data entry user to type the ProdType correctly. If the user makes mistakes (e.g., different capitalization, misspellings, blank spaces, zero instead of O, etc.), then our database queries will not work properly. It is a long time tradition in database design to correct this with "lookup" tables. For example, we can create a table called "ProdTypes," which will have one record for each product type carried by the store. A record in this table will be uniquely identified by a PK, which we will name ProdType (notice that the table is name as a plural because it is a collection of records, whereas the field name is singular because it is a single record in the table). The ProdType field in the ProdTypes table is the PK, which you can now link to the ProdType in the Products table as a FK. Now once you enter a product type for a product, it has to match a valid product type in the ProdTypes table:



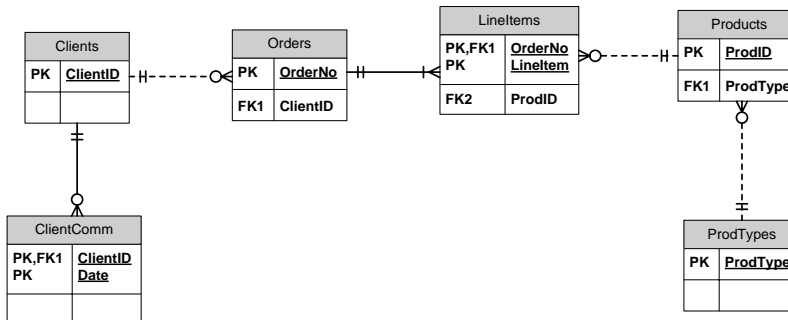
14. OK, one last very useful trick in MS Visio. When you are done with your data model it may look a bit cluttered, especially if you have lots of tables. Many times when you show this to your clients they get confused, which may lead them into detailed questions that are unproductive until you have reached agreement on the conceptual design. This can be easily resolved by selecting Database → Display Options → Table tab. Now you can show or hide anything you wish. For example, if I uncheck everything in the Display group, the data model will look like this, which is a **conceptual data model**:



If you then check the PK, Vertical Line and Annotations boxes, it will look like this:



And if you then check the FK box, it will show the **logical data model**:



Some questions for you to consider (as analysts you should be able to answer these and therefore are great exam questions):

- Can you identify and model other possible intersection tables (i.e., many-to-many relationships?)
- Can you identify and model other possible lookup tables?
- How would you model other transaction tables (e.g., shopping cart, wish lists, etc.)
- Which tables would you add to capture more analytics data, in addition to transactional data (e.g., shopping cart behavior, consumer sentiment, product type purchase patterns, etc.). Think of the Target example, if you wanted to analyze purchase patterns from pregnant women so that you can predict that they are pregnant in the second trimester (rather than the third trimester as competitors do), how would you modify the database?

ENJOY!!