

Darbas su duomenimis

0. Aplinkos paruošimas

1. Susipažinimas su SQL kalba

Aplinkos paruošimas

- Mes naudosime
 - Dbeaver <https://dbeaver.io/download/>
 - Sqlite <https://www.sqlite.org/index.html>
 - MySQL <https://dev.mysql.com/doc/>
- Sqlite atsisiųsime per Dbeaver aplikaciją.
- Mysql instaliavimo failo nuoroda
<https://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-web-community-8.0.27.1.msi>

Aplinkos paruošimas SQLite

1. Instaliuojame Dbeaver CE (community edition).
2. Atidarome Dbeaver programą.
3. Einame Database -> New Database Connection. Nurodome duomenų bazės failo **chinook.db** lokaciją.
4. Peržiūrimė lenteles ir jų duomenis.
5. Pasirenkame duomenų bazę ir einame SQL Editor -> New SQL Script
6. Parašome **select * from genrese** ir paspaudžiame Ctrl+Enter.
7. Peržiūrimė rezultatus.

Aplinkos paruošimas MySQL

- Paleidžiame atsisiųstą failą.
- Instaliuojame MySQL serverį.
- Per start mygtuką randame ir paleidžime **MySQL 8.0 Command Line Client**

Aplinkos paruošimas (jeigu negalim dirbti lokasioje aplinkoje)

- <https://www.katacoda.com/mysql-db-sandbox/scenarios/mysql-sandbox>
- Sekite nuorodą, prisiregistruokite ir galėsite naudotis MySQL klientu.

Literatūra

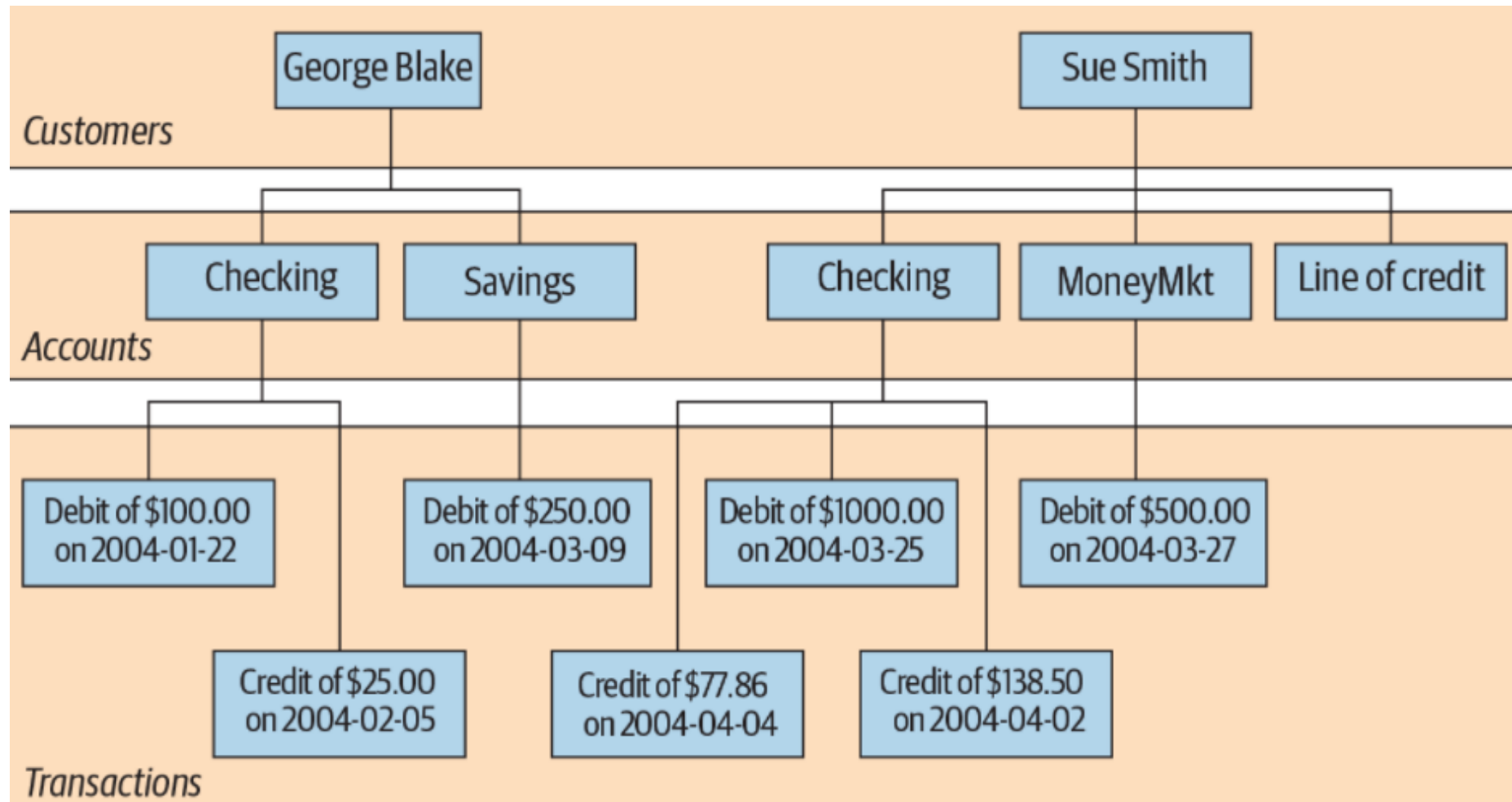
- Alan Beaulieu, Learning SQL, 3rd Edition
<https://learning.oreilly.com/library/view/learning-sql-3rd/9781492057604/>

Kas yra duomenų bazė?

- Duomenų bazė yra aibė susijusios informacijos.
- Pirmosios duomenų bazės buvo atspausdinti katalogai. Pvz.
 - Telefonų knygos
 - Ikea baldų katalogas

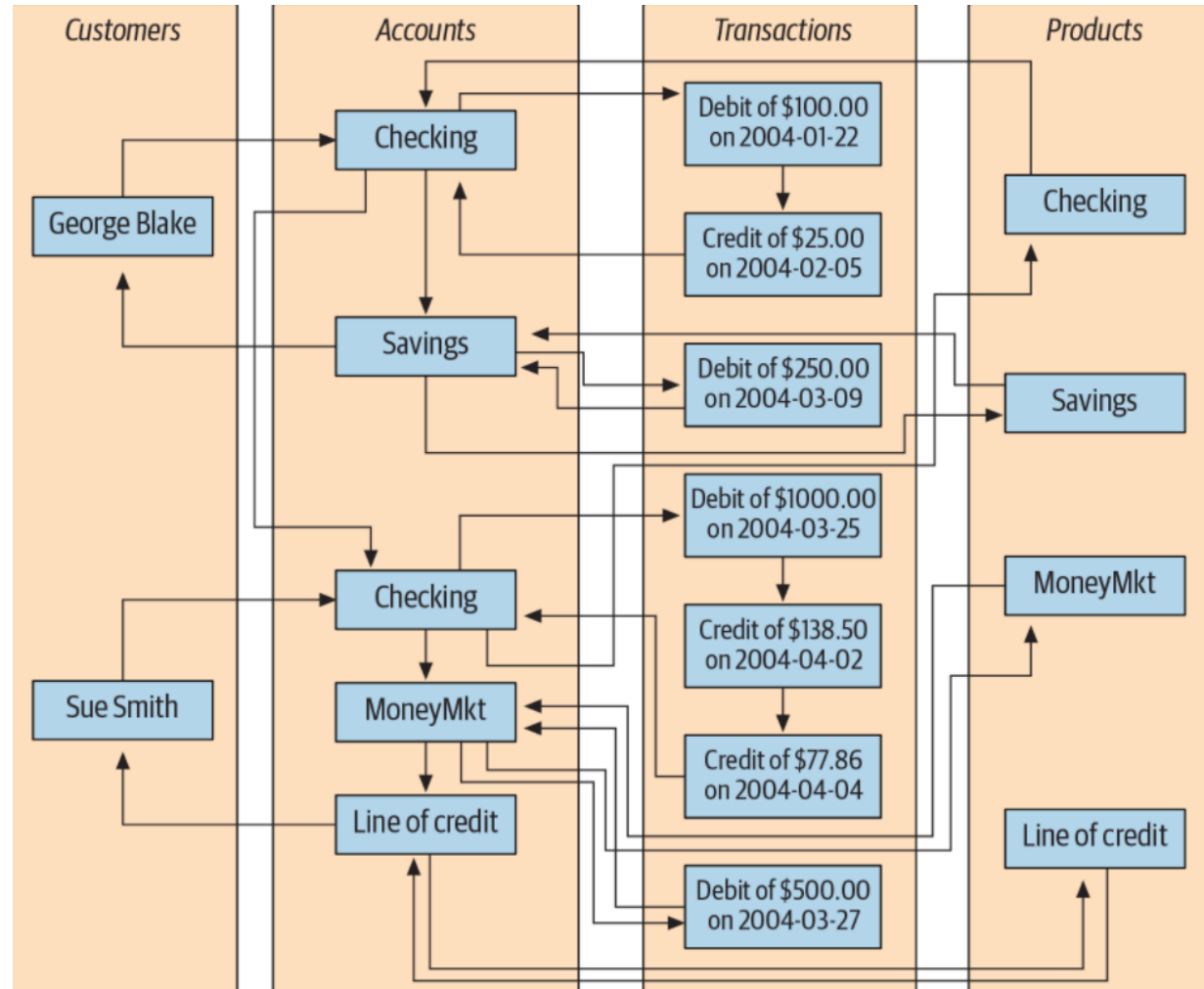
Duomenų bazės kompiuteriuose

- Hierarchinės



Duomenų bazės kompiuteriuose

- network database (tinklinės)



The Relational Model (Santykių modelis)

- Duomenys kaip lentelės buvo sukurti 1970, DR E.F. Codd IBM tyrimų centre.
- Originalius straipsnis *A Relational Model of Data for Large Shared Data Banks*

<https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>

Customer

cust_id	fname	lname
1	George	Blake
2	Sue	Smith

Account

account_id	product_cd	cust_id	balance
103	CHK	1	\$75.00
104	SAV	1	\$250.00
105	CHK	2	\$783.64
106	MM	2	\$500.00
107	LOC	2	0

Product

product_cd	name
CHK	Checking
SAV	Savings
MM	Money market
LOC	Line of credit

Transaction

txn_id	txn_type_cd	account_id	amount	date
978	DBT	103	\$100.00	2004-01-22
979	CDT	103	\$25.00	2004-02-05
980	DBT	104	\$250.00	2004-03-09
981	DBT	105	\$1000.00	2004-03-25
982	CDT	105	\$138.50	2004-04-02
983	CDT	105	\$77.86	2004-04-04
984	DBT	106	\$500.00	2004-03-27

Kas svarbu santykių modelyje

1. Kiekviena lentelės eilutė turi unikalų ID (Primary key)
2. Primary key turintis kelis laukus vadinamas *compound key*
3. Būtų galima naudoti raktą vardas+pavardė
 1. Trūkumai
 1. Pakeitus pavardę, reiktų keisti raktą
4. Stulpelio reikšmė rodanti į kitos lentelės Primary key yra vadinama Foreign key.

Normalizuotas duomenų modelis

- Normalizacijos tikslas – jeigu norime pakeisti informaciją – pvz. asmens pavardę, tai turime atlikti tik vienoje vietoje
- Vienas iš galimų reikalavimų. Stulpelių reikšmės turi būti smulkiausios galimos
 - Pvz. Adresas: Gedimino pr. 11, Vilnius, Lietuva, LT-01103
 - Gatvė, namo numeris, buto numeris, miestas, šalis, pašto kodas
- https://en.wikipedia.org/wiki/Database_normalization

1NF – kiekvienas stulpelis turi mažiausią galimą informaciją

- Pirminė lentelė

EMP_ID	EMP_NAME	EMP_PHONE
14	John	7272826385, 9064738238

- Normalizuota lentelė

EMP_ID	EMP_NAME	EMP_PHONE
14	John	7272826385
14	John	9064738238

2NF – ne raktiniai atributai yra priklausomi tik nuo pagrindinio rakto

- Pirminė lentelė
- Normalizuotos lentelės

TEACHER_ID	SUBJECT	TEACHER_AGE
25	Chemistry	30
25	Biology	30
47	English	35
83	Math	38
83	Computer	38

TEACHER_DETAIL table:

TEACHER_ID	TEACHER_AGE
25	30
47	35
83	38

TEACHER_SUBJECT table:

TEACHER_ID	SUBJECT
25	Chemistry
25	Biology
47	English
83	Math
83	Computer

3NF – nėra tranzityvių sąryšių

- Pirminė lentelė

EMP_ID	EMP_NAME	EMP_ZIP	EMP_STATE	EMP_CITY
222	Harry	201010	UP	Noida
333	Stephan	02228	US	Boston

- Normalizuotos lentelės

EMP_ID	EMP_NAME	EMP_ZIP
222	Harry	201010
333	Stephan	02228

EMP_ZIP	EMP_STATE	EMP_CITY
201010	UP	Noida
02228	US	Boston

4NF – nėra daugiareikšmių priklausomybių

- Pirminė lentelė

STU_ID	COURSE	HOBBY
21	Computer	Dancing
21	Math	Singing

- Normalizuotos lentelės

STU_ID	COURSE
21	Computer
21	Math

STU_ID	HOBBY
21	Dancing
21	Singing

Naudojami terminai

Term	Definition
Entity	Something of interest to the database user community. Examples include customers, parts, geographic locations, etc.
Column	An individual piece of data stored in a table.
Row	A set of columns that together completely describe an entity or some action on an entity. Also called a record.
Table	A set of rows, held either in memory (nonpersistent) or on permanent storage (persistent).
Result set	Another name for a nonpersistent table, generally the result of an SQL query.
Primary key	One or more columns that can be used as a unique identifier for each row in a table.
Foreign key	One or more columns that can be used together to identify a single row in another table.

Kas yra SQL?

- SQL yra programavimo kalba, kuri leidžia manipuluoti duomenimis, kurie yra lentelėse (relational model).

SQL kalba padalinta į dalis

- SQL schema statements – apibrėžia (sukuria) lenteles ir duomenų tipus jose.
- SQL data statements – leidžia manipuluoti duomenimis lentelėse.
- SQL transaction statements – naudojami užtikrinti, kad komandos yra įvykdytos sėkmingai, kitu atveju praneša, kad įvyko klaida.

Lentelēs kūrimas

```
CREATE TABLE corporation  
(corp_id SMALLINT,  
  name VARCHAR(30),  
  CONSTRAINT pk_corporation PRIMARY KEY (corp_id)  
);
```

Duomenų įdėjimas į lentelę

```
INSERT INTO corporation (corp_id, name)  
VALUES (27, 'Acme Paper Corporation');
```

Duomenų peržiūrėjimas

```
SELECT name  
FROM corporation  
WHERE corp_id = 27;
```

Ką galime daryti su duomenimis?

- Select
- Update
- Insert
- delete

Užklausos pavyzdys

```
SELECT t.txn_id, t.txn_type_cd, t.txn_date, t.amount
FROM individual i
  INNER JOIN account a ON i.cust_id = a.cust_id
  INNER JOIN product p ON p.product_cd = a.product_cd
  INNER JOIN transaction t ON t.account_id = a.account_id
WHERE i.fname = 'George' AND i.lname = 'Blake'
  AND p.name = 'checking account';
```

```
+-----+-----+-----+-----+
| txn_id | txn_type_cd | txn_date           | amount |
+-----+-----+-----+-----+
|      11 | DBT         | 2008-01-05 00:00:00 | 100.00 |
+-----+-----+-----+-----+
```

```
1 row in set (0.00 sec)
```