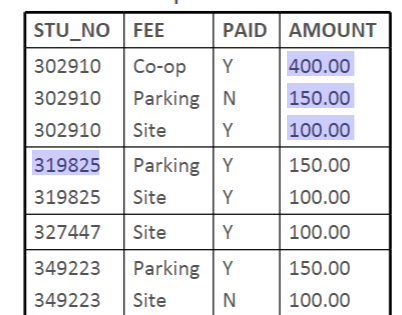
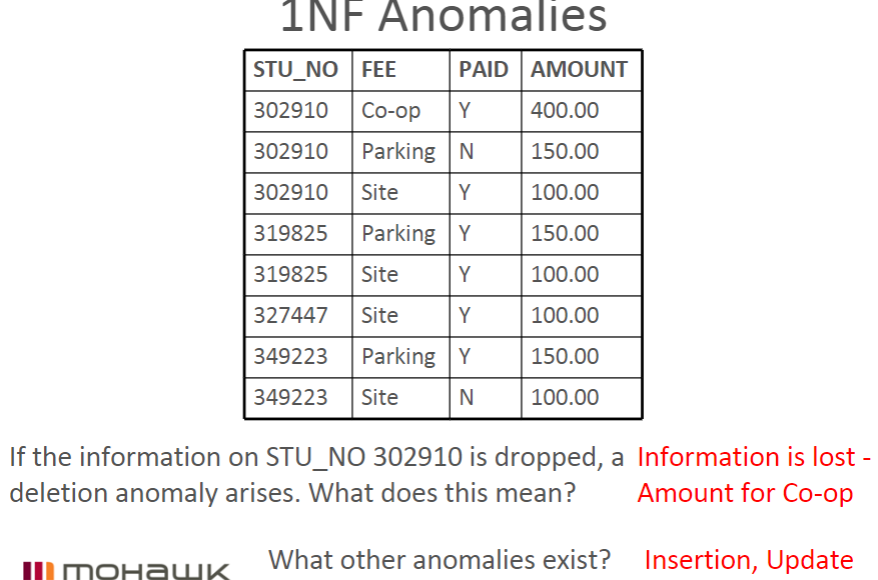


**Three Basic Requirements Of 1NF Table**

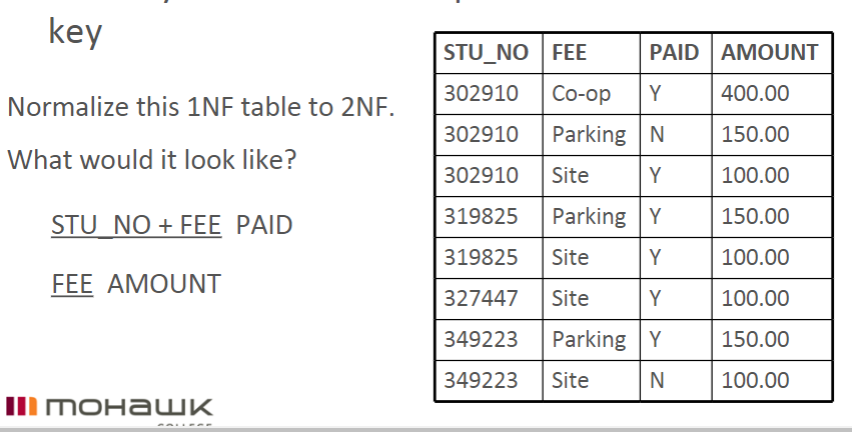
* Cells must be single valued
* All values in a column must have the same domain
* Each row must be unique





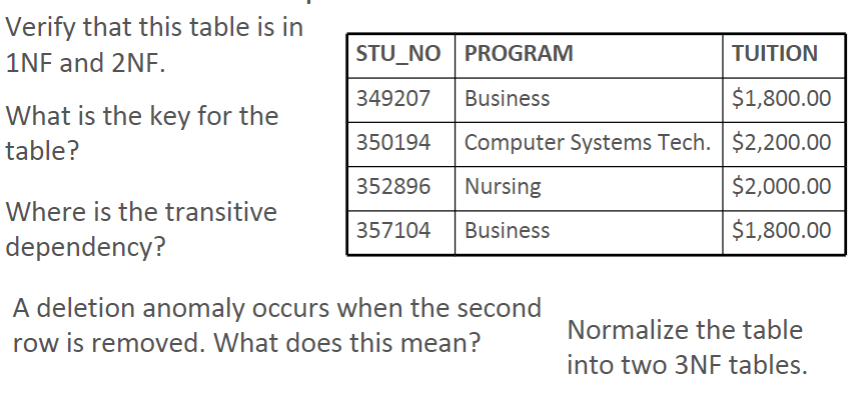
**Second Normal Form**

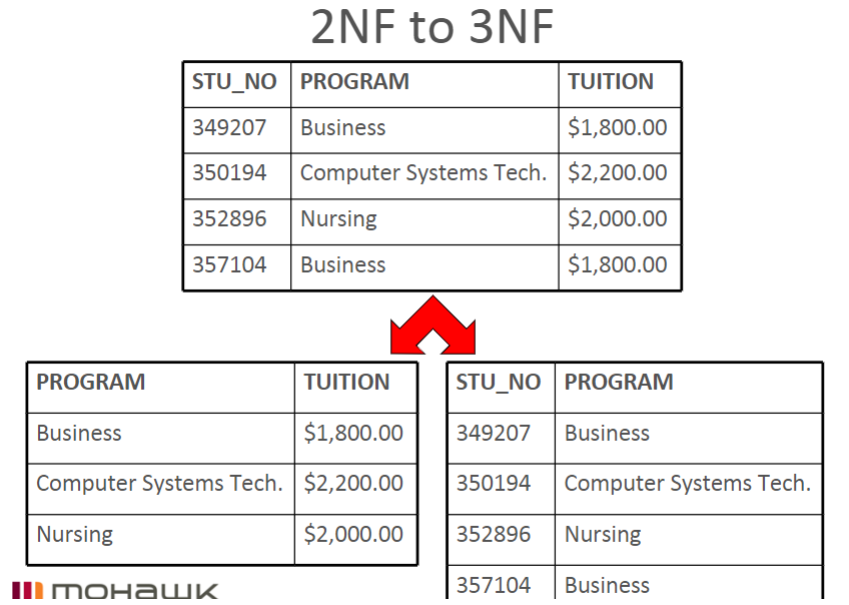
* A relation is in second normal form if all of its non-key attributes are dependent on all of the key



**Third Normal Form**

* A relation is in 3NF if it is in 2NF and has no transitive dependencies





**Normalization Case Study Revisited**

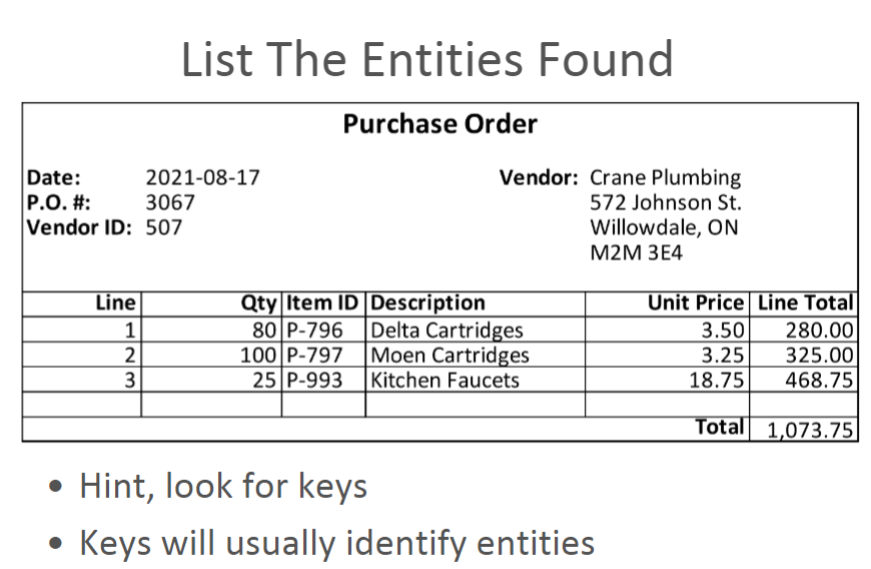
* Pipes N Things Inc.\
  + A distributor of plumbing parts and fixtures
  + Purchase from manufacturers
  + Supply retailers
  + Designing corporate data base

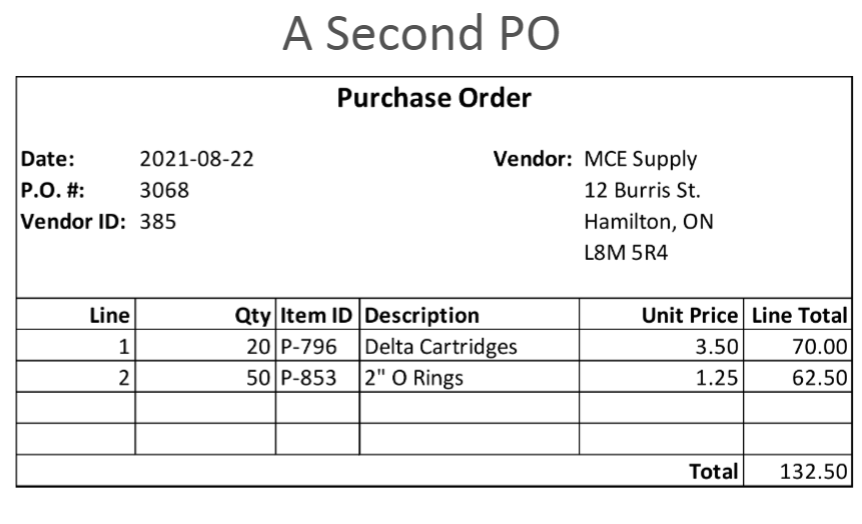
**Purchasing Department**

* Determine vital entities
* Purchase Order document is critical
* This purchase order has a line number, which is simply a count of the number of items that appear

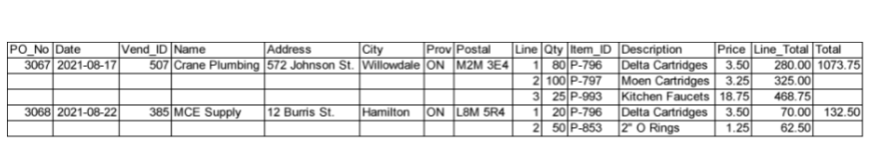
on each particular purchase order

* + Some times this is known as a sequence number





**Convert The Purchase Order Documents Into A Two Dimensional Table**



* Results in an unnormalized table
* Multiple occurrences of data exist
* We can find cells that have more than one value
  + Item\_ID, Desc, Qty, Description, Price, Line\_Total

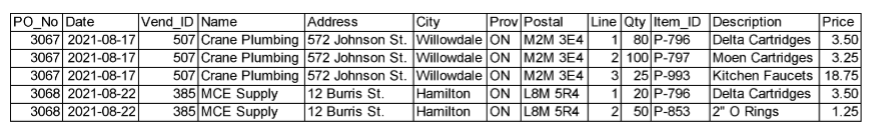
**Normalization – The First Step**

* Transforming an un-normalized table into one that is in First Normal Form
* 1NF has no repeated occurrences of data items in any of its cells
* Go back to the table and put it into 1NF by repeating some of the data values and

removing calculated values

**Normalization – The First Step**

* Repeating Values For Attributes
  + PO\_No, Date, Vend\_ID, Name, Address, City, Prov, Postal
* Calculated values (Line\_Total, Total) removed



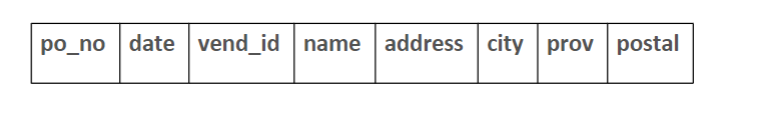
What is the table’s key?

...PO\_No + Line

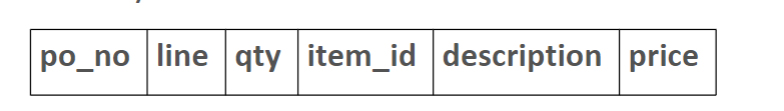
**Normalization – The Second Step**

* Isolating
  + What data items are dependent on parts of the key
  + What data items are left dependent on the whole key
* Dependent on PO\_No
  + Date
  + Vend\_ID
  + Name
  + Address, City, Prov, Postal

The resulting table with PO\_No as the key

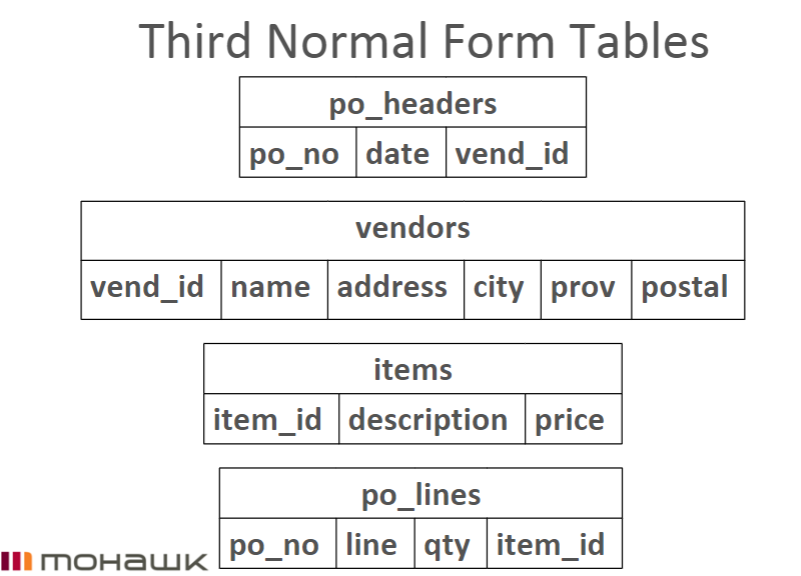


* Dependent on PO\_No and Line
  + Qty
  + Item ID
  + Description
  + Price
* The resulting table with po\_no and line as the key



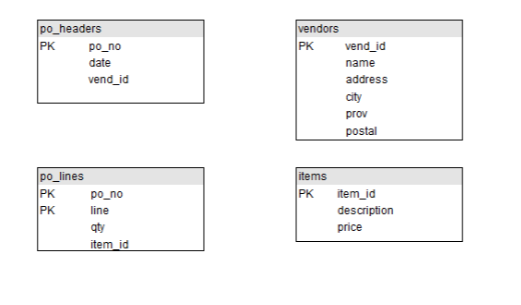
**Normalization – The Third Step**

* Separating any entity that deserves its own relation in the database
* Name each entity
* Look for a key field that has a dependent attribute; a key for another entity
* Is this occurring in our 2NF tables?
* If so, where?



**Creating the ERD**

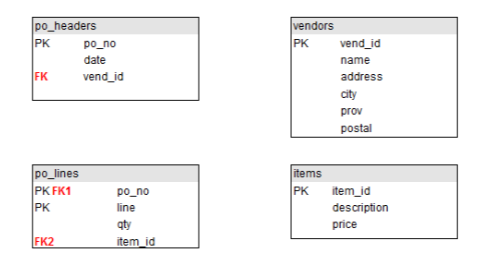
* Draw the entities with their primary keys



**Creating the ERD**

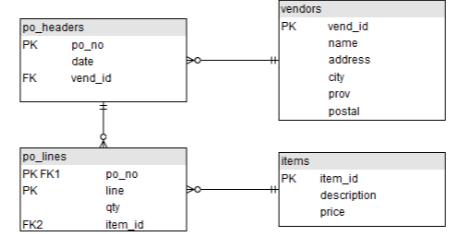
* Examine each primary key to see if it appears in another table, if so it is a foreign key in the

other table



**Creating the ERD**

* Each foreign key means there is a relationship between the two tables
* The primary key is on the “one” side and the foreign key is on the “many” side



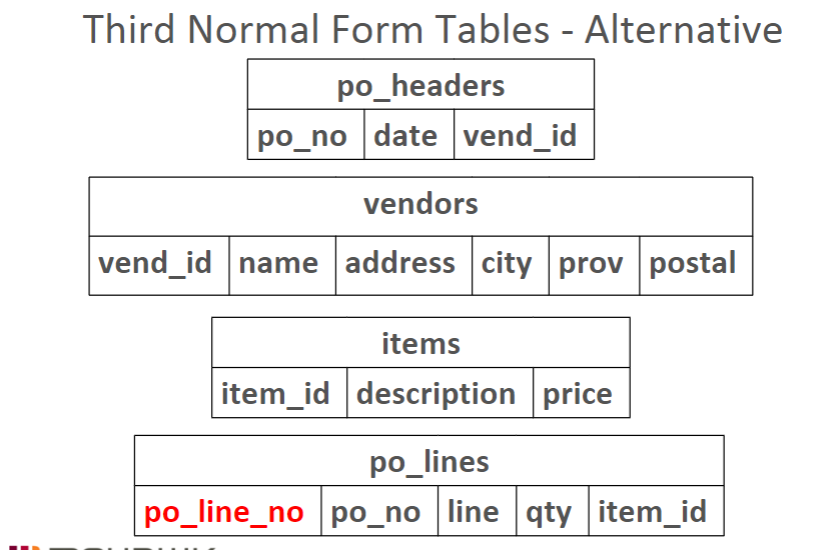
**Composite Keys**

* There are some database designers that do not like composite keys
* An alternative is that any table that would have a composite key, get it’s own unique key
* The column that would’ve been part of the composite key becomes just a foreign key

**Primary Key Types**

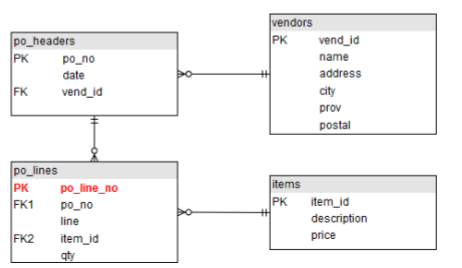
* There are two types of primary keys
  + Natural
  + Surrogate
* A natural key is one that “naturally” occurs in the data, like nursing unit id in CHDB
* A surrogate key does not naturally occur in the data and is usually a meaningless but

unique number (MBUN)



**ERD**

* The po\_line table doesn’t have a composite key
* po\_line\_no is a surrogate key
* In fact, all of the primary keys in this ERD are surrogate



**OLTP**

* All examples considered thus far have been geared to Online Transaction Processing (OLTP)
* Most common
* Normalization rules promote unique (non redundant) data

**Normalized Data**

* Due to all of the joining required to return meaningful results, normalized data is complex to

work with

* Derived columns cannot be indexed, therefore normalized data can be slow to work with
* For the sake of data integrity, in OLTP scenarios, these are tradeoffs we are willing to accept

**OLAP and BI**

* OLTP is not the only use for data
* Why would data be collected, if it wasn’t going to be analyzed too?
* Online Analytical Processing (OLAP) is one of the other significant database uses
* Business Intelligence (BI) tools work with OLAP data to provide end users with the ability

to analyze and report on the data in whatever way makes sense to the particular business

**OLAP**

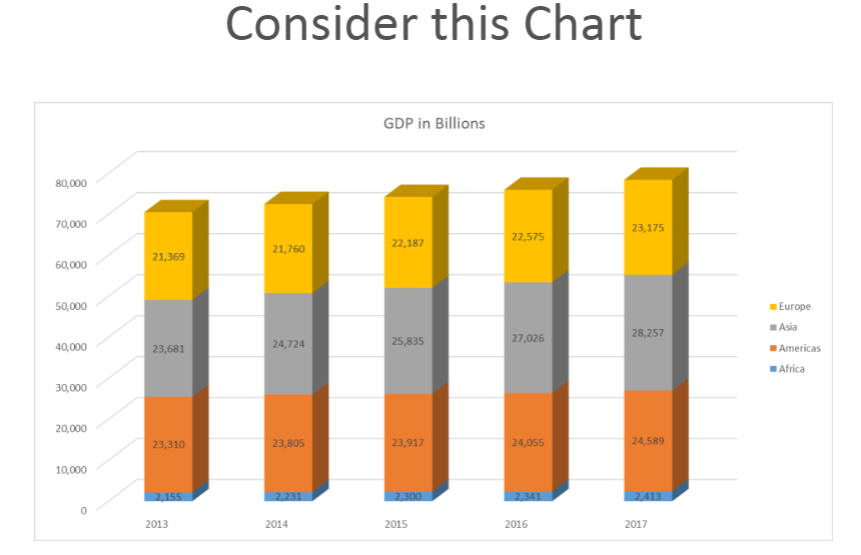
* OLAP emphasizes speed and simplicity of access over data integrity
* In fact, OLAP databases are explicitly de-normalized for this purpose
* Data is also pre-summarized according to expected queries
* Also known as Big Data or a Data Warehouse

**BI Tools**

* Data in an OLAP database is often conceptualized around the concept of a cube, where one of the

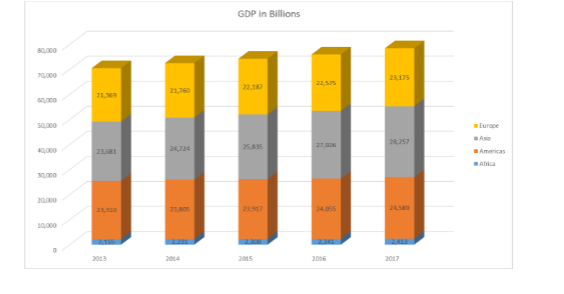
dimensions is time

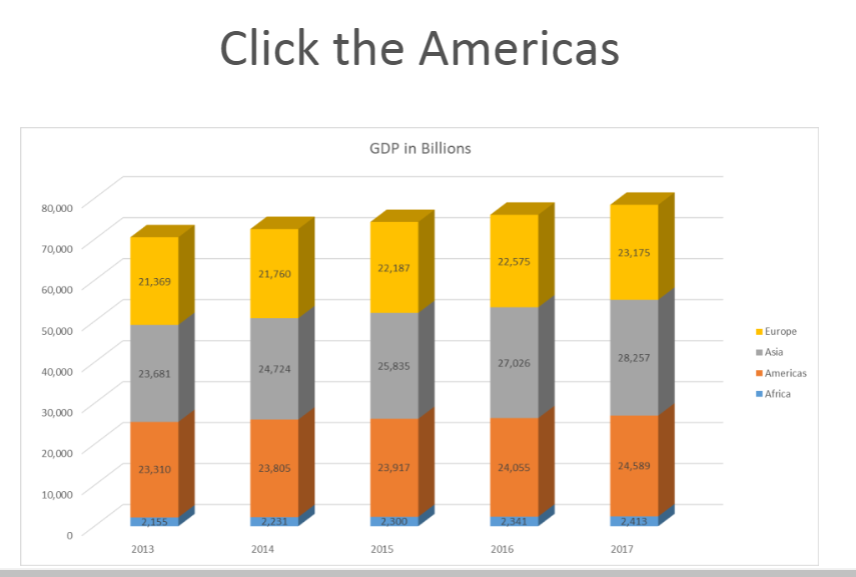
* BI tools allow slicing, dicing, drilling down, etc.

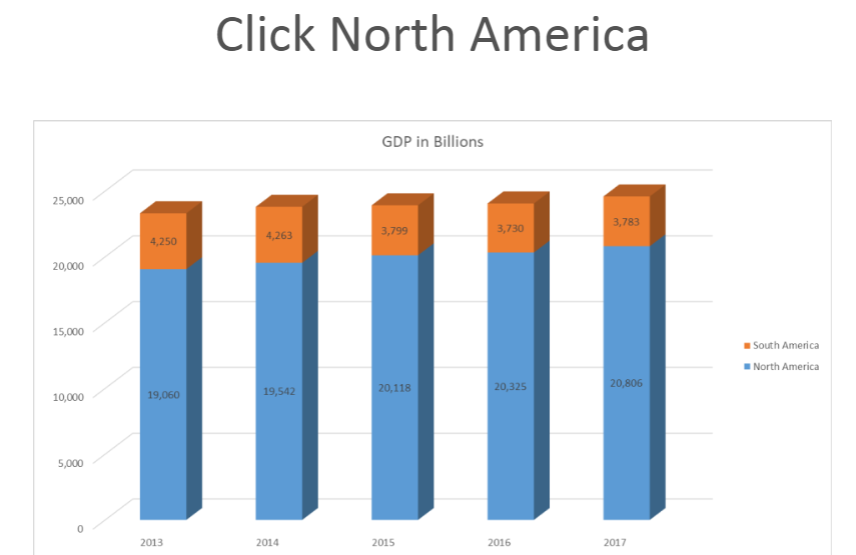


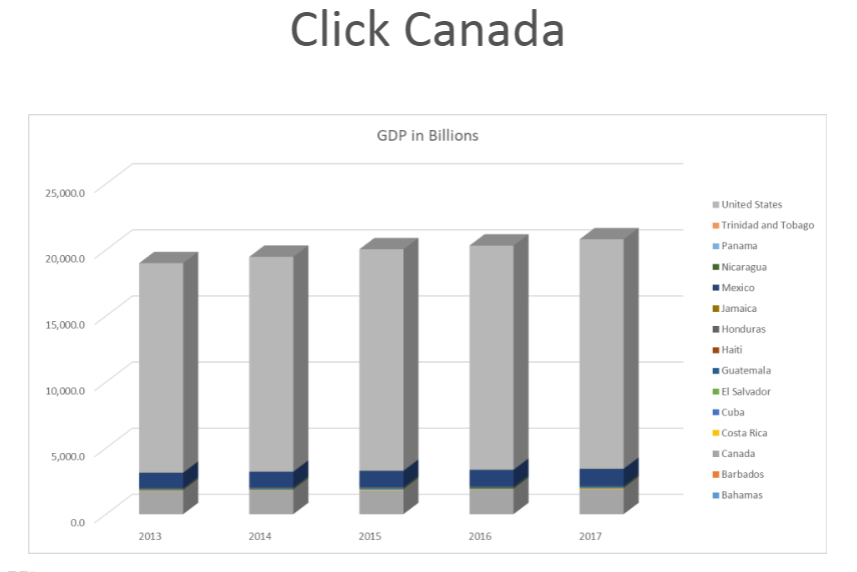
BI Simulation

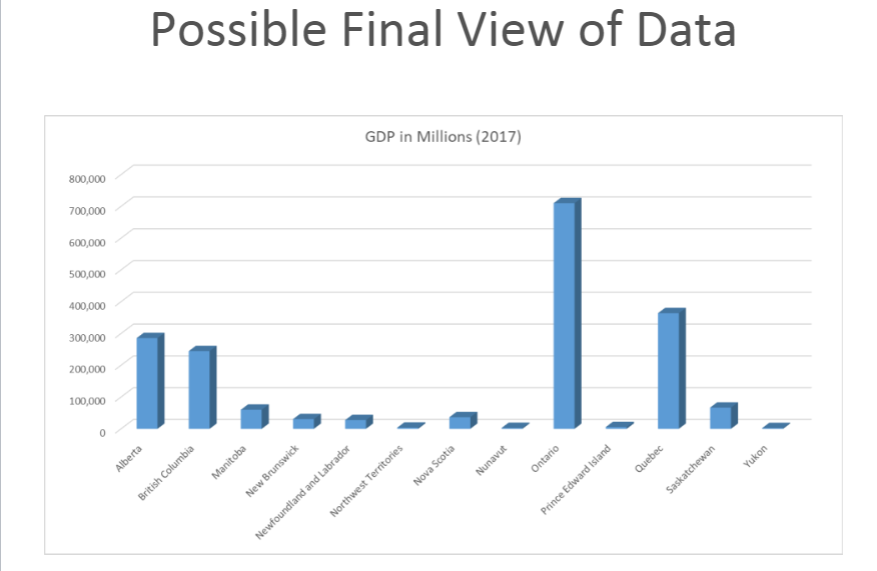
* In a real BI tool, you could click any element and “drill-down” to see more detail
* The following slides will present a simulation











**Gapminder**

* Gapminder is a Swedish non-profit venture
* It promotes sustainable global development
* It provides an excellent BI tool to analyze a number of statistics
* https://www.gapminder.org/tools/

**Data Warehouse**

* Offloading data from the OLTP to the OLAP database allows each database to be

optimized for its particular task

* This separation ensures that complex, heavy inquiries into the OLAP database don’t cause

slow downs in the “production” database

**Summary**

* Normalizing data to 3NF is almost always adequate for “real world” processing
* OLAP de-normalizes and summarizes data for quick and easy analysis