The Track of Manufacturing Company Orders

Leader: Tianyu Bao tianyuba@buffalo.edu

Member: Xiaojie Yang xiaojiey@buffalo.edu

Problem Statement

Systematically organize customer orders and visualize the order progress.

There are many close and complicated relations between each table. These relations cannot be directly described by Excel sheet. And when the data size becomes larger, it will be time consuming to search for some specific entries from Excel sheets. There are different types of users, and databases can be accessed by multiple users and multiple applications at the same time.

Target user: Customers and staff who are related to manufacturing, like company manager, labor worker (group leader), and driver etc.

Administrator: company manager (higher leader) or database manager (a special position aiming to manage data)

Process

Customers can place an order (which can include several kinds of product), and then the manager will determine if the order is doable or not. Once the order is accepted, the producing labor will be divided to workers (May formed by group). Based on their work rate, generate the estimate date for shipment. After all parts of the order is done, package then load onto transportation. All the prices for product, packaging, and transportation will calculate automatically. The order will be finished after the customer receives the product and clicks the confirmation of receipt button.

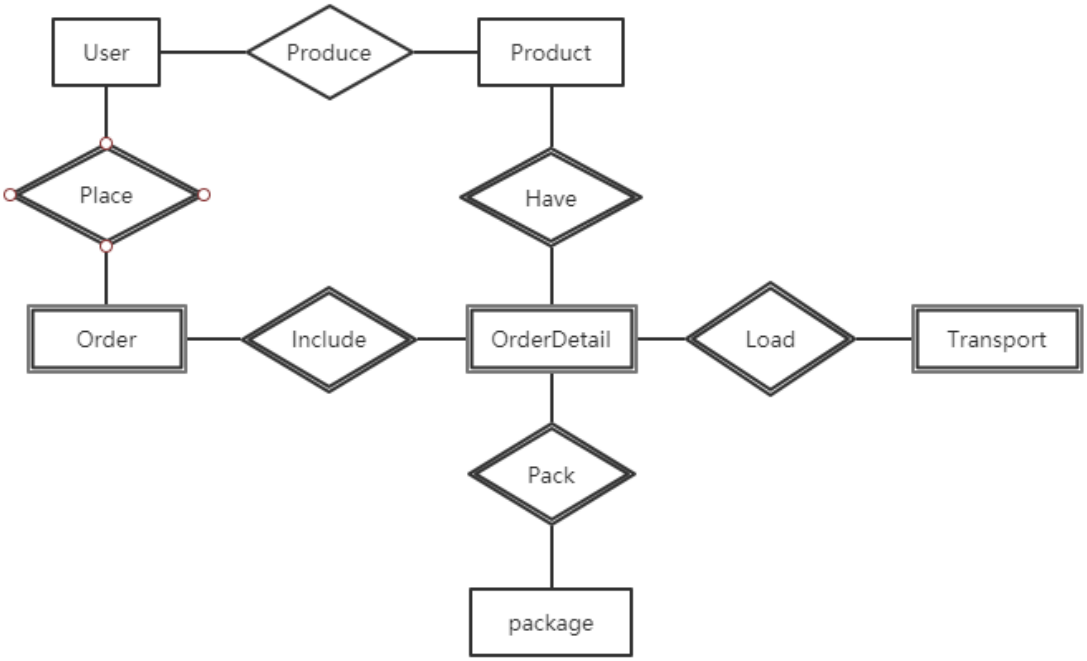
Managers with higher authority will have the ability to access more information, and modify data, for example, update the price for a sale, or hire more workers then add them into the database.

Advantage

The process of producing flow lines is totally eyeable and under control. For customers, they can keep eyes on their order status. Is it shipped? When can I get them? Furthermore, Each customer can specialize their order in detail, choosing a gift packaging, or add some spark chips on your hat. They can do any combination that the

factory offered. For company managers, it is much easier to make the production efficient, do the best in reducing the probability of delay, and give a better experience to your customer.

A simple E-R model (without attributes list) for our project



Data Resource

Considering there is not a real-world dataset that is applicable for the prior database we designed. We will generate a set of "fake" data that suits my database. The estimated size of data will be around 100 rows each table. I will implement a web app to make our database visible and manageable.

Database Creation

From the E-R diagram above, we finally keeped all of the strong entities and weak entities, then converted them into schemas. For the relationships between entities, Considering it is a one-to-many relationship, we used a foreign key constraint in the "Order" schema for the "Place" relationship between "User" and "Order", as well as for the relationships between "Order" and "OrderDetail", and between "OrderDetail" and "Package". Then we used a schema named "WorkRecord" to represent the relationship "Produce", "Pack", and "Load". Those three relationships are many-to-many while those work may assign to many work groups to complete parallelly.

The script (MYSQL script) for creating the database schema until milestone 1 can be found in the following github link:
https://github.com/bao-bao/CSE-560/blob/master/create.sql

Constraints

1. **Naming Pattern:**

   Table name and attribute name are following the snake case. Table names start with "order_track", while the attribute name starts with the initial of their table (if the same initial occurs, add more characters until not same).

2. **Primary Key:**

   Each table has a unique integer id as their primary key, set "auto_increment" to them for uniqueness. It is cheap (both for storage and indexing computing) and easy to use as a foreign key in most cases.

3. **Foreign Key:**

   There are seven foreign keys in our database. All the foreign keys referenced to the "User" table are referenced on "name" (an unique attribute in the "User" table), others are referenced on the primary key of the target table. All the foreign keys have constraints on deleting and updating.

   Considering the product and package information should not be changed once the order is placed, we set the "on update" constraint as "no action" for the foreign key of "OrderDetail" table to "Product" and "Package" table. Other "on update" constraints are "cascade", while "on delete" constraints are "restrict".

4. **Default Value:**

   Some numerical attributes will have a default value of "0", like "count", and "price", etc. Also, "start_time" and "finish_time" in the "WorkRecord" table have a default value of "0000-00-00 00:00:00", which is a zero value for timestamp type variables.

5. **Null or Not Null:**

   All the attributes have the "not null" constraint, even they could be a "null" value. The reasons for this are, first, reduce space; second, to make the indexing in query faster.

6. **Other:**

   Most tables have two attributes of "change time" and "update time" for tracking the updating and inserting operation in the database. They will have a default value of current time and an "on update" trigger. That will give the manager a reference if the database needs to rollback or other recovery operation.