MY472 - Week 10: Creating and Managing Databases

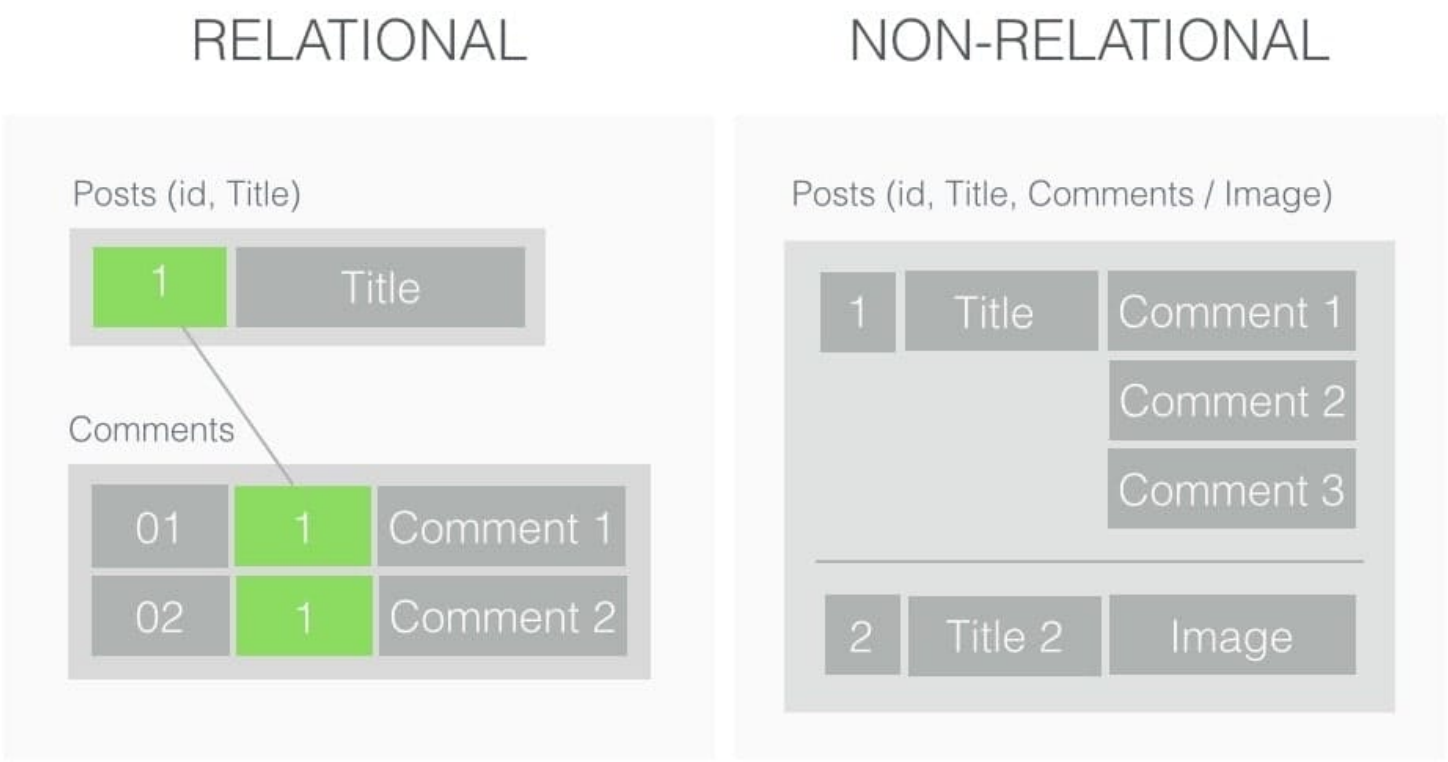
Outline

- Database systems
- **Relational** vs. non-relational databases
- SQL language
- Example
    - Querying public Facebook data

Databases

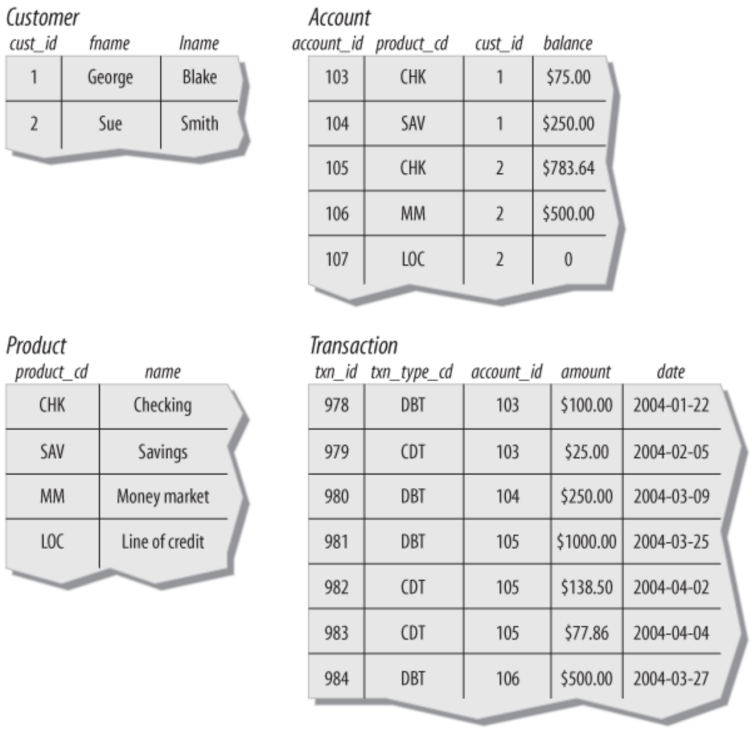
- **Database system**: an organized collection of data that is stored and accessed via a computer
- **Relational databases**: data stored in tables that are linked based on common keys (to avoid redundancy)
- **Non-relational databases:** data stored in a way that is not based on tabular relations (e.g.,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

- Software: MySQL, PostgreSQL, SQLite, MariaDB, etc.
- DBaaS: Amazon RDS, Google Cloud SQL, Microsoft Azure SQL Database
- DBaaS at a scale: Amazon RedShift, Google Big Query, Microsoft Azure

SQL

- **S**tructured **Q**uery **L**anguage; pronounced S-Q-L or SEQUEL
- Language designed to define, control access to, manipulate, and query relational databases
- It's a **nonprocedural/declarative language**: define inputs and outputs; how the statement is executed is left to the *optimizer*
- How long SQL queries depends on optimization that is opaque to user (which is great!)
- Performance will vary, but generally faster than standard data frame manipulation in R (and much more scalable)

Components of a SQL query

- The result of a SQL query is always 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 mutliple tables, use **JOIN**

SQL query example

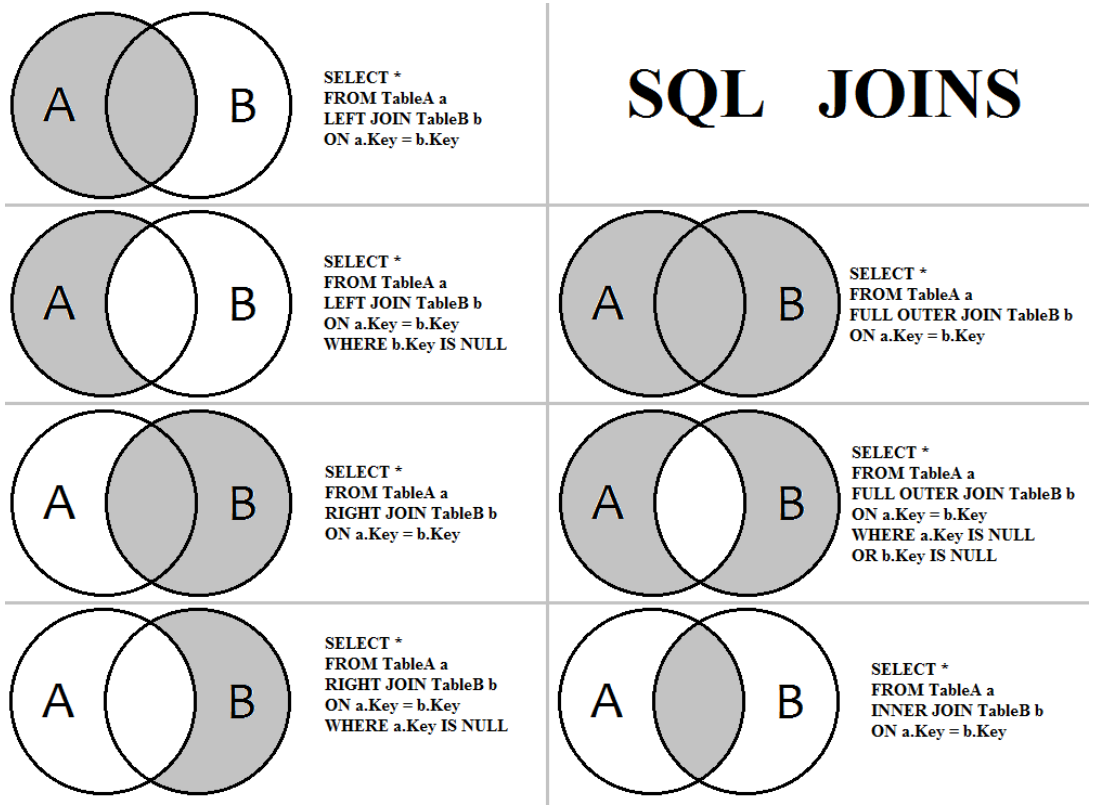
```
SELECT name, account_id FROM client;
```

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

SQL JOIN example

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

SQL JOINs


SQL JOINS
A
B
SELECT *
FROM TableA a
LEFT JOIN TableB b
ON a.Key = b.Key
SELECT *
FROM TableA a
LEFT JOIN TableB b
ON a.Key = b.Key
WHERE b.Key IS NULL
SELECT *
FROM TableA a
FULL OUTER JOIN TableB b
ON a.Key = b.Key
SELECT *
FROM TableA a
RIGHT JOIN TableB b
ON a.Key = b.Key
SELECT *
FROM TableA a
FULL OUTER JOIN TableB b
ON a.Key = b.Key
WHERE a.Key IS NULL
OR b.Key IS NULL
SELECT *
FROM TableA a
RIGHT JOIN TableB b
ON a.Key = b.Key
WHERE a.Key IS NULL
SELECT *
FROM TableA a
INNER JOIN TableB b
ON a.Key = b.Key

Example: Using SQL to query public Facebook data

- See `01-sql-intro.Rmd`
- See `02-sql-advanced.Rmd`