MY472 - Week 9
Relational Databases and SQL

Friedrich Geiecke

Outline

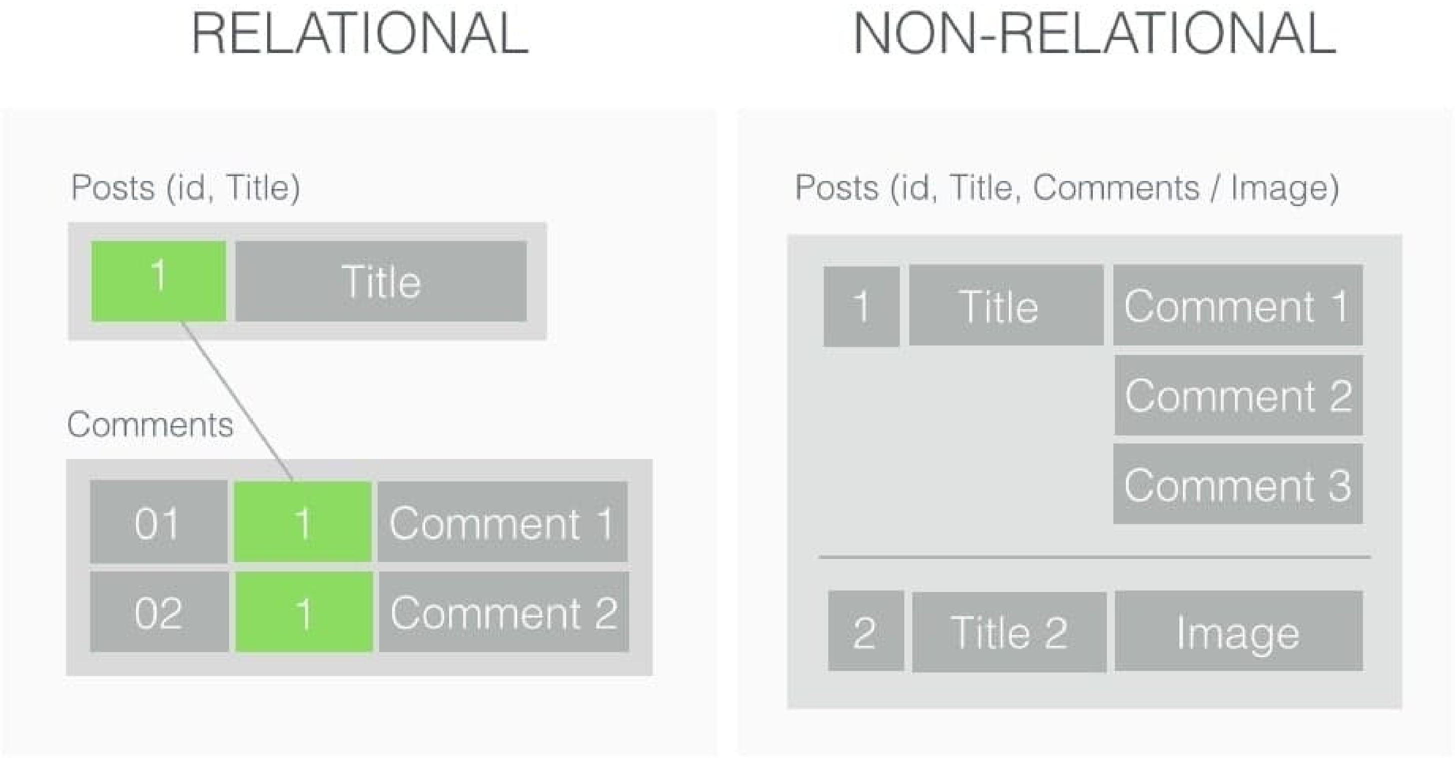
- **Relational** vs non-relational databases
- The SQ Language
- Coding session

Relational vs non-relational databases

Databases

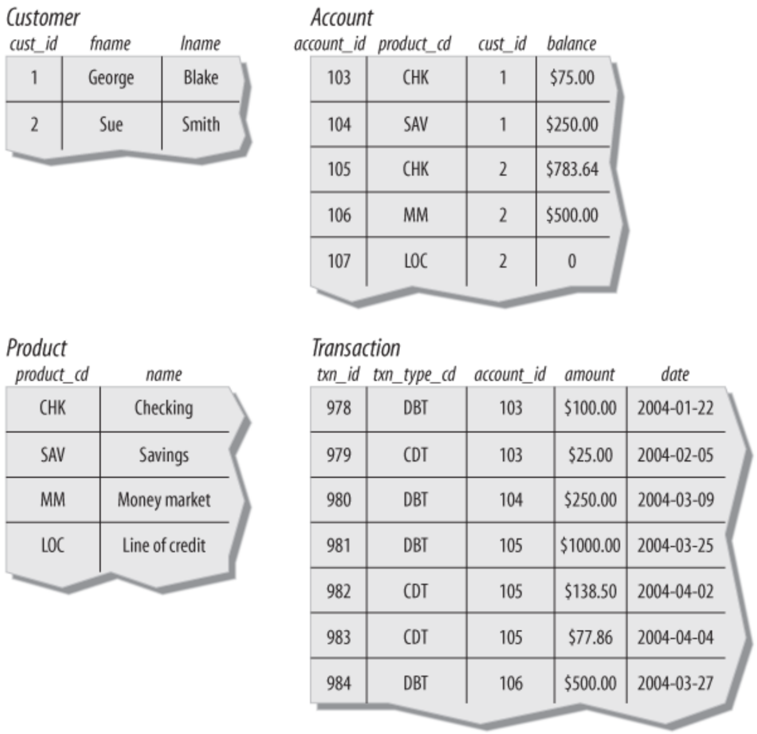
- **Database system**: An organized collection of data that is stored and accessed via a computer
- **Relational databases**: Data stored in multiple tables to avoid redundancy. Tables are linked based on common keys
- **Non-relational databases:** Data stored in a way that is not based on tabular relations (e.g. MongoDB uses JSON like documents)

Relational vs non-relational databases



From: Codewave Insights

Relational databases



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

- Relational database management systems (RDBMS): MySQL, PostgreSQL, SQLite, MariaDB, etc.
- Database as a Service (DBaaS): Amazon RDS, Google Cloud SQL, Microsoft Azure SQL Database
- DBaaS at a scale: Amazon RedShift, Google BigQuery, Microsoft Azure

Some vocabulary

Relational database term	SQL term
Relation	Table
Tuple, record	Row
Attribute, field	Column

Excerpt from: https://en.wikipedia.org/wiki/Relational_database

Keys

- Primary key: A column or set of columns (composite key) which uniquely identifies each row/record in the table
- Foreign key: A primary key of another table

Structured Query Language

SQL: Structured Query Language

- **Language** designed to define, control access to, manipulate, and query **relational databases**
- Initially written SEQUEL (Structured English Query Language), but later changed to SQL because of trademark issues
- Pronounced both S-Q-L and SEQUEL today
- It is a **nonprocedural/declarative language**: User defines what to do, inputs, and outputs, but not the control flow; how the statement is executed, is left to the *optimizer*
- How long SQL queries depends on optimization that is opaque to user
- Performance will vary, but generally faster than standard data frame manipulation in R (and much more scalable)

Some components of common SQL queries

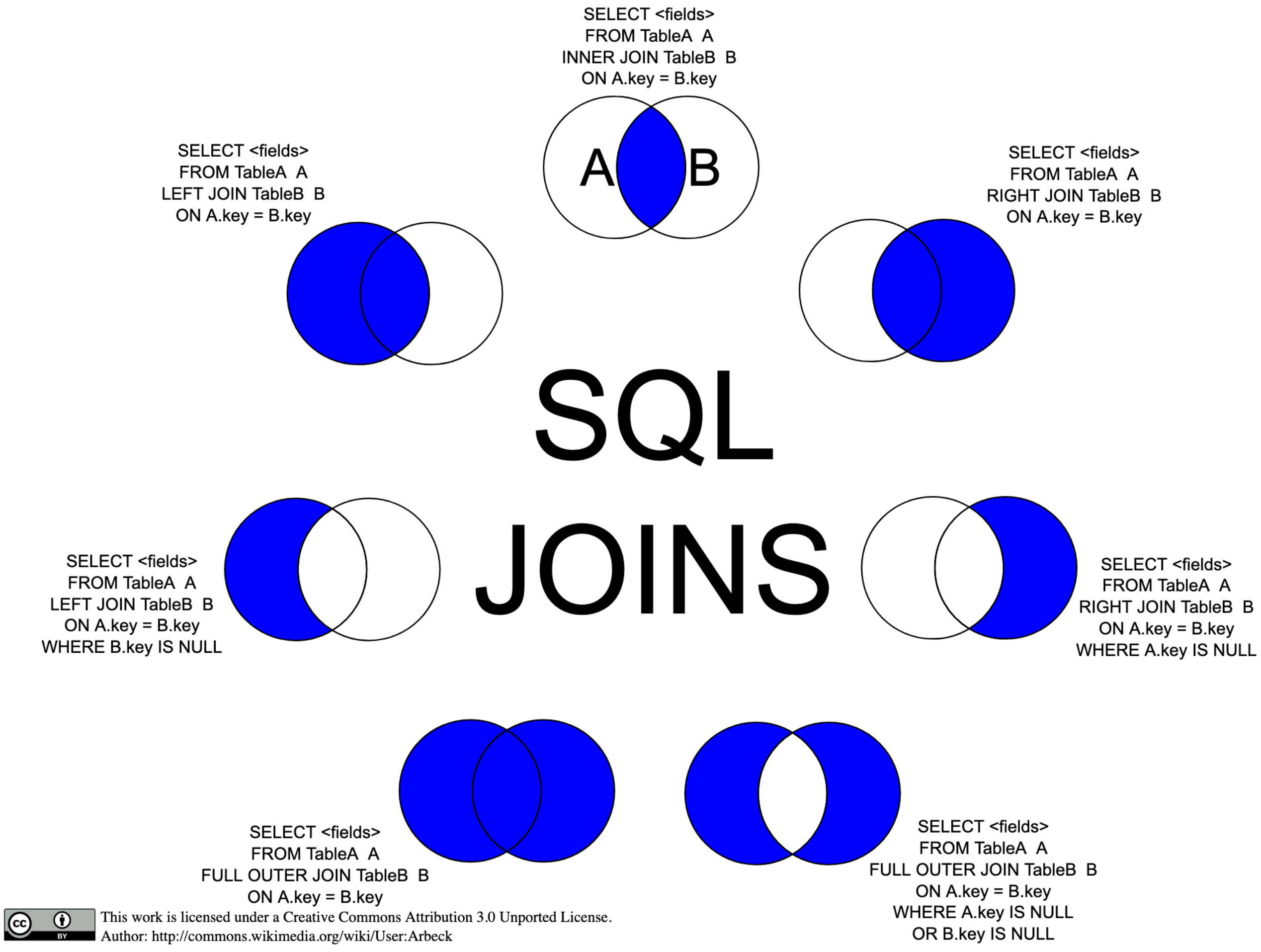
- The result of a SQL query is a table
- **SELECT** columns
- **FROM** a table in a database
- **WHERE** rows meet a condition
- **GROUP BY** values of a column
- **ORDER BY** values of a column when displaying results
- **LIMIT** to only X number of rows in resulting table
- Always required: **SELECT** and **FROM**; rest are optional
- **SELECT** can be combined with operators such as **SUM**, **COUNT**, **AVG**…
- To merge multiple tables, use **JOIN**

SQL query example

```
SELECT name, account_id FROM client;
```

```
SELECT * FROM client WHERE gender = 'F';
```

SQL JOINs



```
SELECT <fields>
FROM TableA A
INNER JOIN TableB B
ON A.key = B.key
```

```
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
```

```
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
```

```
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
WHERE B.key IS NULL
```

```
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL
```

```
SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key
```

```
SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL
```

From: https://upload.wikimedia.org/wikipedia/commons/9/9d/SQL_Joins.svg

SQL JOIN example

```
SELECT client.name, account.balance
FROM client JOIN account
ON client.account_id = account.id;
```

Coding session

Coding session

- See `01-sql-intro.Rmd`
- See `02-sql-join-and-aggregation.Rmd`

General information on how to connect to SQL databases with R: https://db.rstudio.com/getting-started/