Basics of database systems

Project – Database design

Lappeenranta-Lahti University of Technology LUT
Software Engineering

Basics of database systems
Spring 2022

TABLE OF CONTENTS

TABLE OF CONTENTS.. 3

1 DEFINITION 4

2 MODELING 5

2.1 Concept model..5

2.2 Relational model 6

3 DATABASE IMPLEMENTATION 7

1 DEFINITION

This database is created for a company that has a webstore and needs a system to control all current products and orders. The database contains information about the company, its warehouses, products, orders, payments, and customers. All information has its own id that is also a foreign key for the classes that need them. The products are located in one of the company’s warehouses and every order can have one product. The order id is used for users and payments to recognize the correct orders. Other important information that the classes have are name and price info. For this project, the database is made simple, and it is designed just for one company. It would be easy to enlarge this project for multiple companies to use.

Because the database is created for the company, it is the only one who can use the program. With the program, the company can change already existing orders, list all their products from warehouses at the same time or separately, search due dates of certain orders and add more products to the database. The program helps the company to keep up with the orders and products in different warehouses. It can also be used as a customer service task because it finds the due dates fast by the id of the order.

To get a working program with the database, the program should have at least these queries implemented: (1) Print the existing products from one or every warehouse (2) Print every order detail so you can see who ordered and what. (3) Search the due date by order id. (4) Change the ordered product to another for the specific order. (5) Add more products to the database.

2 MODELING

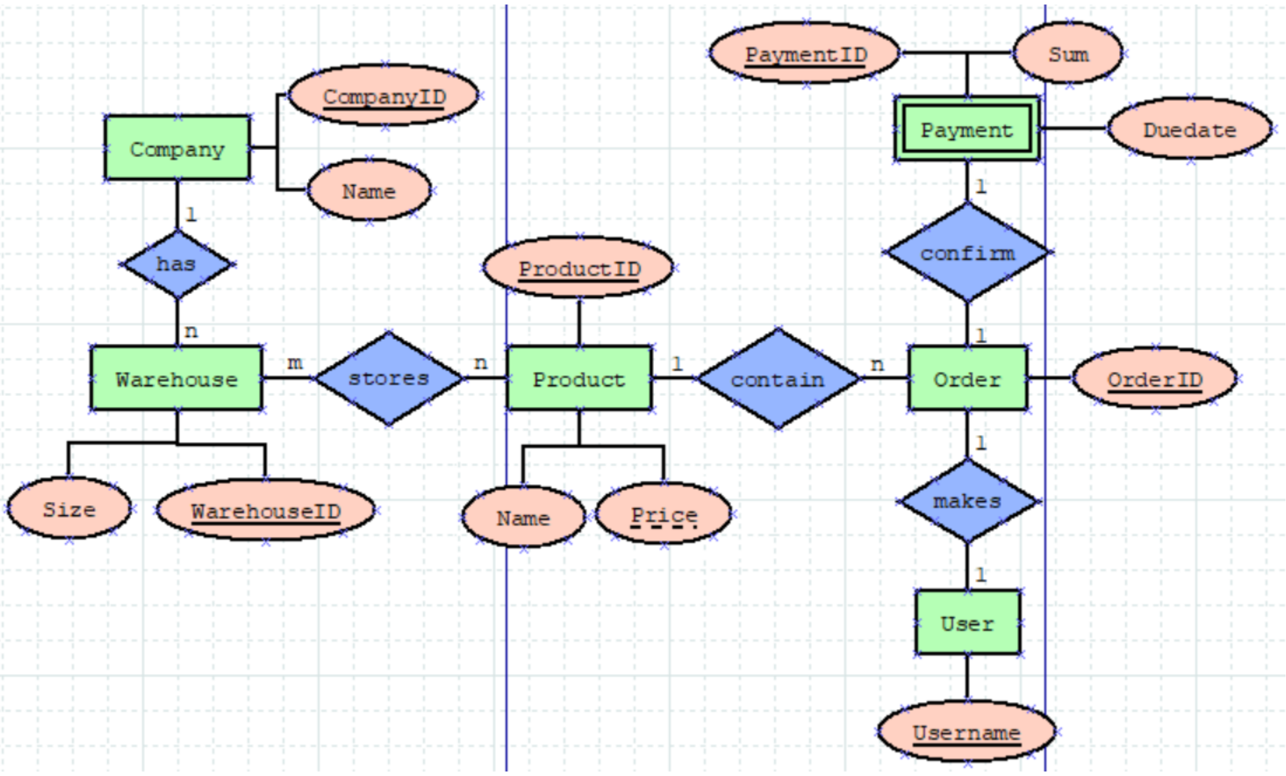
2.1 Concept model

This database is modeled with the ER model (see picture 1). At the models and in the source code the customer is referred to the user.

The model implements all the entities, their relationships with each other, and their attributes. At this database, the entities are a company, warehouse, product, order, payment, and user. From these entities, the payment is called a weak entity because it cannot exist without the order.

Every entity has a relationship with at least one other entity. The company can have multiple warehouses, so their relationship is one-to-many and called “has”-relationship in the model. Warehouses although can store many different products of the webstore, when also products can be stored at different warehouses. Their relationship is many-to-many and it is called “store”-relationship in the model. The relationship between the product and order could also be implemented as a many-to-many relationship, but to make this database simpler it is now one-to-many. So, the product can be in many orders, and the order, there can be only one product. This relationship is called the “contain”-relationship in the model. The order is made by the user and their one-to-one relationship is called “makes”. The last relationship is between order and payment. This is also a one-to-one relationship, and it is called “confirm”.

The attributes are connected to entities that they are related to. From these, the underlined ones are key attributes. They define all the data to differ from each other. There is also one dotted line, and it implements that the price of the products can change. For example, in sales.

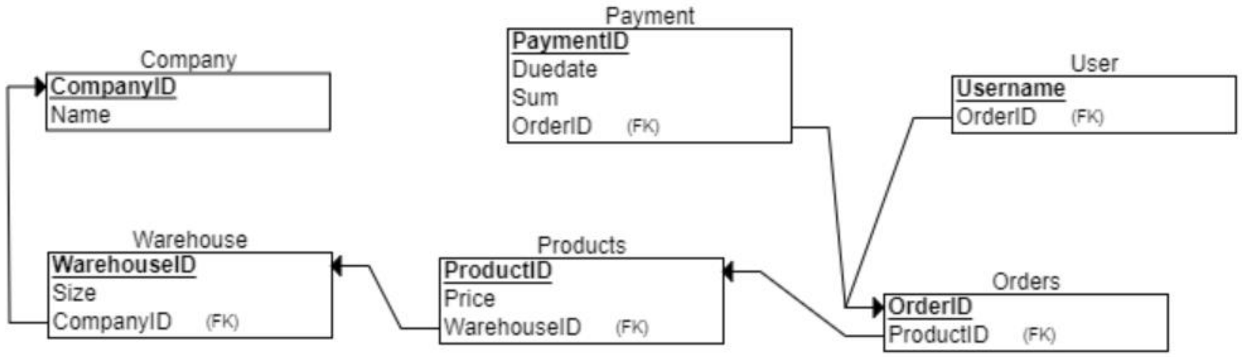


Picture 1 ER model

2.2 Relational model

There were not many changes in transforming the ER model into a relational model. The entities are now relations, their key attributes are primary keys, and the attributes are secondary keys. In addition to these, there must be foreign keys in the relations. These come from the rules for the relationships with the entities. The primary keys are underlined, and the foreign keys are marked as "(FK)" in the model (see picture 2).

There are only a few text constraints in this model and all of them are defined with varchar to be 20 characters long. These are the company's name, products name, payments, due dates, and user's name.



Picture 2 Relational model

3 DATABASE IMPLEMENTATION

There did not occur any bigger problems with the transformation of the ER model to the relational model. The main thing was to understand which data should be as a foreign key in another relation.

I started the implementation of the project by making a Python program without any SQL queries. Then one by one I started to build up the queries and test them along the way. I used extra print commands between the actions to see the change of some queries and that way I found the right formats.