Dietrich Mosel
Prof. Labouseur
Database Mgmt.
9/11/17

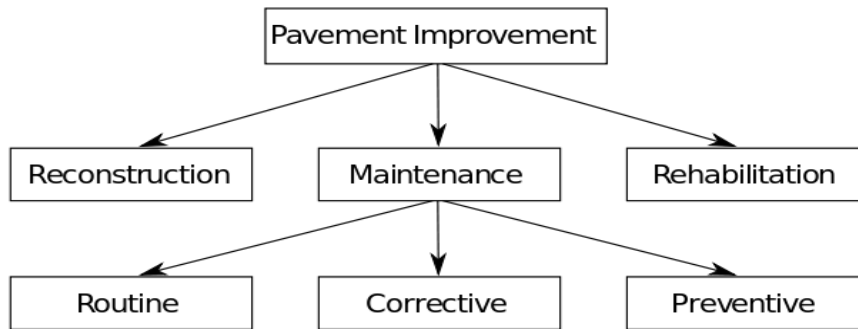<center>Lab 1: PostgreSQL</center>

Data vs. Information:

Online television streaming is becoming more and more popular with the younger generations as less are paying for conventional cable subscriptions and opting for the Netflix deal instead. In reference to databases, Netflix utilizes a highly-specialized program known as Cassandra that can turn vast amounts of useless data into information that can generate huge profits for the streaming giant. According to the Apache website, Cassandra can give you scalability and high availability without compromising performance. It is perfect for scaling horizontally because it can add more servers without rebooting, avoiding any roadblocks produced by using vertical scalability. Services like Netflix and Hulu can utilize databases to generate lists of shows or movies to watch, track a customer's preferences and from that, recommend new shows based on viewing patterns. Knowing that I watch Silicon Valley for example is pretty useless, but taking that data and then recommending that I watch VEEP might keep me tied to my tv or computer using Netflix more frequently. Data, without context and organization, is not useful as proven in my last example. However, the utilization of that data given some background can significantly help companies better reach their customer base. Back to the Netflix example, receiving a list of people who watched a show is just that… it's a list and no more. But when that data is taken and analyzed or organized, you can figure out shows to recommend that will keep them watching more and subscribing for longer periods of time.
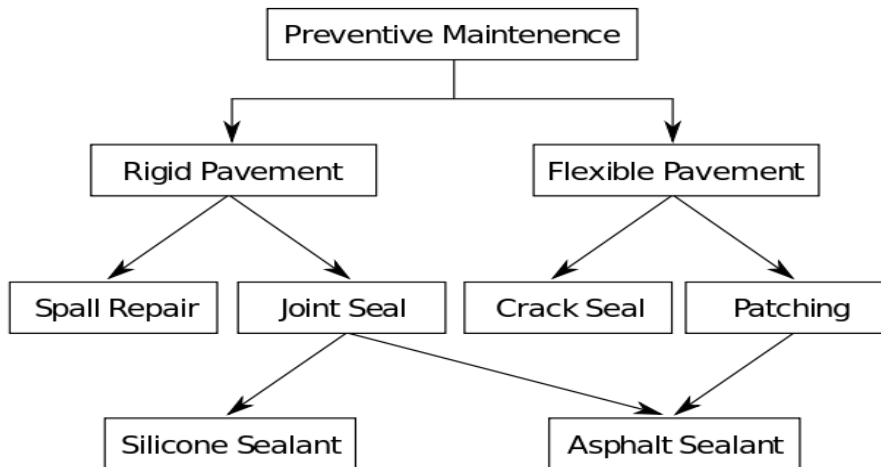
Data Models:

The hierarchical database model was first developed by IBM in the 1960s that would organize data into a tree structure. This model hinders the user as each child record can only have one parent but each parent can have one or more child records. Meaning, in order to properly interpret the data, the user must follow the tree starting from the root node. It is a very rigid model that is not quite user friendly. The network model on the other hand, is a more flexible way of expressing relationships between objects. The schema which is the model's layout, can be viewed as a graph which is not as restrictive as the hierarchical model. This model allows each record to have a multitude of parent and child connections which is beneficial to the user in handling non-hierarchical relationships. Both models have shortcomings in regard to the relational model which can support one to one, one to many, or many to many relationships. It is the most flexible out of the three and can yield results based on one search request. For instance, you have two tables (one of which corresponds to the other in reference to an activity code let's say) that contain different sets of information. You can create a third table with activity codes that will produce further information pooled from the other table.

Hierarchical Model

```
                    ┌─────────────────────┐
                    │ Pavement Improvement │
                    └─────────────────────┘
           ┌───────────────┼───────────────┐
┌────────────────┐  ┌─────────────┐  ┌────────────────┐
│ Reconstruction │  │ Maintenance │  │ Rehabilitation │
└────────────────┘  └─────────────┘  └────────────────┘
              ┌──────────┼──────────┐
       ┌─────────┐  ┌───────────┐  ┌────────────┐
       │ Routine │  │ Corrective │  │ Preventive │
       └─────────┘  └───────────┘  └────────────┘
```

Network Model

```
                 ┌───────────────────────┐
                 │ Preventive Maintenence │
                 └───────────────────────┘
           ┌──────────────────────┴──────────────────────┐
    ┌────────────────┐                          ┌──────────────────┐
    │ Rigid Pavement │                          │ Flexible Pavement │
    └────────────────┘                          └──────────────────┘
      ┌──────┴──────┐                            ┌──────┴──────┐
┌──────────────┐ ┌────────────┐         ┌────────────┐ ┌──────────┐
│ Spall Repair │ │ Joint Seal │         │ Crack Seal │ │ Patching │
└──────────────┘ └────────────┘         └────────────┘ └──────────┘
                      └──────┐              ┌────────┘
               ┌──────────────────┐   ┌──────────────────┐
               │ Silicone Sealant │   │ Asphalt Sealant  │
               └──────────────────┘   └──────────────────┘
```

Relational Model

Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

Key = 24

Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495
02/08/01	24	I-66