**ANA650 - Design Project - Part 2**

**Team 1**

|  |
| --- |
| Aditi Bhujbal, Stefan Francisci, Shila KC and Jennifer Knight  Department of Engineering and Computing, National University  ANA 650: Database Design for Analytics |

Professor Trzos

May 21, 2022

**Database Type:**

This is a **transactional or operational database**, as our goal is to store the day-to-day transactions generated by customer acquisition, orders, troubleshooting issues and complaints from customers.

**Main Entity:**

**CUSTOMER** is main entity in our database, as this will store the customer information and is connected to almost all entities directly or indirectly.

**Assumptions:**

1. Each employee will require credentials to log in to ticketing tool which will have different permissions based on the role assigned to the employee and each user will be a member of a queue.
2. Each department will have its own queue to create or modify the tickets.
3. End Date for a ticket will be calculated with the help of TktPrio and TktDur (duration) in **SLA.** We would not store it in table but will be directly displayed in the ticketing tool.
4. A complain will be registered by a customer hence CusID is required attribute in **Complaints** table, however, not all employees will receive a complaint against them, hence EmpNum is not a required attribute in it.
5. Each Product will be supplied by a vendor and each vendor can supply multiple products, hence there is a one-to-many (1:N) relationship between **Vendor** and **Product** entities.
6. An order can be placed for multiple products together and a product can be present in many orders i.e., there is many-to-many (M:N) relationship between **Product** and **Order** entities. Hence an associative entity called **Purchase** is created to implement this M:N relationship.
7. There are different payment methods available now a days, hence a separate entity called **Payment** is created to store payment related details associated with each order.
8. As this is an online business, our products need to be stored at different location for ease and the earliest delivery of products to customers. Hence a separate entity called **WareHouse** is created to store the information about availability of a product at different locations which will be connected to Product entity using ProdCode.