

## 4. Релационен модел

Category	Empty
Files	Empty
Created	May 28, 2023 11:22 AM
Reminder	Empty
Status	Open
URL	Empty
Updated	May 28, 2023 11:22 AM

---

### Задача - концептуален дизайн

Малка БД за комерсиална банка

Следните същности

- клиенти
- акаунти
  - акаунти към branches
- заеми
- клонове на банки

Как да представим тази информация:

Клиенти на банката, които теглят някакви заеми, обикновено имат някакви сметки/ акаунти

Банката има много клонове на много места

#### 1. Определме същности

Клиент атрибути:

- Имена, ЕГН, адрес, дата на раждане

Акаунт атрибути

- Клиент, наличност, валута, дневен лимит за теглене от банкомат, **номер на акаунта**

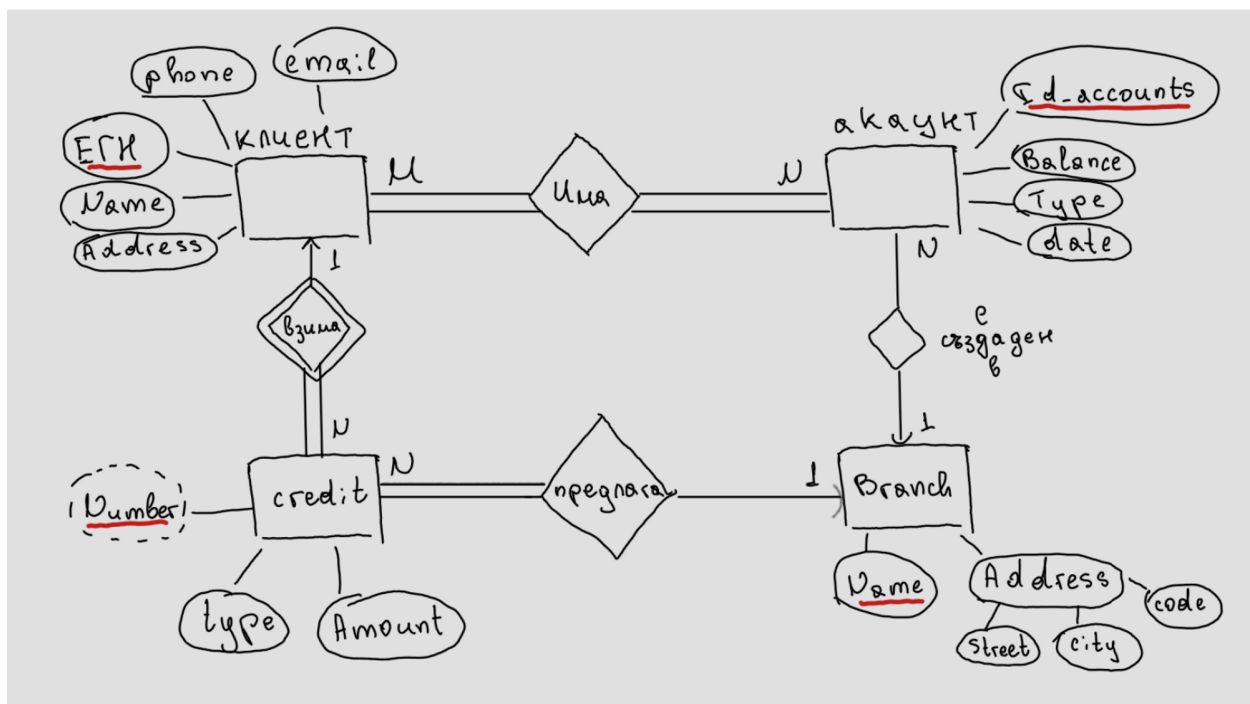
Заеми

- от кои клиенти, сума, дата на даване, сума за изплащане на месец

Клонове на банки

- Име на клона, идентификационен номер?

Много към много - клиенти и заеми



Number - атрибут на слабо множество същности, потенциален атрибут на ключа

## Multivalue attributes

Примерно човек може да има много телефони

Моделът същност-връзка е модел, който позволява по-голяма абстрактност

# Релационен модел

Релация - двумерна таблица, в която се съхраняват данни

- основава се на математическото понятие **релация**
- Името на релацията и множеството от атрибутите ѝ формират схемата на релацията

Множеството от всички схеми на релации в една БД - схема на БД - relational database schema

Съвкупността от всички схеми, която представлява съвкупността от имената и таблиците ни дава схемата

- Съответства на човешкото мислене

## Основни понятия

- Релация
- Атрибут
- Схема
- Кортеж
  - самите данни, които се съхраняват в тази БД
  - Може да заемат стойности в определен домейн
- Домейн

Пример

Заглавният ред определя схемата на БД

## Атрибути

Атрибутите са с означения на колоните на релацията

- title, year,...
- Асоциират се с имената на колоните в релацията

Разлика с Моделът същност-връзка:

- имаме ограничения за атрибутите, свързани с това какви стойности може да имат - това са т.нар. **атомарни стойности**

Задължително трябва да определим кои атрибути определят еднозначно същностите

## Кортежи

Редовете в релацията, които съдържат конкретните стойности на всеки компонент

- пример: (Star wars, 1977, 124, Color)
- следва се вече определена последователност от атрибутите от релационната схема
- Релацията е множество от кортежи
- Релационният модел изисква всеки компонент на кортеж да бъде атомарен

## Домейни

- Асоциират се с всеки атрибут
- Компонентите на всеки кортеж принадлежат на домейна за съответния атрибут



**Cardinality** of a relation is the number of tuples in it

## Екземпляри на релацията

Множеството от кортежи за дадена релация ще наричаме екземпляр на релацията

Крайните потребители могат да променят екземплярите на релацията.

Една реална релация се променя с времето

- релациите съхраняват current instance



### Схема ≠ Екземпляр

Важно е да различаваме схемата какво представлява, кои са начините да я редактираме и екземплярите - текущото състояние на базата данни

По-бавно се правят промените в релационните схеми.

- има изискване за консистентност/атомарност...

## Схеми

Схемата на релацията се образува от името на релацията и от имената на атрибутите

Изписваме схемата на релацията като посочваме името, следвано от атрибутите в скоби

- редът на атрибутите няма значение, атрибутите в релационната схема са множество, а не списък. Независимо от това ние трябва да укажем стандартна наредба

Schema: Movies(title, year, length, filmType)

Самата база данни - самите данни, които са структурирани в релации

- съвкупността от всички релации, които имаме

Релационните схеми са дефинициите на релациите

Релацията е **множество от уникални кортежи**

## От E/R диаграми към релации

Основни правила

- **всяко множество** от същности се преобразува в отделна **релация** със **същите атрибути**
- **всяка връзка** се преобразува в **релация**, чиито атрибути са:
  - **ключовете** на свързаните множества същности
  - **атрибутите** на връзката (ако има такива)

## Изключения

- слабите множества същности не могат да се преобразуват директно в релации



Ще добавим ключовите артибути на "поддържащата" същност

Поддържащата връзка става излишна, оказваме се от отделна релация за слабата връзка и към слабото множество същности просто добавяме ключовия атрибут на силното множество същности

- "ISA"-връзките не могат да се пренесат директно

В примера с Relationship Types 1:1

се отказваме от едната релация, като съберем двете релации и получим в една всички необходими връзки. Използваме задължителността на връзката от страната на студиото

По същия начин може да процедираме, когато имаме връзка N:1

Когато имаме връзка N:1, тогава не използваме отделна релация за връзката, а към множеството същности, стоящо откъм страната много добавяме ключовете от множеството същности от страната 1.

## Рискове при връзки M:N



Ако се опитаме да представим връзката подобно на N:1, ще получим доста повторения/излишество, което не е добре.

- Ако има атрибути на връзката се добавя при табличката на "многоото"



Когато връзката е много към много (M:N) генерираме **отделна релация**

## Преобразуване на йерархиите от множества същности

(Superclasses, subclasses)

## 1 Начин - E/R подход

За всяко множество същности се създава релация, включваща ключовите атрибути на корена

- ако множеството не е корен се добавя ключът на корена, както и допълнителните атрибути(които са допълнителни/отличаващи за тези подкласове), ако има такива

## 2 Начин

За всяко възможно поддърво, вкл. корена да направим отделна релация и в тази релация да включим всички атрибути. Връзката отново представяме като отделна релация, защото е много към много

## 3 Начин Null стойности

Всички възможно атрибути от всички множества същности в една релация

Отново имаме отделна релация за връзките. Имаме само една релация и тя включва всичко.

Основно се ползват 1. или 3.-ти подход

## Преобразуване на атрибути

### Съставен атрибут

Разделяме го на отделни атрибути - всеки компонент на съставния атрибут се преобразува в отделен атрибут

### Ключов атрибут

Очакваме че ключът на множеството същности става и ключ на релацията (но има особености)

Избира се само един за **primary key**

### Множествени атрибути

Могат да имат няколко различни стойности. Пример с телефонен номер - всеки човек може да има няколко телефонни номера/имейл адреса

За тях се прави отделна релация с атрибут името на множествения атрибут

## Изчислим атрибут (Derived)

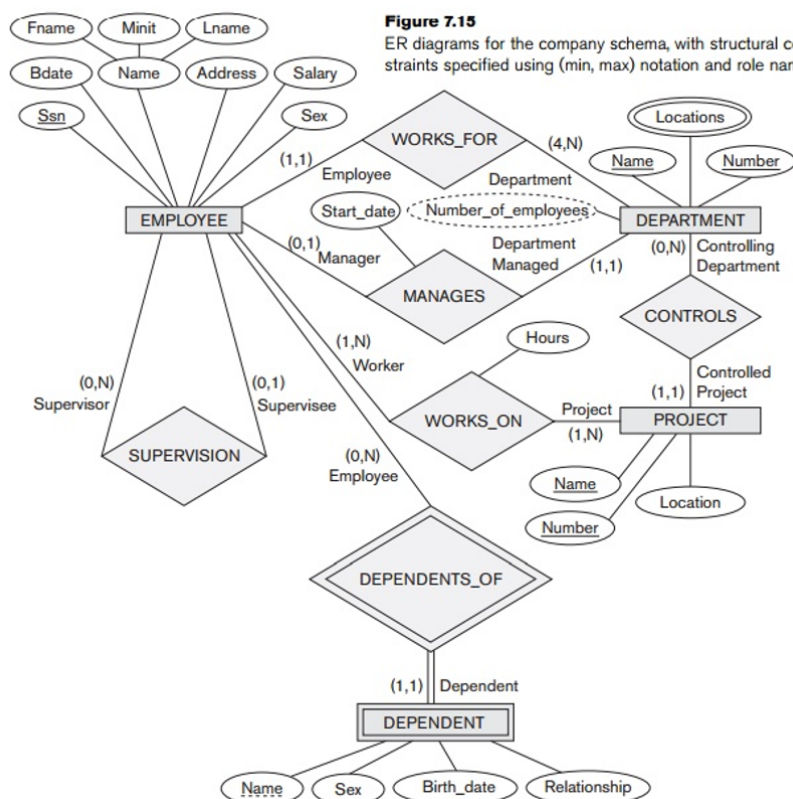
Може да бъде изчислен от друг атрибут

Примерно, ако имаме дата на раждане, по нея можем да изчислим възрастта

Не е хубаво тези изчислими атрибути да бъдат вкарвани като атрибути на релацията, за да не се получава излишество (тази възраст трябва да се сменя непрекъснато всяка година)

Тези атрибути не се преобразуват в атрибути на релацията

## Задача 7.15



### 1. Преобразуваме същностите



## 2. Преобразуваме връзките

### Employee - атрибути:

(SSN, Bdate, Fname, Minit, Lname, Address, Salary, Sex, Supervisor, WorksForDepartmentNumber)

Department - атрибути:

(Name, Number, ManagedBy\_EmployeeSSN, ManagedEmployeeStartDate, DLNumber)

## Department Location - атрибути

(DLNumber, DLocation)

## Project - атрибуты

(Name, Number, ProjectLocation, DepartmentNumber)

## Dependent

(Name, Employee\_SSN, Sex, Birth\_date, Relationship)

## Works on - relation - атрибути

(SSN, ProjectNumber, Hours)

Employee									
Ssn	Bdate	Fname	Minit	Lname	Adress	Salary	Sex	Ssn_Supervisor	Dep_Number
Dependent									
Name	Sex	Birth_date	Relationship	Ssn					
Project									
Name	Number	Location	Dep_Number						
Department									
Name	Number	D_Number	Ssn_Manager	Start_date					
Works_on									
Ssn	P_Number	Hours							

