# 

PKDD'99 Discovery Challenge

Курсова работа по “Складове от данни и бизнес анализ”

Зимен семестър, 2022 / 2023

| *Изготвено от:*  *Мирослав Дионисиев, 62390*  *Надежда Францева, 62391*  *Павел Сарлов, 62393* | *Преподавател: Тодор Кичуков* |
| --- | --- |

[**1. Описание на задачата**](#_qwv0c7k8rk1k) **3**

[**2. Обосновка на избрания подход за изграждане на DWH**](#_tr3gvgcbtqs6) **5**

[**3. Описание на ETL процеса**](#_gxm4qtx4fiiz) **6**

[**4. Модел на входните данни**](#_aqfaxrbmah6f) **7**

[**5. Staging модел на данните**](#_ukojk1jm29uy) **8**

[**6. Data Vault модел на данните**](#_rmwm1mz7ik92) **9**

[**7. Дименсионен модел**](#_3v6susy5bwsu) **11**

[**8. Полезни доклади**](#_r2u3t818g1i2) **13**

[**9. Използвани софтуерни инструменти**](#_kgxo1cv8e35u) **16**

# 

# Описание на задачата

Имало едно време банка, предлагаща услуги. Сред тях: управление на акаунт, предлагане на заеми и т.н. Гореспоменатата банка се стреми да подобри своите услуги, като прави разлика между потенциално „добри“ и „лоши“ клиенти. Банковите мениджъри имат само бегли идеи за това на кого биха искали да предложат някои допълнителни услуги и кого да наблюдавате внимателно, за да опитат и минимизират риска и впоследствие - банковите загуби. За щастие банката съхранява данни за своите клиенти, техните сметки, извършените транзакции, отпуснатите заеми и издадените кредитни карти. Така че банковите мениджъри се надяват, че има начин, по който биха могли да се възползват от данните и да извлекат някои важни прозрения.

**Описание на данните:**

* релация ***account*** (4500 кортежа във файла **ACCOUNT.ASC**) - всеки запис описва статичните характеристики на акаунта
* релация ***client*** (5369 кортежа във файла **CLIENT.ASC**) - всеки запис описва характеристиките на клиента
* релация ***disposition*** (5369 кортежа във файла **DISP.ASC**) - всеки запис свързва клиент с акаунт
* релация ***permanent\_order*** (6471 кортежа във файла **ORDER.ASC**) - всеки запис описва характеристиките на платежно нареждане
* релация ***transaction*** (1056320 кортежа във файла **TRANS.ASC**) - всеки запис описва една транзакция на един акаунт
* релация ***loan*** (682 кортежа във файла **LOAN.ASC**) - всеки запис описва заем, предоставен за дадената сметка
* релация ***credit\_card*** (892 кортежа във файла **CARD.ASC**) - всеки запис описва кредитна карта, издадена към сметката
* релация ***demographic\_data*** (77 кортежа във файла **DISTRICT.ASC**) - всеки запис описва демографските характеристики на областта

Всеки акаунт има както статични характеристики (напр. дата на създаване, адрес на клона) в релацията ***account***, така и динамични характеристики (напр. плащания, салдота) в релациите ***permanent\_order*** и ***transaction***. Релацията ***client*** описва характеристиките на хората, които могат да боравят с акаунтите. Всеки клиент може да има няколко акаунта, няколко клиента могат да менажират един акаунт. Клиентите и акаунтите са свързани чрез релацията ***disposition***. Релациите ***loan*** и ***credit\_card*** описват някои от услугите, които банката предлага на клиентите си. Към един акаунт могат да бъдат издадени няколко кредитни карти, но максимум един заем може да се асоциира с акаунт. Релацията ***demographic\_data*** дава публично достъпни данни за регионите (напр. процент на безработица), които могат да послужат като допълнителна информация за клиентите.

# 

# Обосновка на избрания подход за изграждане на DWH

За реализация на текущата задача екипът ни се спря на хибридния подход на Дан Линстед. *Data Vault* моделът осигурява баланс между нуждата за зареждане на данните в реално време и пълната им интеграция в складът за данни. Също така е доста гъвкав и следователно подходящ за системи с променливи бизнес изисквания. Този модел е много стабилен във времето заради своята разделена структура и съдържание. Също така е мащабируемо решение, като позволява бързо интегриране на данните на различни платформи. Самият модел реагира добре на нови предметни области. Така може да се разширява и да се адаптира с минимално въздействие.

За текущата задача имаме само един единствен източник на данни, т.е. данните са предвидими и с ограничен обхват. Също така, предимствата на зареждането на данните в реално време и тяхната проследимост за текущата задача са малко излишни, но в случая на подобна система от реалността тези черти на склада от данни биха били от най-голямо значение, имайки предвид, че боравим с транзакции.

# 

# Описание на ETL процеса

*CBS (Core banking system)* е единственият източник на данни в съответствие с целите на проекта. Данните в базата са заредени от предоставените *CSV* файлове. Всички *ETL* процеси се изпълняват веднъж (с други думи, данните се качват / актуализират само веднъж, а не ежедневно / седмично / месечно, както се случва в действителност).

* От *CBS* до *SA (Staging Area)* - всички записи от *CBS* таблиците се вмъкват в съответните *SA* таблици. Датата на изпълнение на заявката се добавя като ***staged\_at*** във всяка *SA* таблица. Процеса на зареждане на данните в *SA* е осъществен чрез изпълнение на заявки, които извличат нужните данни от source базата и ги записват в съответните таблици в *SA*.
* От *SA* до *DV (Data Vault)* - за всяка от същностите в *SA* се създава хъб (*Hub)*, държащ сурогатен ключ (***id***), генериран спрямо брояч на постъпилите в таблицата кортежи, дата на зареждане на кортежа (***loaded\_at***), източник на постъпилите данни (***source***) и бизнес ключа на самите данни (<***\*>\_id***). Всеки хъб си има съответен сателит (*Satellite)*, който се свързва към хъба посредством външен ключ и включва характеристичните данни за съответната същност, както и споменатите дата и източник. Хъбовете са свързани посредством линкове (*Link*), които отново съдържат сурогатен ключ, дата и източник. Всеки линк представлява част от бизнес логиката на системата. Отделно има и референтни таблици, които съдържат фактологични данни. Те съдържат сурогатен ключ и съответстващите им данни.
* От *DV* до дименсионен модел - преминаването от хъбове, линкове и сателити към дименсии и факти е доста праволинеен - хъбовете и съответните им сателити стават дименсии, а линковете и съответните им сателите стават факти. В случая нашите линкове нямат сателити. Референтните таблици по принцип също се превръщат в дименсии, но за текущата задача решихме те да присъстват като колони в дименсиите, които ги използват. Това се прави с цел да не се губят кортежи, при които липсва съответната референция.

Заявките за целия процес по създаването на базите, ETL процеса и извличането на докладите по-надолу, са в папката ***scripts/***. Файловете за самите модели са в папката ***models/***.

# Модел на входните данни

# 

# Staging модел на данните

За модела на *SA* сме въвели следните промени в сравнение с модела *CBS*:

* Таблиците вече не са свързани една с друга.
* Няма ограничения в таблиците (с изключение на *PKs*).
* Ново поле - ***staged\_at*** от тип DATE - се добавя към всяка таблица, така че се знае точно кога данните са били заредени/актуализирани за последен път.

# Data Vault модел на данните

Хъбовете са таблиците в червено (или чиито имена започват с ***h\_***). Сателитите са таблиците в жълто (или чиито имена започват с ***s\_***). Линковете са таблиците в зелено (или чиито имена започват с ***l\_***). Референтните таблици с тези без никакви релации (или чиито имена започват с ***r\_***).

С цел по-добра яснота на данните са добавени някои изменения по имената на някои колони (най-вече за ***demographic\_data***, където колоните бяха от типа ***a1***, ***a2***, ***a3…***). Новите наименования са съгласувани с обясненията в документа на предизвикателството.

Също така е извършено следното трансформиране на част от данните:

* Колоните ***date*** са извлечени като цели числа, които представляват дати във вида ***YYMMDD***. В *DV* вече са колони от тип ***DATE*.**
* За колоната ***issued*** при кредитните карти в пояснението на данните е упоменато, че ***issued*** е във вида ***YYMMDD***. Забелязахме обаче, че беше във вида ***‘YYMMDD hh:mm:ss’***, което си е чиста времева марка. Затова и използвахме ***TIMESTAMP*** като тип.
* Фактологичните данни като ***k\_symbol, operation, type*** и т.н. са пренесени в референтните таблици и ключовете за съответните стойности се реферират от сателитите, където е приложимо. Някои от кортежите съдържаха мръсни фактологични данни (например ***k\_symbol*** при някои транзакции или липсваше, или беше някакъв низ с интервали). Подобни стойности са игнорирани и ключът в съответните сателити е *NULL*.

# Дименсионен модел

Дименсиите са таблиците в лилаво (или чиито имена започват с ***d\_***). Фактите са таблиците в бежово (или чиито имена започват с ***f\_***).

Нашият дименсионен модел се състои от 6 факт таблици, които се допълват от 10 дименсии. Както споменахме по-горе, хъбовете със съответните им линкове стават на дименсии, а линковете - на факти. Тъй като референтните данни са характеристични и поради липсата им в някои от кортежите като външни ключове сме ги преместили в дименсиите, където намират приложение. Така няма да срещаме проблеми поради *NULL* външни ключове.

Предварително са направени следните дименсии:

* ***d\_date*** - раздробение на дата на година, месец, ден. Съдържа всички дати, които се срещат в данните.
* ***d\_time*** - раздробение на време на час и минути. Съдържа 24 x 60 = 1440 кортежа за всеки час и всяка минута от денонощието.

За някои от фактите имаше числени данни, които могат да послужат за най-различни агрегации:

* ***f\_transaction***
  + **amount** - стойността на изпълнената парична транзакция
  + **balance** - ново салдо на съответния акаунт
* ***f\_permanent\_order***
  + **amount** - стойността на поставената поръчка
* ***f\_loan***
  + **amount** - размер на изтегления заем
  + **payments** - размер на месечните вноски по заема
  + **duration** - също е числена колона, но не намерихме причина да се използва в някаква агрегация, затова е изместена в дименсията

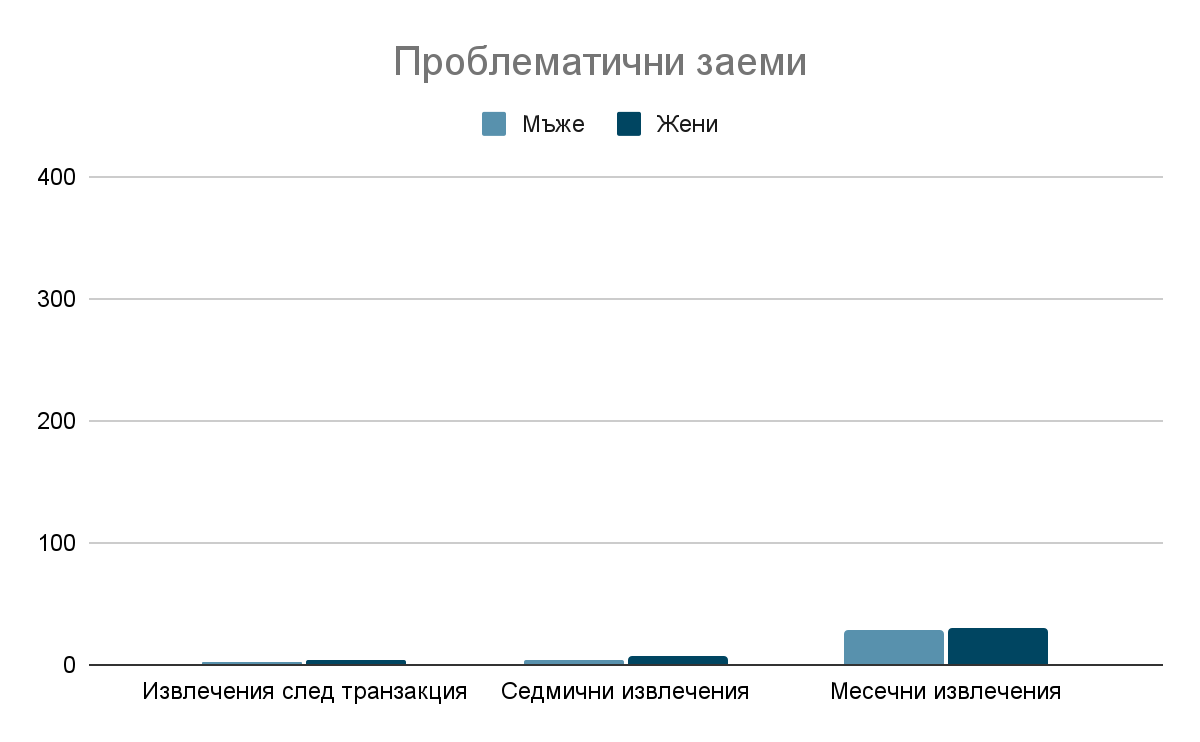
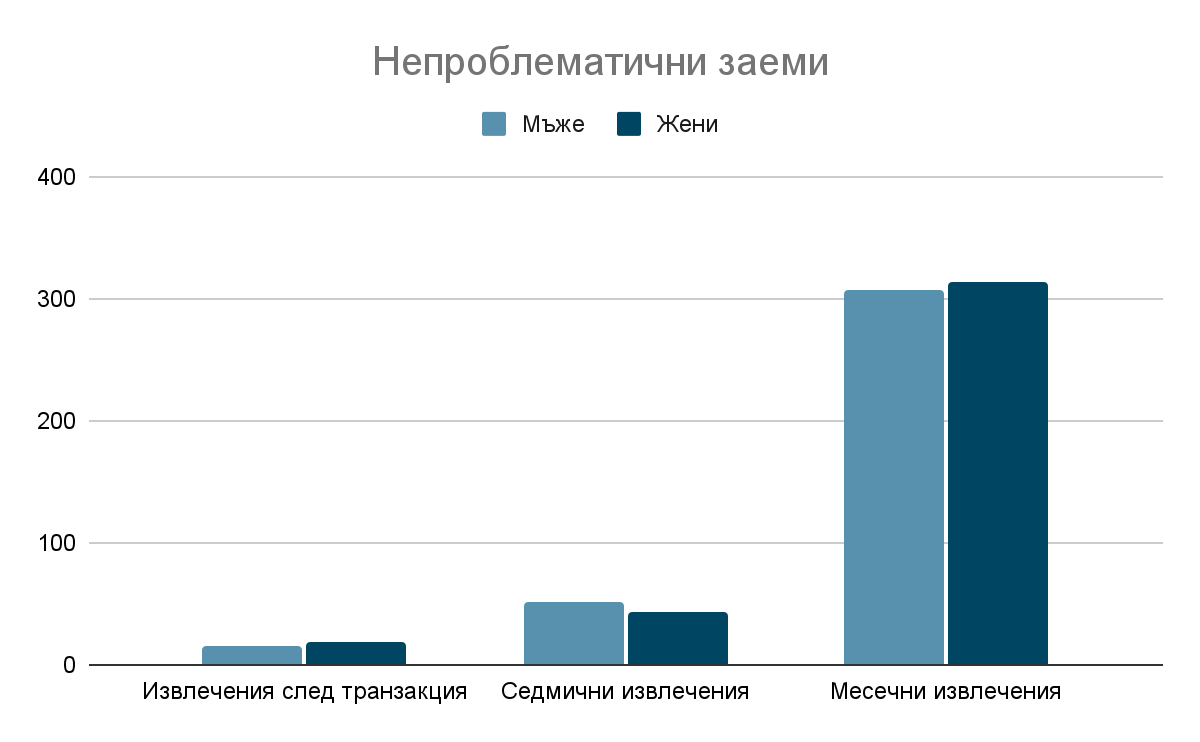
# Полезни доклади

* Ако някога сте се чудели колко пари се изплащат за застраховки, ще разберем с една проста заявка към ***f\_transaction*** таблицата, обединена с ***d\_transaction***, като за застраховки търсим ***k\_symbol*** ***= ‘POJISTNE’***. В следната таблица можем да видим десетте акаунта с най-големи такива транзакции:

|  | account\_id | max\_amount |
| --- | --- | --- |
| 1 | 4386 | 12504.00 |
| 2 | 3592 | 10608.00 |
| 3 | 4312 | 9115.00 |
| 4 | 3448 | 8742.00 |
| 5 | 1733 | 8253.00 |
| 6 | 4079 | 7996.00 |
| 7 | 4349 | 7793.00 |
| 8 | 4121 | 7396.00 |
| 9 | 4400 | 7082.00 |
| 10 | 2229 | 6970.00 |

**Таблица 1: Топ 10 акаунти с най-високи застраховки**

* Нека да разгледаме заемите, групирани по пол на клиента, вид на заема (проблематичен - ***status*** е ‘A’ или ‘C’; непроблематичен - ***status*** е ‘B’ или ‘D’), както и честота на извлечения от съответния клиент:



Както може да се види от горните графики повечето извлечения се правят месечно. Тоест можем да стигнем до извода, че клиенти, които по-рядко следят извлеченията си, са по-изкушени от това да поискат заем.

Не се забелязва корелация между изтеглените заеми и пола - и мъжете, и жените са еднакво вероятни да изтеглят заем. Също така, можем да кажем, че тенденцията е подобна и за двата вида заеми (проблематични и непроблематични). Тоест честотата на извлеченията не е определима за това дали клиентът ще изплати заема си.

Заявките за горните кратки доклади могат да бъдат намерени във файла ***reports.sql*** в папка ***scripts/***.

# Използвани софтуерни инструменти

* **Oracle Data Modeler** - моделиране на логическите и релационните модели и генериране на SQL код
* **PostgreSQL** - прилагане на генерирания SQL код, като за целта се наложиха малки поправки поради разлики в използваната релационна база данни