**Lecture 10 -Normalization – Part 1**

**Introduction**

* We want to develop a conceptual model that represents the entities of the enterprise and the relationships between them
* This model is independent of any DBMS package, hardware or the physical model
* Later on, it will be mapped to an internal model that complies with the chosen DBMS

**Relational Concepts For The Design Process**

* We use the normalization process to group data elements into tables that represent entities

**Normalization**

• Normalization is the process of organizing data to minimize redundancy

• Normalized data doesn’t have storage anomalies

• The organizing of data is done by applying a series of rules to a data model

**Normal Forms**

* 0 Normal Form (0NF): un-normalized model before the normalization rules have been applied
* First Normal Form (1NF)
* Second Normal Form (2NF)
* Third Normal Form (3NF)
* Boyce Codd Normal Form (BCNF)
* Fourth Normal Form (4NF)
* Fifth Normal Form (5NF)
* Domain Key Normal Form (DKNF)
* Each previous level must be applied before proceeding to the next

**Normalization**

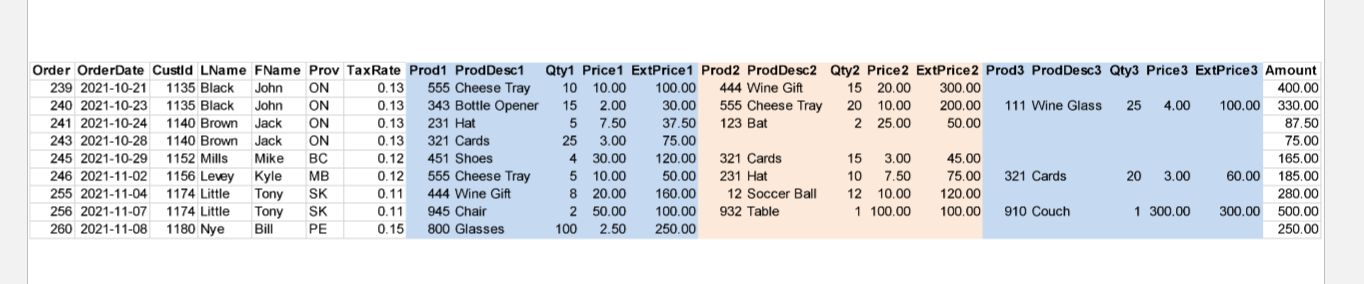
* While normalization makes databases more efficient to maintain, they can also make them more complex because data is separated into many different tables
* This will require most queries to use table joins

**First Normal Form (1NF)**

* Reduce entities to first normal form (1NF) by removing repeating or multi-valued attributes to another, child entity
* To simplify, you cannot have multiple fields to capture multiple values
* Most common ‘sins’ against the first normal form are the lack of a primary key and the use of repeating columns
* Also, remove calculated fields

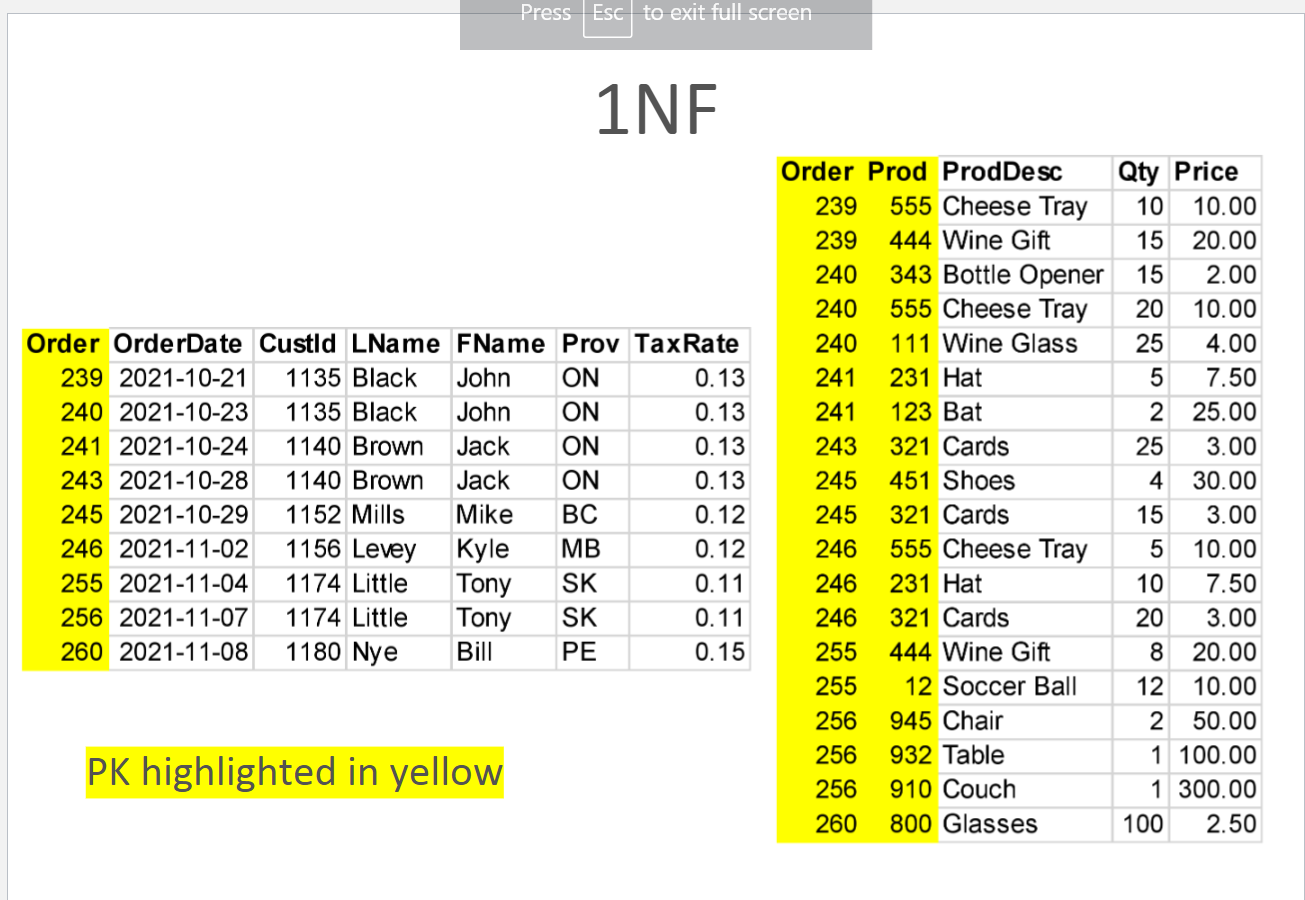
**0NF (un-normalized)**

* No rules have been applied to this model
* Note the repeating product columns
* This does not allow for more products to be added and creates cells with null values
* Also note the calculated fields: ExtPrice1 = Qty1 \* Price1, Amount = ExtPrice1 + ExtPrice2 + ExtPrice3



**1NF**

* Two tables have been created: order and product
* They are linked by the Order field
* This eliminates the repeating fields in a record
* Calculated values ExtPrice1, ExtPrice2, ExtPrice3 and Amount have been discarded



**1NF**

* Here knowledge of the Product # alone provides the product description and price without the Order#
* We must create another table for this to conform to 2NF

**2NF**

* Another table has been created: order\_line
* It is linked to the product table with the common field product
* Therefore each product description is entered once in the database rather than each time someone purchases the item
* The combination of Order# and Product# are needed to identify quantity

**Getting Smaller**

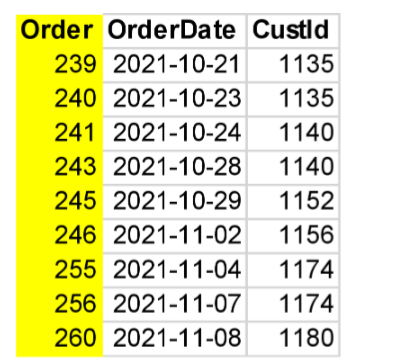
* The order table that started with 23 fields has now shrunk considerably
* 3NF will reduce this further by removing the redundant names, provinces and tax rates

**Third Normal Form (3NF)**

* Third normal form is violated when a non-key field is a fact about another non-key field
* Reduce second normal form entities to third normal form (3NF) by removing attributes that are dependent on other attributes

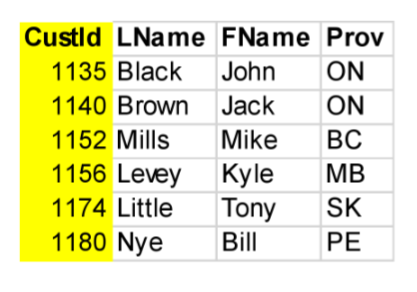
**3NF**

* By creating another two tables our order table is down to 3 fields
* We’ve eliminated the name fields by linking the CustID to a separate customer table, and likewise for the tax rate using the province abbreviation

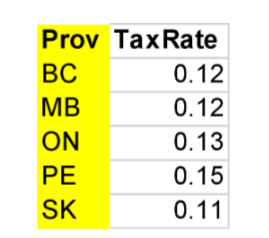


**3NF**

* This removes the redundancy of the customer name and province each time an order is placed



* Here the redundancy of the tax rate is removed by creating a table with
* province tax information



**Order Normalization**

• The original un-normalized Order table became 5 normalized tables

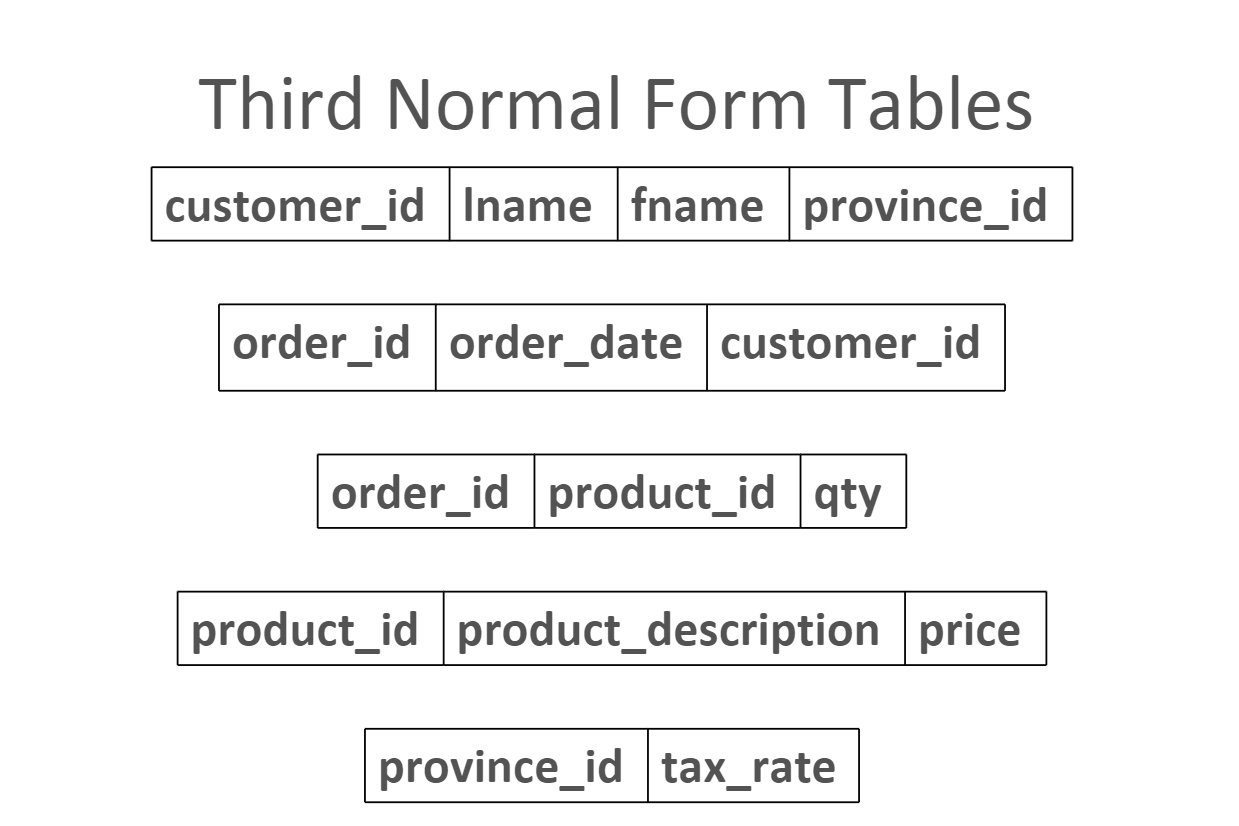
– customers

– orders

– order\_lines

– products

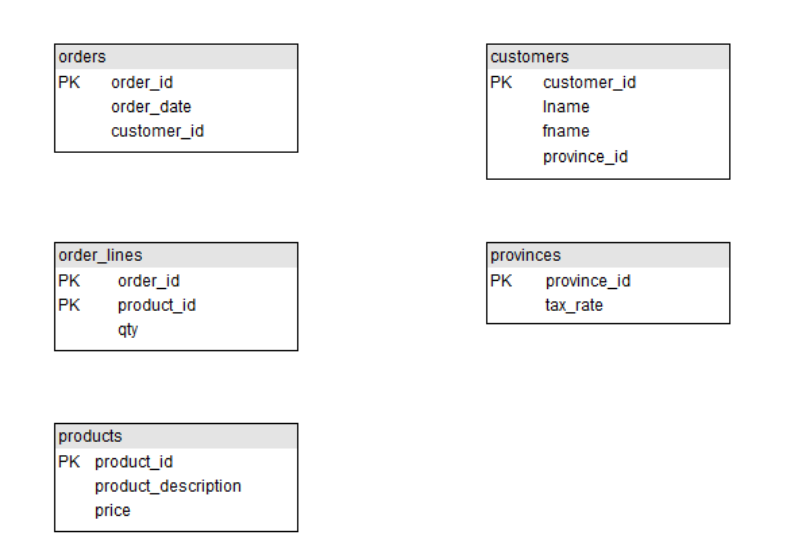
– provinces

****

**Third Normal Form Tables**

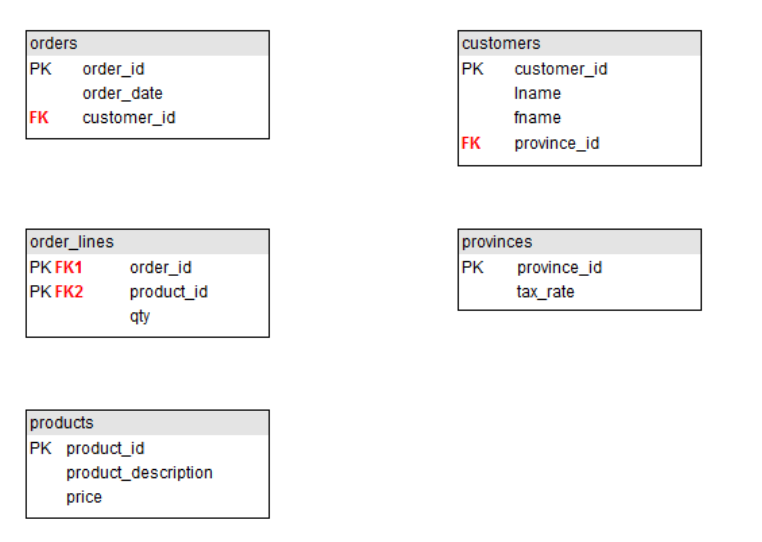
Creating the ERD

• Draw the entities with their primary keys



**Creating the ERD**

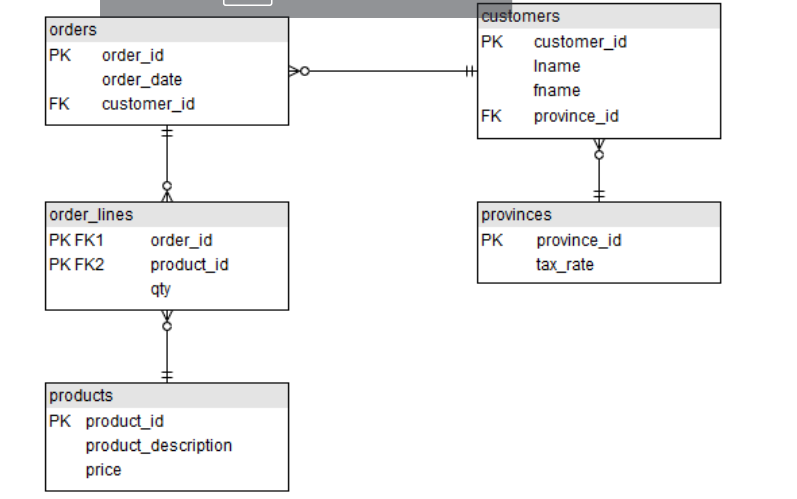
• Examine each primary key to see if it appears in another table, if so it is a foreign key in the other table



**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



**Normalization Good News/Bad News**

* The bad news is that this increases processing complexity
  + In order to get any useful information out of the database, most queries will need to join 2 or more tables
* The good news is that the data is organized and unique
  + Each fact is only stored in one place

**Normalization Quick Summary**

* 1NF - Eliminate Repeating Groups And Derived Attributes
  + Make a separate table for each set of related attributes, and give each table a primary key
* 2NF - Eliminate Redundant Data
  + If an attribute depends on only part of a composite key, remove it to a separate table
* 3NF - Eliminate Columns Not Dependent On Key
  + If attributes do not contribute to a description of the key, remove them to a separate table

**Summary**

• Normalization is the process of organizing data to minimize redundancy

• Normalized data doesn’t have storage anomalies

• 1NF - Eliminate Repeating Groups And Derived Attributes

• 2NF - Eliminate Redundant Data

• 3NF - Eliminate Columns Not Dependent On Key