Basics of database systems

Project – Sales Management System

Lappeenranta-Lahti University of Technology LUT
Software Engineering

Basics of database systems
Spring 2022

TABLE OF CONTENTS

1 DEFINITION

The project "Sales Management System" is intended for managing a sale details of a small export firm. Due to the small scale of this business, the database should be compact and lightweight. It enables users to efficiently track relations between orders, items and customers. Important information includes items, customers, and shipments. The database also stores information on customers' delivery address and contact details. If a new customer is ordering a product, the customer's information (name, age, address, phone number and email) is entered into the database. Then the customer could choose to order a variety of items and the order is finalized by recording the details of the order to a join table. In case the items requested are not available, then they should be added to the database.

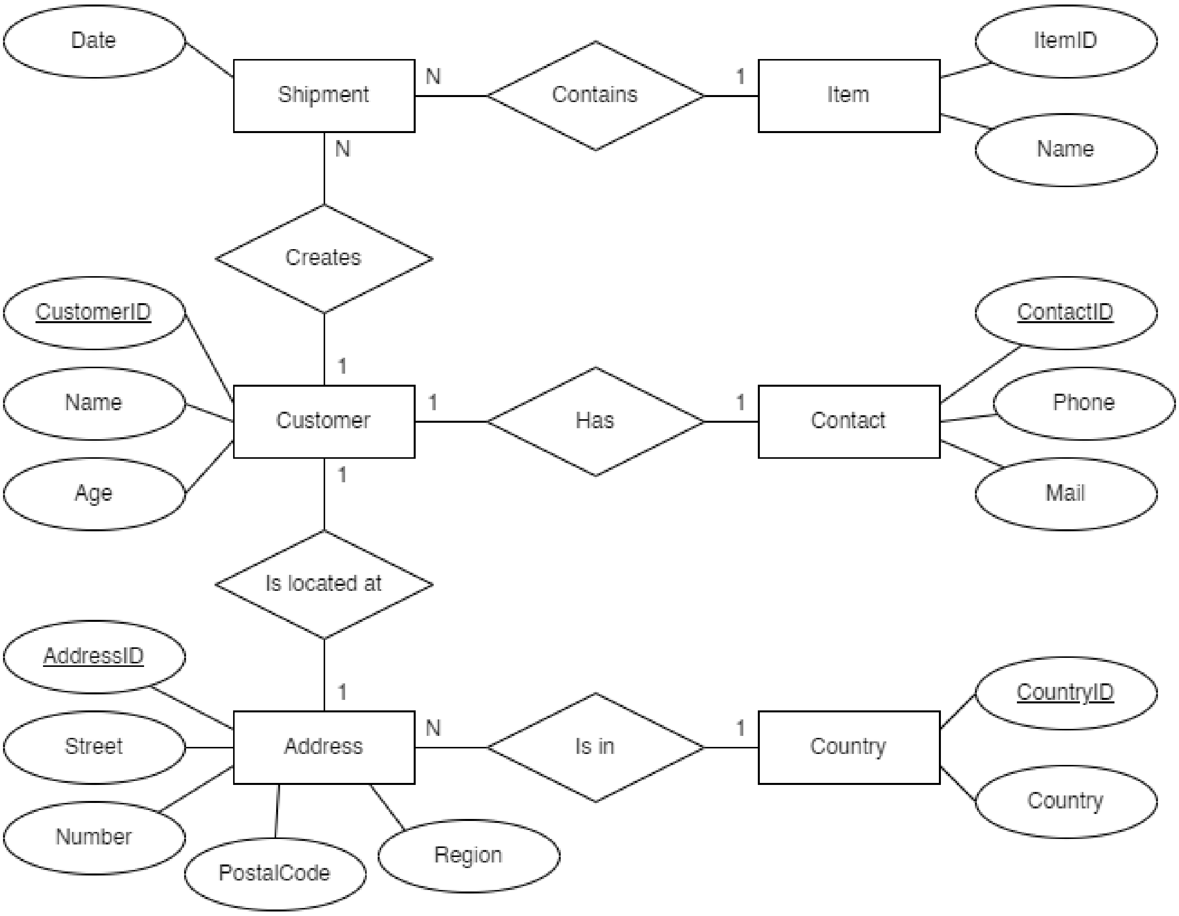
The database includes one many-to-many relationship, which connects the Customer table to the Item table. To avoid violating the rule "No multivalued attributes" of the first normal form, this relationship has been broken down into two one-to-many relationships, connecting Customer table to the join table Shipment and the join table Shipment to the Item table.

Possible database users: Sales employee, managers, delivery/postal services. Sales employees enter the information mentioned above to the database. Managers check the information and make necessary changes. Postmen or couriers could see the contact details and address, but could not add, delete or change any data. Some queries are implemented as follow:

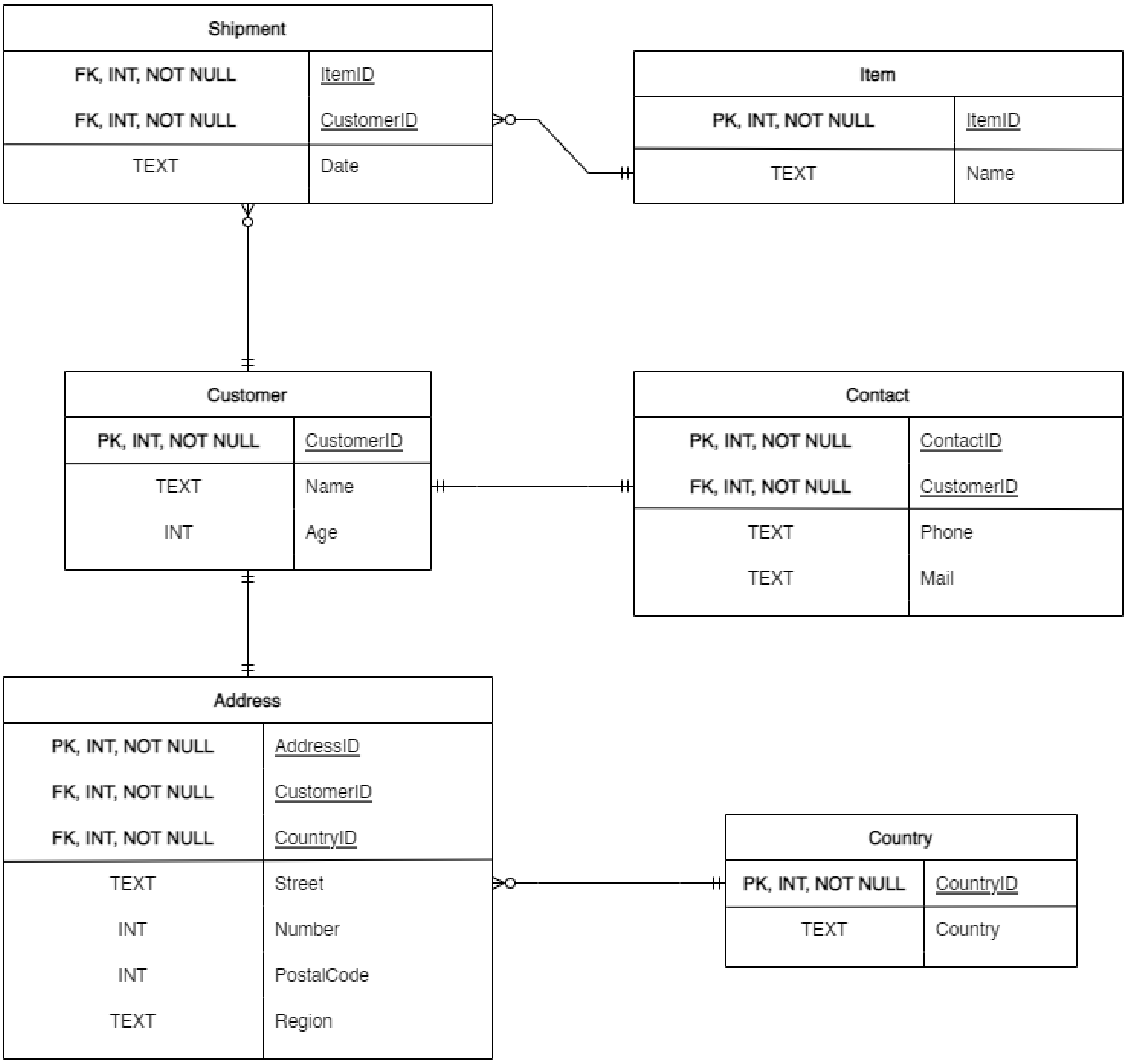
(1) Select a table and view all the information stored in it
(2) Select a table and insert a new row
(3) Select a table and delete a row
(4) Select a table and update a row
(5) Create a chart displaying number of times each item was ordered
(6) List all items each customer has ordered (M-N relationship)

2 MODELING

2.1 Concept model



2.2 Relational model


Shipment
FK, INT, NOT NULL ItemID
FK, INT, NOT NULL CustomerID
TEXT Date
Item
PK, INT, NOT NULL ItemID
TEXT Name
Customer
PK, INT, NOT NULL CustomerID
TEXT Name
INT Age
Contact
PK, INT, NOT NULL ContactID
FK, INT, NOT NULL CustomerID
TEXT Phone
TEXT Mail
Address
PK, INT, NOT NULL AddressID
FK, INT, NOT NULL CustomerID
FK, INT, NOT NULL CountryID
TEXT Street
INT Number
INT PostalCode
TEXT Region
Country
PK, INT, NOT NULL CountryID
TEXT Country

Transformation steps:

2. Group together tables (or entities previously) according to their relationships. If the primary key of one table appears on the other table, it becomes one of the foreign keys of the other table. Foreign key constraints are also added.

Rules:

1. A two-dimensional table contains rows and columns. Rows are instances of the entity and columns are attributes of the entity.
2. No two rows of the table must be identical – each row must be unique.
3. Each column must have a unique name. They must also contain single values – multiple and repeating values are not allowed.
4. A key is a group of one or more attributes that uniquely identifies a row – each table must have at least one key.
5. Every column (attribute) in a table must depend solely on the primary key. It should not depend on any other column in the table.

3 DATABASE IMPLEMENTATION

In general:

Foreign key constraints are created for all foreign keys of each table. During the creation of the database, changes were made to the data types of several columns (or attributes). For instance, the phone number of each customer was integer, but was changed to text due to the implementation of country code (+420, +421). Furthermore, the name format of the customer was originally first name + last name, but was scrapped in favor of a simple full name. An index was to be created at column "Name" of the Customer table to make the 6th query run faster, but the idea was scrapped as it was not necessary. The database is relatively small and compact so an index would not improve performance drastically.

Implementation:

The Python program uses the library PrettyTable to make presentation of data more logical and efficient. When the program is executed for the first time, the user is prompted to hit enter to allow PrettyTable to be installed if it is not already available.

Note during testing: Nothing special to note.

4 DISCUSSION

Nothing to discuss.