

Проект

Бази от данни-практикум

Верига супермаркети

Изготвил: Кардам Гергов

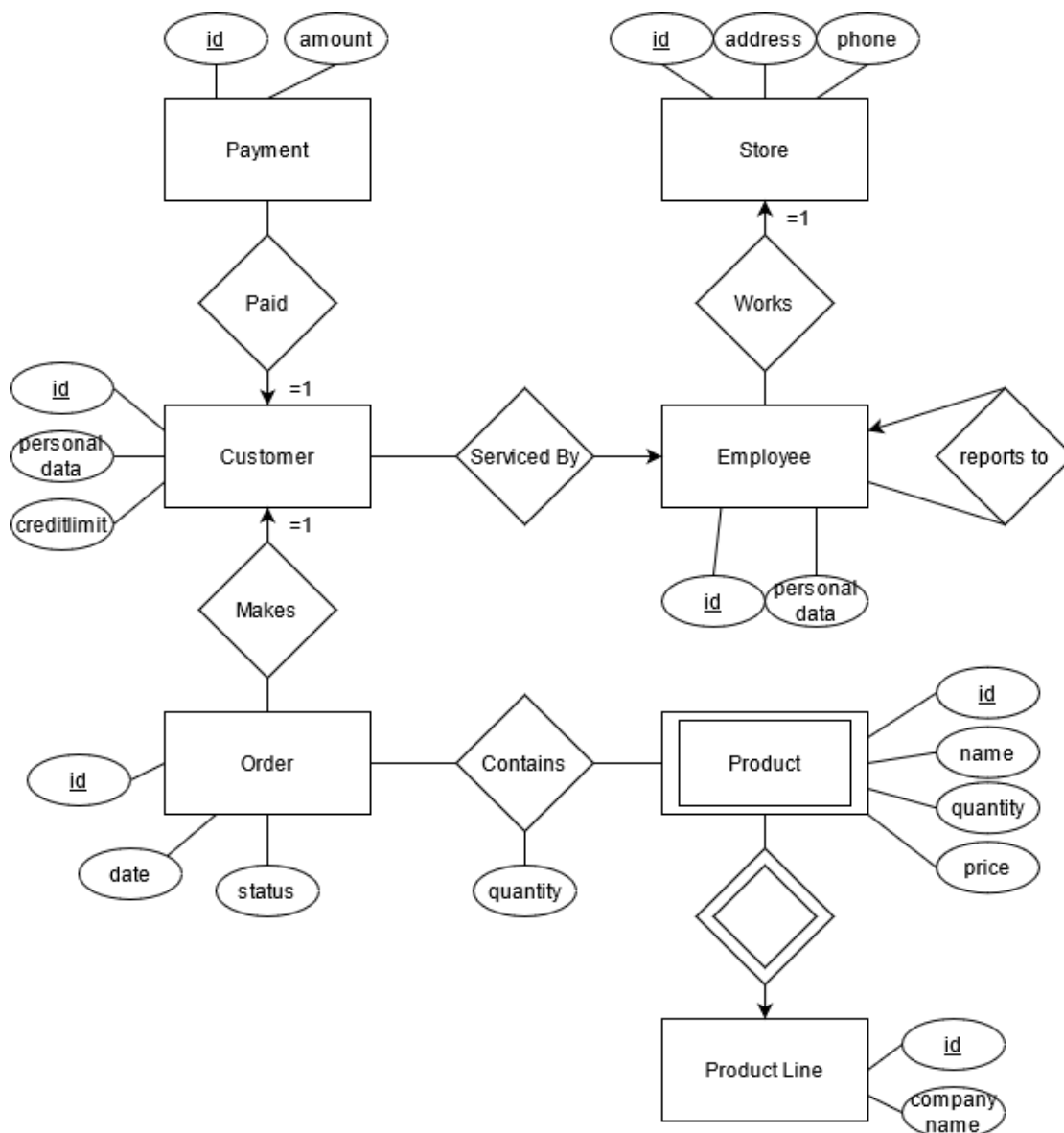


Проектът цели да имплементира проста база от данни за някоя търговска верига супермаркети като има възможности за разширяване на вече съществуващи и добавяне на нови модули.

Базата от данни съдържа функционалности и ограничения които покриват всекидневни проблеми свързани с изготвяне на поръчки, плащания и управление на работниците. Има възможност за доразвиване на модула за плащане(плащане в брой или онлайн, купони) както и добавяне на таблици, така че базата на данни да поддържа транспорт и менажиране на запас.

Моделът поддържа магазини и работници към тях. Всеки път като клиент направи покупка в базата от данни се записват информация за плащането, допълнителни данни за покупката, информация за обслужилия го работник както и сведения за самия клиент.

E/R диаграма:



Store - ИД и адрес на магазин от търговската верига

Employee - ИД, лични данни, началник(и те са служители)

Customer - ИД, лични данни, информация за обслужилия го служител, лимит на картата

Payment - ИД, дължима сума, информация за клиента който се опитва да направи плащането

Order - ИД, информация за клиента който е направил поръчката, състояние на поръчката, дата и допълнителна информация

Product - ИД, име, брой в наличност, информация за продуктова линия

ProductLine - ИД, име на компанията на която е продуктова линия

Релационен модел:

store(storeid, phone, country, city, adressline)

employee(employeeid, storeid, names, email, jobtitle, reportsto)

payment(paymentid, customerid, date, amount)

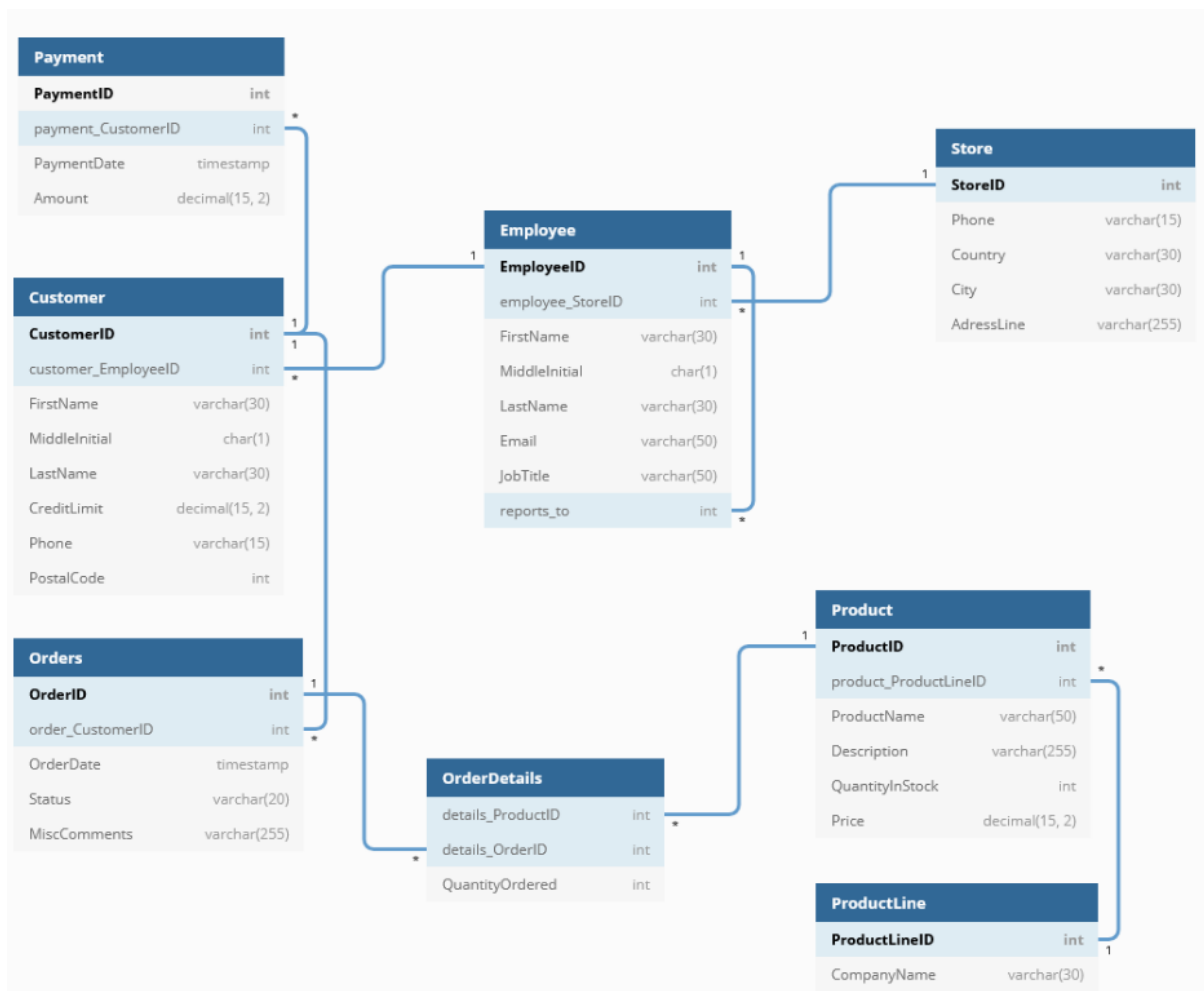
customer(customerid, employeeid, names, creditlimit, phone, postalcode)

orders(orderid, customerid, date, status, comments)

orderdetails(productid, orderid, quantity)

product(productid, productlineid, name, desc, quantity, price)

productline(productlineid, companyname)



Функции:

(1) num_unsuccessful_orders(emp_id int)

По подадено ИД на служител, връща броя на всички поръчки в които той е бил участник и са със статут 'failed'.

(2) num_declined_cards(emp_id int)

По подадено ИД на служител, връща броя на всички поръчки в които той е бил участник в които кредитната карта на клиента е била отказана(всички такива поръчки винаги са със статут 'failed', но е възможно да има и такива при които служителят е допуснал някаква грешка и по негова вина поръчката да не е била осъществена).

(3) should_be_fired(emp_id int)

По подадено ИД на служител, връща булева стойност дали той трябва да бъде уволнен или не(ако има повече от една провалена поръчка по вина на служителя).

Съхранени процедури:

(1) `update_order_status(ord_id int)`

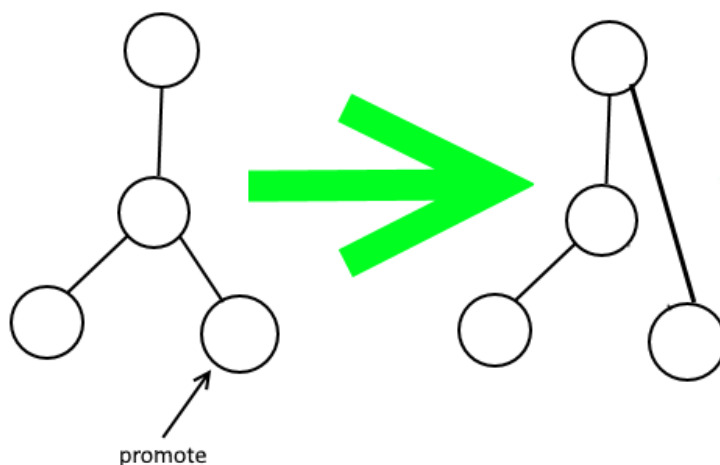
*Проверява дали сумата която трябва да се заплати за дадена поръчка(подадена по ИД на поръчката) надхвърля сумата която клиента има в наличност, ако е така - променя статута на поръчката на *failed*.*

(2) `fire(emp_id int)`

*По подадено ИД на служител проверява дали той изпълнява условието `should_be_fired()` и ако е така, той бива изтрит от базата данни. В противен случай процедурата дава *EXCEPTION*, че не можем незаслужено да уволним служител.*

(3) `promote(emp_id int)`

*Между служителите има йерархия, когато повишим даден служител, той вече директно отговаря на началника на стария си шеф. Ако служителя вече е на върха на йерархията процедурата дава *exception*, че е невъзможно да го повишим.*



Тригери:

(1) after_insert_orderdetails

При вмъкване на кортеж в orderdetails, ние сме включили определено количество от някакъв продукт към конкретна поръчка.

*Този продукт съответно има цена, за това в таблицата payment цената която трябва да бъде заплатена се увеличава с (цена на продукта) * (количество от продукта).*

(2) before_delete_orderdetails

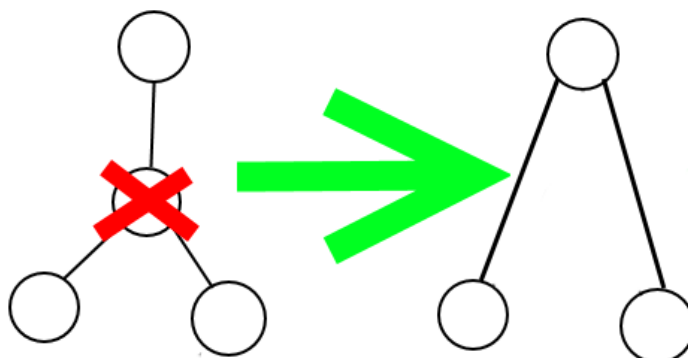
Напълно аналогично на горното, при премахване на продукт от кошницата вадим от сумата която трябва да бъде заплатена.

(3) before_delete_orders before_delete_customer before_delete_store

Тъй като част от връзките много-един в базата данни имат условие за not-null е необходимо специално да се погрижим при изтриване.

(4) before_delete_employee

При изтриване на даден работник е необходимо да запазим целостта на йерархията за това преместваем подчинените му.



Изгледи:

(1) invalid_payments

Таблица в която за всяка поръчка, прекратена поради отхвърлена кредитна карта, са изредени дата на поръчката, името на клиента и всичките продукти които се е опитал да закупи.

(2) best_employee

*Таблица в която за всеки магазин е изведен най-добрият служител с ИД и първо име.
Най-добър служител е този, който е участвал в най-много поръчки.*