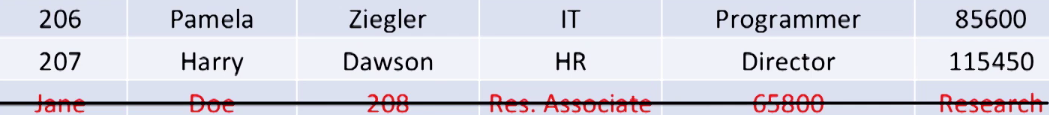
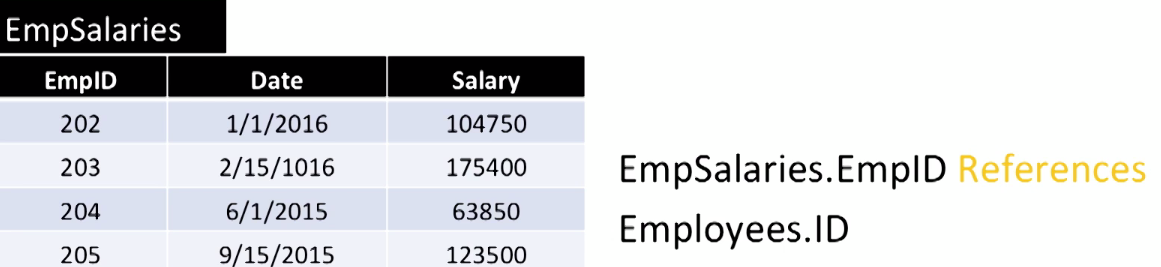
**I. RELATIONAL DATA MODEL**

* Primary data structure for a **relational model** is a **table**, which is very similar in structure to a Python DataFrame (which can be used to load **relations** (relational tables) into Python)
* provides a way to describe unique relationships between entities and data records
* Pandas DataFrames implement some features from relational models to make it easier to work w/ RDBs
* Tables represent a set of **relational tuples** as *rows* (records)
* A table of 6 rows is a **set** (collection of *distinct* elements of the same type) of 6 tuples
* Many relational systems allows duplicate tuples, but mechanisms are provided to prevent these is a user chooses so
* Also cannot add a tuple w/ info in the wrong order
* 
* 
* The header (1st row) is part of the **schema** of a table = columns in a Pandas DataFrame
* Schemas can also specify keys (ID as **primary key =** uniquely ID a record/tuple and uniquely know the other 5 attributes for that employee)
* Tables w/ primary keys prevent duplicate records via a **uniqueness constraint**
* **Foreign Keys** refer to primary keys



* **“References” =** values in the EmpSalaries.EmpID column can only exist if the same values appear in Employees.ID (exists in *table being referenced* = **parent table**)
* **Natural Join** = Inner Join
* Joins are one of the most expensive (in terms of time and space) consuming operations, especially w/ very large tables
* For Data Science involving big data, it’s very important to choose a suitable RDBMS to make this operation efficient
* **Data retrieval** = the way you specify how to get desired data out of a relational data store, + the internal processing that occurs w/in the data management system to compute/evaluate that specified retrieval request.
* Ex: Database Schemas:
* Bars(name, addr, license) Beers(name, manf) Sells(bar, beer, price)
* Sells records which bar sells which beer at what price (differences in establishment costs)