

Обхват на модела. Дефиниране на задачата

Базата от данни за агенцията за коли под наем ще съхранява информация за данните в агенцията. Софтуерът, който ще обслужва е вътрешна система за агенцията, която се използва само от служителите на агенцията при работа с клиенти, т.е. няма да обслужва сайт на агенцията.

Агенцията разполага с няколко обекта, които са разположени в различни градове. В рамките на един и същи град може да има само един обект. Всеки обект се характеризира с име и адрес като името, който е името на града, на обекта еднозначно определя съответния обект. Два различни обекта не могат да имат еднакви имена, защото са в различни градове.

Всеки обект се управлява от управител, който може да бъде управител само на един обект и всеки обект може да има само един управител. Управителя се характеризира с име, ЕГН, телефон, емайл и служебен номер – уникален.

Към всеки обект работят набор от служители, които се характеризират със служебен номер – уникален, ЕГН, име, емайл, телефон и позицията, която заема. Всеки служител отговаря само за обекта, в който работи т.е. служител може да работи само в един от обектите, а всеки обект може да има повече от 1 служител.

Всеки обект има определен брой набор от уникални коли на разположение, които се дават под наем на клиенти. Колите се характеризират с модел, застраховка, километри, регистрационен номер – уникален, информация за това дали е нает и цена за ден.

Агенцията работи с клиенти, за които пази информация. Съхраняват се номер на лична карта, ЕГН, име на клиента, емайл и телефонен номер. Клиентите се определят еднозначно по телефонен номер. Клиентите могат да наемат кола от съответен обект като се изисква предоставяне на шофьорска книжка. При наем на кола се пази датата на наемане, периодът, за който се наема, дата на връщане и дължимата сума.

Базата от данни пази информация за историята на всички сключени сделки за наеми на всеки обект. Една сделка за наем на кола се състои между служител от съответния обект, клиент и колата, която се наема. Историята на сделките се характеризира със съотношения между обекта, от който се наема, служител, който дава съответната уникална кола под наем и клиент, който наема колата за период от време за определена цена. При наема на кола клиента предоставя шофьорска книжка и личните му данни се записват. Пази се информация за датата на наемане, периода и цената, която е заплатена. Има номер на сделка, която уникално определя една история.

Множества от същности и техните атрибути

- Гражданин – ЕГН, име, телефон, емайл, рождена дата.
 - Управител – служебен номер, един обект за който са отговорни ○
Служител – служебен номер, обект, в който работят.
 - Клиент – номер на лична карта, шофьорска книжка
- Обект: име, адрес и един управител, който е отговорен за обекта.
- Кола: регистрационен номер, модел, застраховка, километри, цена за ден, дали е наета
- Сделки: обект, служител, клиент, кола, дата на сделка и период на наем, цена, шофьорска книжка на клиент.

Домейн на атрибутите

- Гражданин – ЕГН: низ, име: низ, телефон: низ, емайл: низ, рождена дата: низ ○
Управител – служебен номер: низ, един обект за който са отговорни: низ ○ Служител –
служебен номер: низ, обект, в който работят: низ ○ Клиент – номер на лична карта: низ,
номер на шофьорска книжка: низ
- Обект: име: низ, адрес: низ, един управител, който е отговорен за обекта: низ
- Кола: регистрационен номер: низ, модел: низ, номер на застраховка: низ, километри:
цяло число, цена за ден: реално число, наета: булев тип
- Сделки: обект: низ, служител: низ, клиент: низ, кола: низ, дата на сделка: низ, период
на наем: цяло число, цена: реално число, номер на шофьорската книжка на клиента:
низ.

Връзки

- Един гражданин може да бъде управител, служител или клиент.
- Един обект има точно един управител. Един управител може да бъде управител само
на един обект.
- В един обект работят много служители. Служителите работят точно в един обект.
- Един обект може да притежава много коли. Една кола е точно на един обект.
- Една кола в даден момент може да бъде наета само от един клиент.
- Един клиент може да наеме повече от една кола.
- За всяка сделка има точно една запазена история.

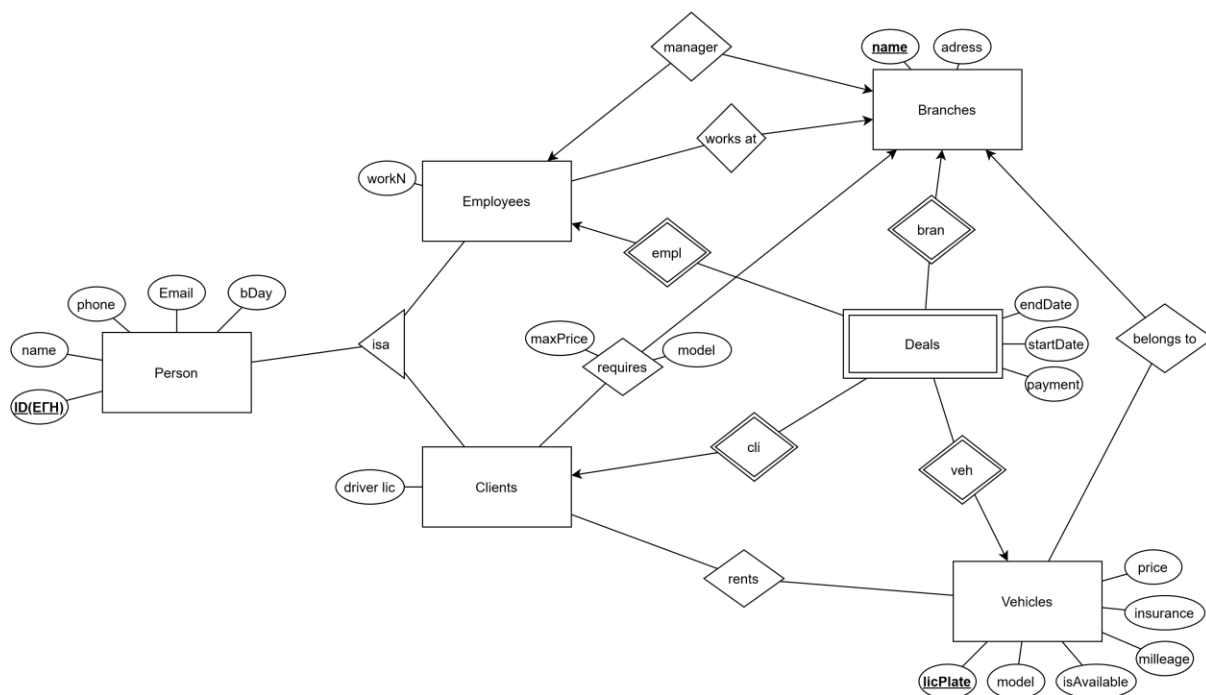
Ограничения по единствена стойност, референтна цялостност и друг тип ограничения

- Гражданин: ЕГН: еднозначно определя един гражданин, телефон: проверка за валиден
телефон, емайл: проверка за валиден емайл
- Управител: служебен номер: еднозначно определя управителя
- Служител: служебен номер: еднозначно определя служителя
- Клиент: телефонен номер: еднозначно определя клиента
- Обект: име: еднозначно определя обекта
- Кола: регистрационен номер: еднозначно определя колата
- Сделка: номер на сделка: еднозначно определя сделката

Правила и проверки

- При наема на кола от клиент, задължително се прави проверка за наличие на валидна
шофьорска книжка.
- В рамките на един и същи град може да има само един обект

Е/R модел на данни



Релационен модел на данни

Employees (ID, name, phone, email, bDay, workNo, branch_name, manager_id)

Clients (ID, name, phone, email, bDay, driverLic)

Vehicle (licPlate, model, isAvailable, mileage, insurance, price, branch_name)

Branches (name, address, manager_id)

Deals (startDate, endDate, payment, client_id, employee_id, vehicle_licPlate, branch_name)
 Requires (date, model, maxPrice, client_id, branch_name)

Преобразуване:

Първо сме преобразували isa-йерархията, после множествата от същности, после слабото множество, после връзките много-много и накрая връзките много-един. Връзките много-един са оптимизирани. За преобразуването на isa-йерархията може да използвам три подхода: ER, ООП и NULL. За isa-йерархията Person-Client-Employee най-подходящи са ООП и NULL-подхода, защото може служител на обект да бъде и наемател на кола от агенцията (т.е. клиент).

В базата от данни не може да има данни за хора, които не са нито клиенти, нито служители. Може да има хора, които са едновременно клиент и служител. Така като използвам ООП подхода от 4-те възможни релационни схеми смисъл ще имат само PersonClient, PersonEmployee. Не създавам 3та таблица PersonClientEmployee, защото при него така или иначе ще се повтарят данни за служителя. Ако служител е и клиент на агенцията (случаите ще са малки) ще се създава нов ред за него и в таблицата за клиенти и ще се третира като нормален клиент.

Като използваме ООП-подхода и игнорираме празните таблици (релационните схеми, в които никога няма да има редове) получаваме:

1. PersonClient (ID, name, phone, email, bDay, driverLic)
2. PersonEmployee (ID, name, phone, email, bDay, workN)

Преименувам релационните схеми PersonClients на Clients и PersonEmployee на Employees.

Продължаваме с другите множества от същности:

- Vehicle (licPlate, model, isAvailable, mileage, insurance, price) - Branches (name, address)

За слабото множество Deals, след преобразуването на връзките получаваме:

- Deals (startDate, endDate, payment, client_id, employee_id, vehicle_licPlate, branch_name)

След преобразуването на връзките и оптимизирането им окончателно се получава:

- Employees (ID, name, phone, email, bDay, workNo, branch_name, manager_id)
- Clients (ID, name, phone, email, bDay, driverLic)
- Vehicle (licPlate, model, isAvailable, mileage, insurance, price, branch_name)
- Branches (name, address, manager_id)
- Deals (startDate, endDate, payment, client_id, employee_id, vehicle_licPlate, branch_name)
- Requires (date, model, maxPrice, client_id, branch_name)

Ф3 и Нормализация

Проверяваме за Ф3 в релационните схеми, които нарушават НФБК.

- Ф3-1: ID(EГН) -> name, phone, email, bDay, workNo, branch_name, manager_id (Employees)
- Ф3-2: ID(EГН) -> name, phone, email, bDay, driverLic (Clients)
- Ф3-3: licPlate -> model, isAvailable, mileage, insurance, price, branch_name (Vehicles)
- Ф3-4: name -> address, manager_id (Branches)
- Ф3-5: startDate, endDate -> payment, client_id, employee_id, vehicle_licPlate, branch_name (Deals)
- Ф3-6: date, client_id, branch_name -> model, maxPrice (Requires)

За всички Ф3 на релациите е в сила, че в лявата част се намира суперключ за релацията, следователно всички релации са в НФБК. Не се налага да правим нищо допълнително.