**Assignment - Common Database Designs and Normalization**

Shaun Pritchard

Rasmussen College

QMB4000

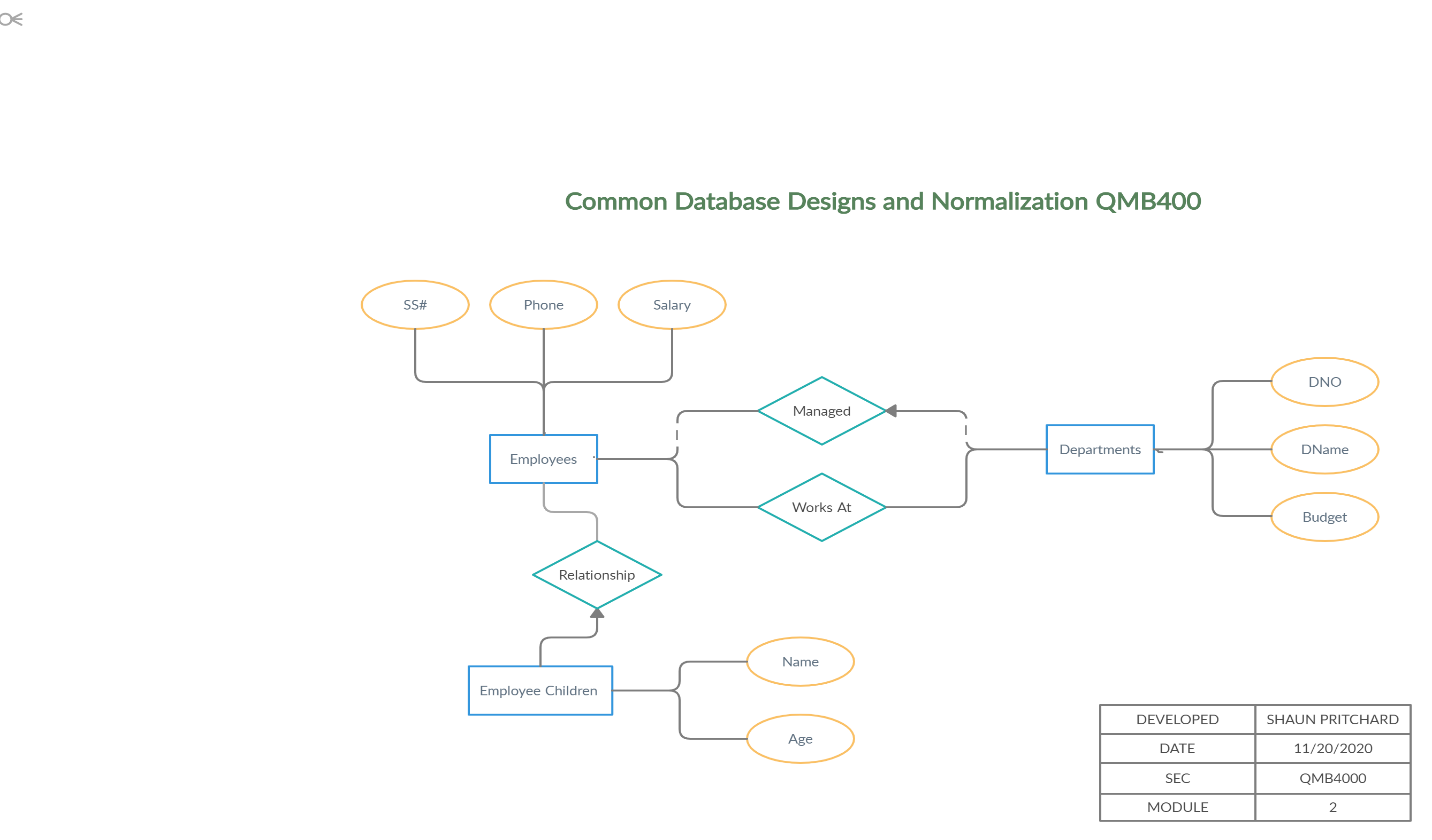
Benjamin Tasker

November 22, 2020

**Assignment - Common Database Designs and Normalization**

**Part 1:**

Based off the mockup customer database, we see that there are direct relationships between the employees and a weak relationship between the employee’s children which are dependent on the employee. While there is essentially ISA relationship between the department based of the employee managing the department and the department the employee works at. This is what I have visualized in the ER diagram below.

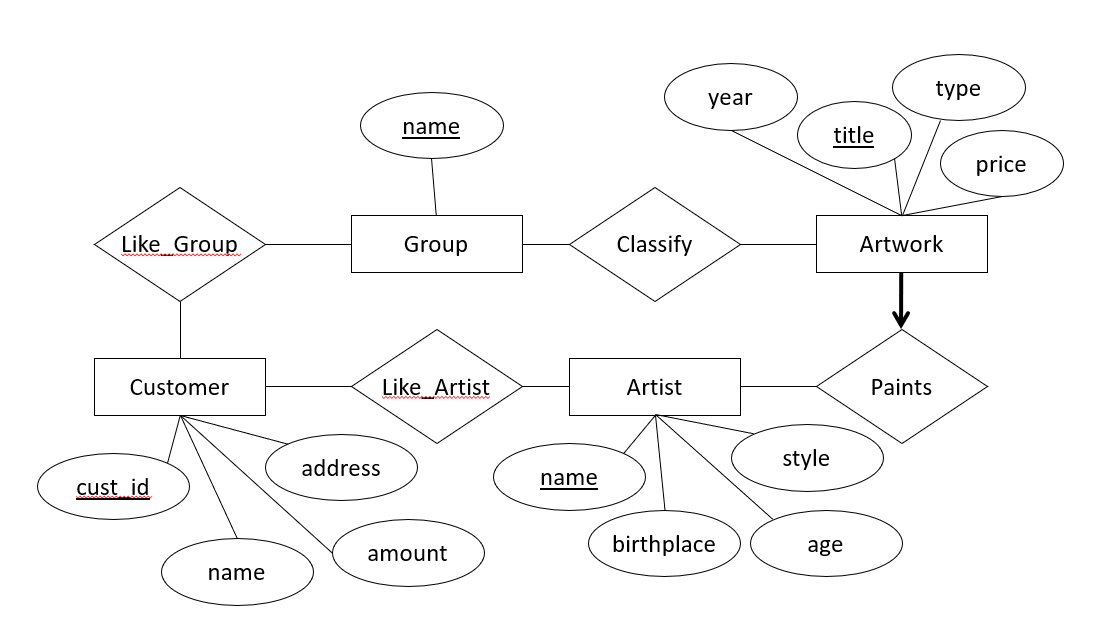


If I were to associate Primary and foreign keys to this model it would look like this.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EMPLOYEE** | **EMPLOYEE CHILD** | **DEPARTMENT** | **EMPLOYEE** | **EMPLOYEE CHILD** | **DEPARTMENT** |
| **PK = EMPLOYEE ID** | **PK = EMPLOYEE CHILD ID** | **PK = DEPARTMENT ID** | **FK = DEPARTMENTS, EMPLOYEE CHILD** | **FK = EMPLOYEE** | **FK = EMPLOYEE** |

**Part 2**

The ER diagram from the assignment shows a relationship between customers and artist in a group. Groups in relationship representing a social media group like relationship based off the relationship of the classification of artwork created by the artist. Some customers can like specific groups of artists based off of those relationship which are in turn based off the type of artwork and artist provides customers can like individual artist from within the relationship of the group. The likes associate the relationship between customers and artist in said group.



**Part 3:**

**What is a primary key and foreign key?**

The primary key consists of one or more columns whose data contained within are used to uniquely identify each row in the table.  You can think of them as an address in the model. A foreign key is a set of one or more columns in a table that refers to the primary key in another table.

**Part 4:**

**What is an ER Diagram, why is it important?**

ER diagrams are used to analyze databases to find relationships between entities of the database modeling and to resolve issues, problems , or logic. An entity relationship diagram describes how entities relate to each other. In simple terms, it is a picture or a framework of your business logic or a certain business process.

Entity Relationship diagram visually shows the various entities or tables and relationship between entities. Such as how two tables are inter-related and their cardinality.

ER diagrams also shows attributes (columns), the keys (primary, secondary, composite, unique, and foreign.) for each table.

**ER diagrams are important due to these advantages:**

* Database design is well documented.
* Database can be visualized and easy to reference.
* Visualized data bases through ER diagrams help solve issues in the database model.
* We could know which tables are interlinked to which ones including their cardinality.
* Many to many relationships needs a new relationship table which is clearly seen in ER diagrams.
* One to many relationship relations will have FK on the many side.
* They help maintain database schema.

**Part 5:**

**Define Normalization in your own words?**

Database normalization is a process used to organize a database into tables and columns. Normalized tables have a schema and relationships between sets, tables, rows, columns, of data to get accurate analyzation from the data. Taking it a step further normalizing a database means to preprocess data to ensure that is free of errors, duplicated, corrupt, and illogical data to work with. Basically, means organized data that is clean and ready to process

**Part 6:**

**How does performance improve with a Normalized Database?**

Database normalization is, at its simplest, a way to minimize data redundancy and errors. To achieve that, certain forms of normalization exist.

Normalization can be summarized as:

* No repeating groups in single tables.
* Separate tables for related information.
* All items in a table related to the primary key.

Normalization can ensure that data is processed correctly, faster, and analyzed , stored ,and maintained in the same manner. Redundancy uses bandwidth and other resources which all cost money to operate. Normalizing data makes implementing the data more cost effective while lowering the amount of resources being used to facilitate it.

**Part 7:**

**Why may a company use a non-normalized database?**

There are many reasons why a company would not use normalize data. Particularly because they did not need to have normal form and relationships between the data that the company collects. Some companies collect a lot of unstructured data or repetitive unstructured data. Realistically it really depends on the importance of the data for specific business metrics.

Transactional data for instance should be structured and should be normalized in a database because if we were dealing with money or finances that that would be very important not to have redundancy that allows a bank to double charge a client. If we are dealing with repetitive notifications for instance every time a user spent money from their account. a specific user actions in a banking system would be triggered that has no correlation to banks business data such as if they wanted to see which customers saved or spent more money at a particular time of the year. Typically, that type of data would not be normalized because In most cases, it's not going to be analyzed and it is redundant.

If we are talking about non-normal data in regard to denormalization then that is very different proclamation of what the answer for this question is? For transactions like purchases inventory maintenance personal data we would not want a lot of issues with it. It would definitely need to be normalized, but when we want to start answering questions like which neighborhoods do customers live in that purchase more items, we can collect data that needs to be denormalized. Denormalized data is the data that is being extracted from large collections of normalized tables and has been organized and aggregated into fewer tables without regard to such things as redundancy unlike the process of normalization.