Software Architecture & Databases

Reicman University

Contact Me

amitm

@

at-bay.com

Feel free to drop me your comments, suggestions, questions

Nice to Meet You!

- Life: 45 yo, married + 3 kids, live in Shoham, Yoga.
- Current: 2 yrs @ At-Bay as a Software Architect (...what's that?)
- Overall: ~15 yrs in the industry, mostly in startups, but fallen a couple of times to the corporate hands
- Fun time 1: Does Tech Due Diligences for a couple of VCs
- Fun time 2: Contributor to the Dapr open source project
- Fun time 3: this thing

What We'll Talk About

- Software architecture - theory
- Where theory meets reality
- The oh-so-boring SaaS company
- The *relational database* as the greatest invention in industrial software engineering
- Where the relational database *fails*
- Q & A

Software Architecture

Wikipedia

- "Software architecture refers to the fundamental structures of a software system and the discipline of creating such structures and systems."
- "Each structure comprises software elements, relations among them, and properties of both elements and relations."
- system:
    - structures:
        - element <-- relation --> element

Software Architecture

Somebody Needs to Choose... ("System Analysis")

- "Software architecture is about making fundamental structural **choices** that are costly to change once implemented" (wikipedia, this time getting it right)
- What's costly to change (over time):
  - Programming Language (changing/adding mostly breaks common tools) --> "element"
  - Data Model (breaks everything if done wrong) --> "property of an element"
  - Runtime Environment (on-prem vs cloud x/y/z) --> "structure+system"
  - Contracts (APIs) --> "relations"

Software Architecture

Conceptualization & Creation of New Things ("System Design")

- "Fighting" the windmills of **complexity** as software obeys the 2nd law of thermodynamics (complexity never decreases)
- But what is complex? ("This part of the codebase doesn’t *feel* right!")
- When do you surrender to the evil called "management"?
- The humanity! (You care about people not losing their minds)

Software Architecture

Reviewing Things ("Evaluation")

- High Level Design Review - system
- Code Reviews - quality
- Security Reviews - security safety
- Test Review - code safety
- Data Modeling Review - correct abstraction/database performance
- Data Integration Review - none applicative systems
- DevOps Review - environment/deployment
- SRE Review - gauges

Make sure stuff alines with your company *vision* and *business status* on how software is being made

Software Architecture

Redesign Bad Stuff Other People Did ("Evolution")

- Fun/easy stuff is to do *NEW* things
- NotSoFun/hardest stuff is to evolve *OLD* things
- Read about: "evolutionary architecture"

Ope There Goes Gravity (E. minem)

- The ideal picture is a fantasy
- You have to **surrender** to accepting these:
  - You and your work are **not eternal**
  - There is no ″generic″ best architecture
  - Even the "best" code is eventually **thrown** to the recycle bean
  - You are working for a company that in its essence is a machine made to make **profit**
  - At the broader perspective, *individual contributions*, are **irrelevant**. Collaboration is key.
- None of the above is in contradiction to software development being a beautiful human mental act

The Oh-So-Boring SaaS Company

- Capture *Data*
- Do something over *Data*
- Arrange the *Data* nicely
- Ask for money from customers/advertisers

Wow! They finally redesigned whatsapp for iOS 7! 9:10 AM

Yep, I hope it now looks at least just as good as the BB10 version 😆 9:12 AM

I have to say it's one of the few iOS 7 redesigns that I actually like 9:14 AM

Instagram

Discovering — and telling — stories from around the world. Curated by Instagram's community team.

blog.instagram.com/

Followed by pagesandplaces, watchthatbook, literaryrage + 146 more

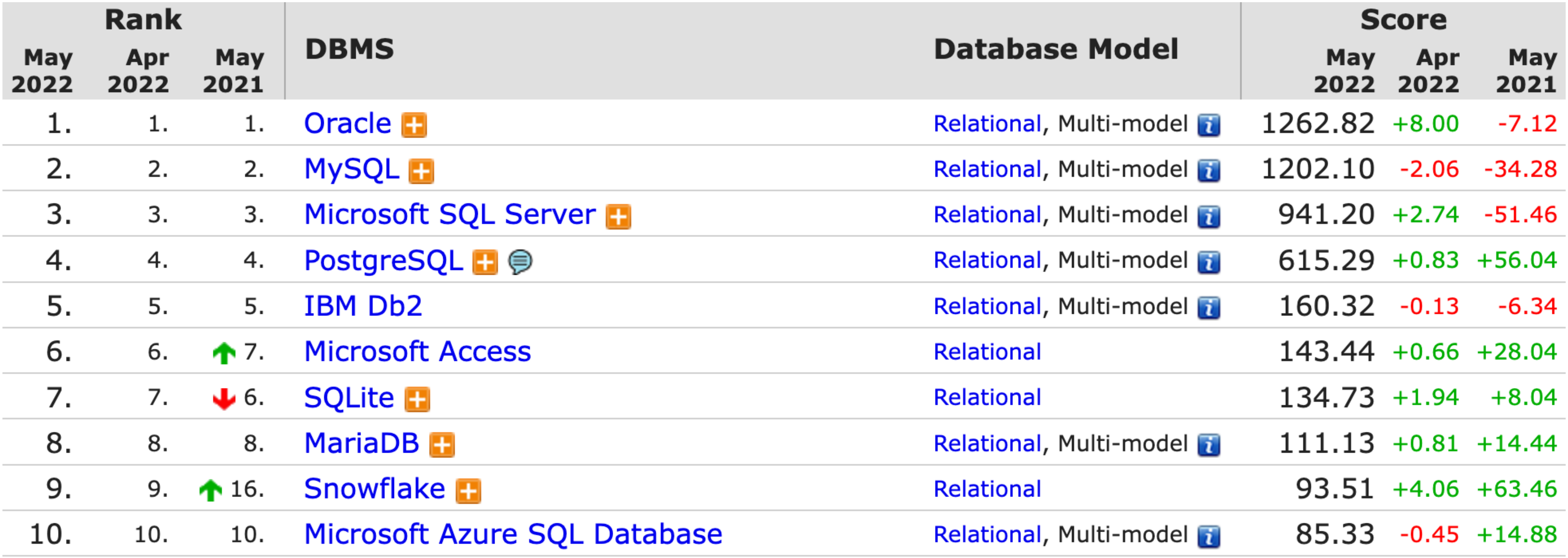
The Data Model

Data Model = Database Tables

- It is how you model the business (entities)
- **It is the single most important element in the system design of a SaaS**
- It is the only effort worth investing BEFORE writing a single line of code (atypical for a startup to do)
- Still, it is ever **evolving** with the business needs, but once in use, *it is hard to change*
- Things that aren't persisted are potentially lost, so the use of in-memory data structures (with all due respect to BigO) is just temporary and the **significant stateful operations are done over a database**

Data Model - Implementation

Usually, we're talking about a *Relational Database*

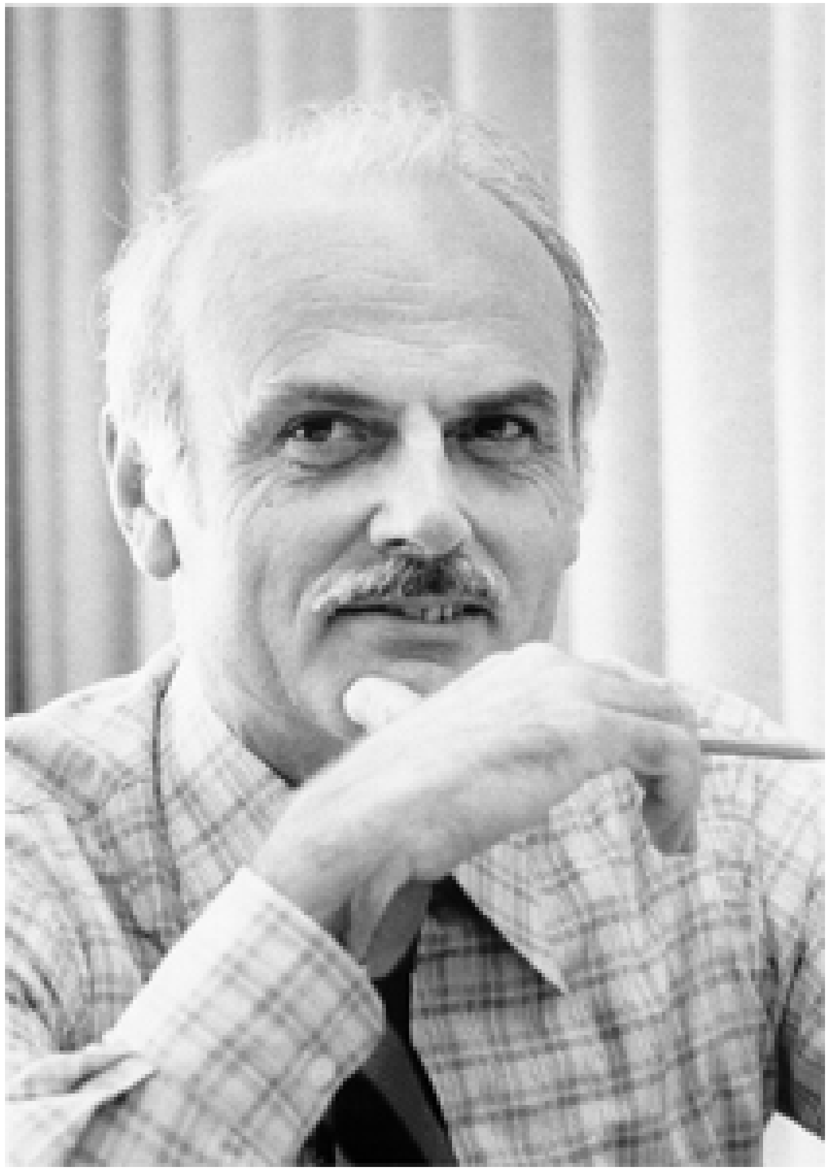


Rank			DBMS	Database Model	Score		
May 2022	Apr 2022	May 2021			May 2022	Apr 2022	May 2021
1.	1.	1.	Oracle	Relational, Multi-model	1262.82	+8.00	-7.12
2.	2.	2.	MySQL	Relational, Multi-model	1202.10	-2.06	-34.28
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	941.20	+2.74	-51.46
4.	4.	4.	PostgreSQL	Relational, Multi-model	615.29	+0.83	+56.04
5.	5.	5.	IBM Db2	Relational, Multi-model	160.32	-0.13	-6.34
6.	6.	↑ 7.	Microsoft Access	Relational	143.44	+0.66	+28.04
7.	7.	↓ 6.	SQLite	Relational	134.73	+1.94	+8.04
8.	8.	8.	MariaDB	Relational, Multi-model	111.13	+0.81	+14.44
9.	9.	↑ 16.	Snowflake	Relational	93.51	+4.06	+63.46
10.	10.	10.	Microsoft Azure SQL Database	Relational, Multi-model	85.33	-0.45	+14.88

Relational Databases

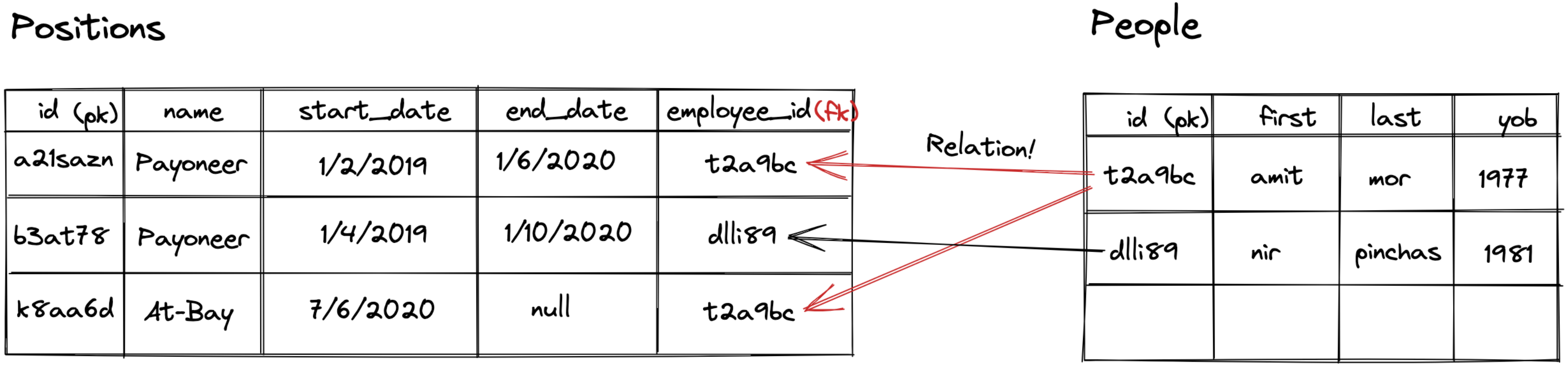
The founding father of the relational model is *Edgar F. Codd*

- Based on his work published in 1970 (yes, 52 yo technology) while he was working for IBM
- He won the Turing Award in 1981 for this work
- He applied Relational Algebra and proposed such an algebra as a basis for database query languages
- Five primitive operators: selection, projection, Cartesian product (also called cross join), set union, set difference.



Relational Databases - Basics

- Relational databases are all about **tables**
- Tables are able to *relate* to one another



Positions

id (pk)	name	start_date	end_date	employee_id (fk)
a21sazn	Payoneer	1/2/2019	1/6/2020	t2a9bc
b3at78	Payoneer	1/4/2019	1/10/2020	dlli89
k8aa6d	At-Bay	7/6/2020	null	t2a9bc

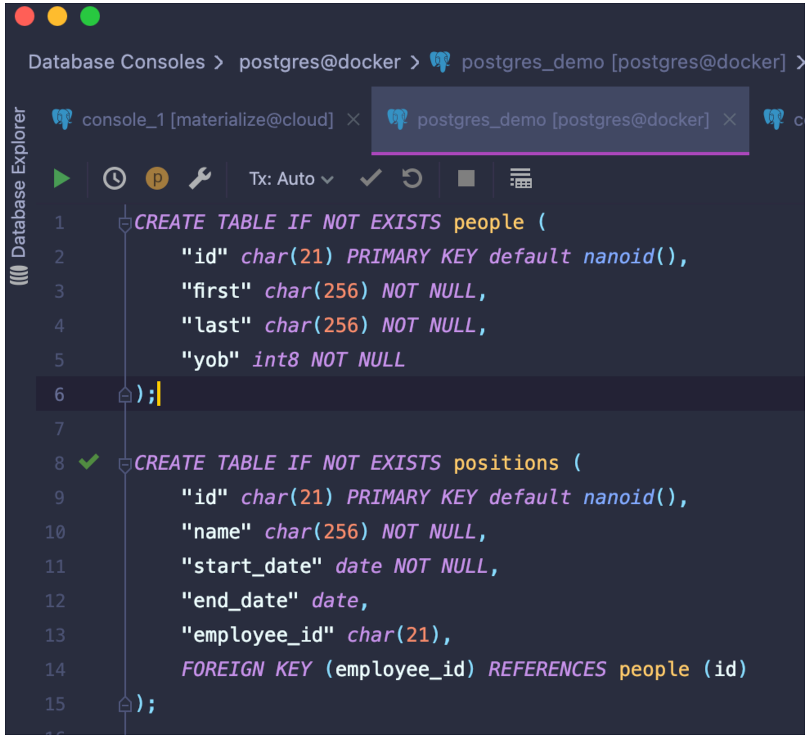
People

id (pk)	first	last	yob
t2a9bc	amit	mor	1977
dlli89	nir	pinchas	1981

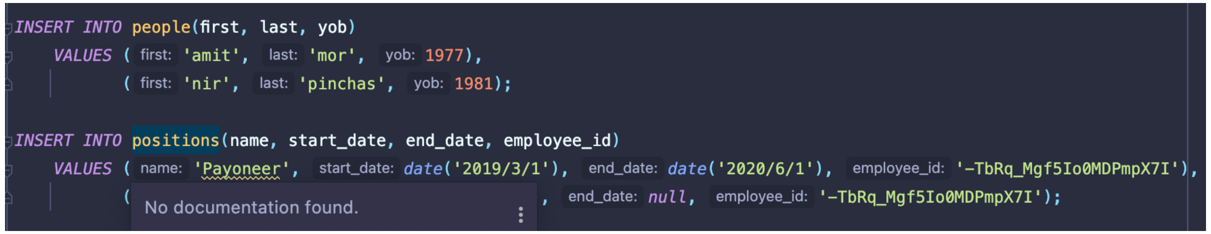
Relational Databases - Basics 2

- Use a **declarative** language to apply operations over the logical representation and don't mind the physical aspects
- Ability to create meaningful information by **joining** of tables
- To ensure that data is always accurate and accessible, relational databases follow certain integrity rules (**A.C.I.D**)
- Relational databases are **transactional**—they guarantee the state of the entire system is consistent at any moment
- The relational model means that the *logical data structures*—the data tables, views, and indexes—are separate from the *physical storage structures*

Relational Databases -
Declarative Language

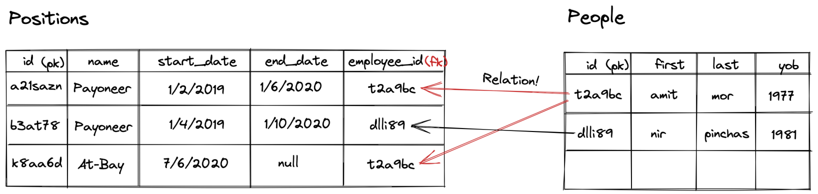

Database Consoles > postgres@docker > postgres_demo [postgres@docker] >
console_1 [materialize@cloud]
postgres_demo [postgres@docker]
Tx: Auto
Database Explorer

```sql
CREATE TABLE IF NOT EXISTS people (
    "id" char(21) PRIMARY KEY default nanoid(),
    "first" char(256) NOT NULL,
    "last" char(256) NOT NULL,
    "yob" int8 NOT NULL
);

CREATE TABLE IF NOT EXISTS positions (
    "id" char(21) PRIMARY KEY default nanoid(),
    "name" char(256) NOT NULL,
    "start_date" date NOT NULL,
    "end_date" date,
    "employee_id" char(21),
    FOREIGN KEY (employee_id) REFERENCES people (id)
);
```



```sql
INSERT INTO people(first, last, yob)
    VALUES ( first: 'amit',  last: 'mor',  yob: 1977),
           ( first: 'nir',  last: 'pinchas',  yob: 1981);

INSERT INTO positions(name, start_date, end_date, employee_id)
    VALUES ( name: 'Payoneer',  start_date: date('2019/3/1'),  end_date: date('2020/6/1'),  employee_id: '-TbRq_Mgf5Io0MDPmpX7I'),
           (                                              ,  end_date: null,  employee_id: '-TbRq_Mgf5Io0MDPmpX7I');
```

No documentation found.



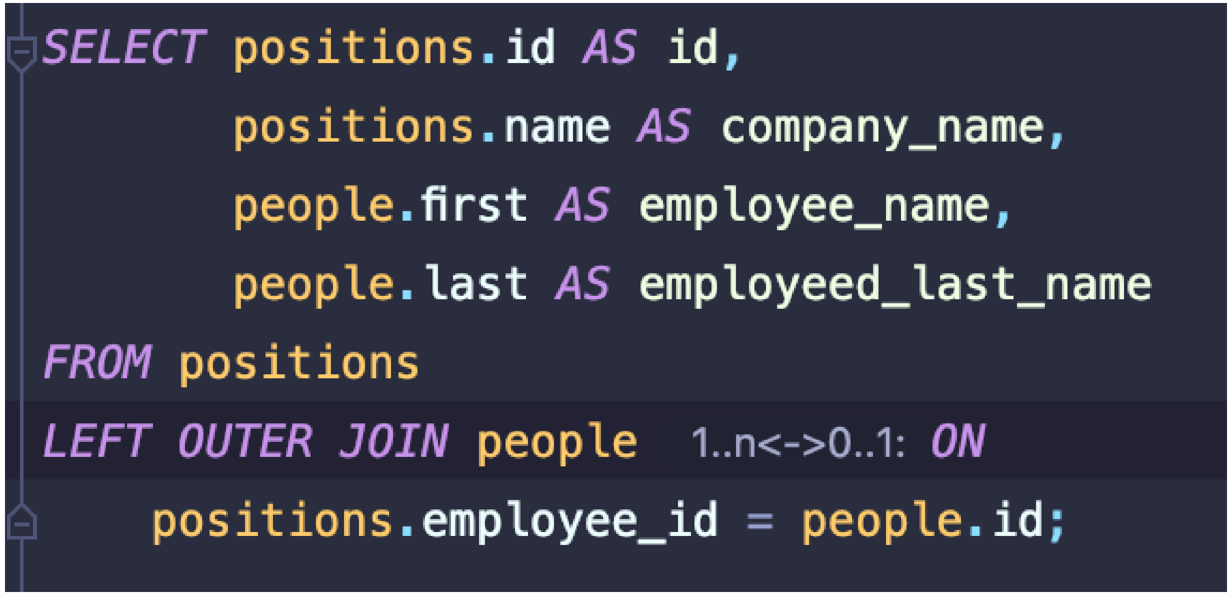
Positions

id (pk)	name	start_date	end_date	employee_id(fk)
a21sazn	Payoneer	1/2/2019	1/6/2020	t2a9bc
b3at78	Payoneer	1/4/2019	1/10/2020	dlli89
k8aa6d	At-Bay	7/6/2020	null	t2a9bc

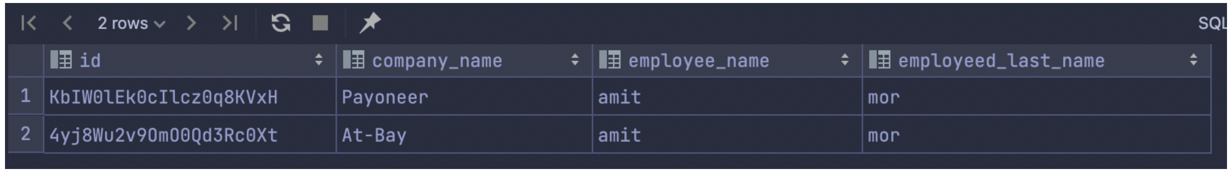
People

id (pk)	first	last	yob
t2a9bc	amit	mor	1977
dlli89	nir	pinchas	1981

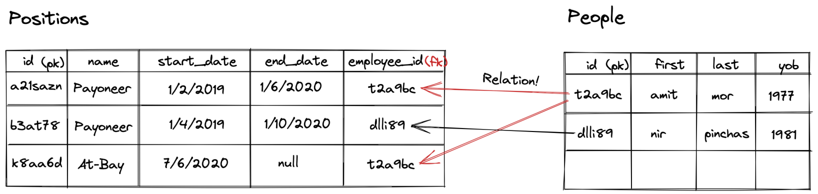
Relational Databases - JOINS



```
SELECT positions.id AS id,
       positions.name AS company_name,
       people.first AS employee_name,
       people.last AS employeed_last_name
FROM positions
LEFT OUTER JOIN people  1..n<->0..1: ON
    positions.employee_id = people.id;
```



	id	company_name	employee_name	employeed_last_name
1	KbIW0lEk0cIlcz0q8KVxH	Payoneer	amit	mor
2	4yj8Wu2v9Om0OQd3Rc0Xt	At-Bay	amit	mor



Positions

id (pk)	name	start_date	end_date	employee_id(fk)
a21sazn	Payoneer	1/2/2019	1/6/2020	t2a9bc
b3at78	Payoneer	1/4/2019	1/10/2020	dlli89
k8aa6d	At-Bay	7/6/2020	null	t2a9bc

People

id (pk)	first	last	yob
t2a9bc	amit	mor	1977
dlli89	nir	pinchas	1981

Relational Databases - Transactions & A.C.I.D

ACID = Atomicity, Consistency, Isolation, Durability

Atomicity - a guarantee that either all of the transaction succeeds or none of it does. “all or nothing”.
Consistency - All data will be valid according to all defined rules, including any constraints, cascades, and triggers that have been applied on the database.
Isolation - a transaction cannot read data from any other transaction that has not yet completed.
Durability - once a transaction is committed, it will remain in the system – even if there’s a system crash immediately following the transaction

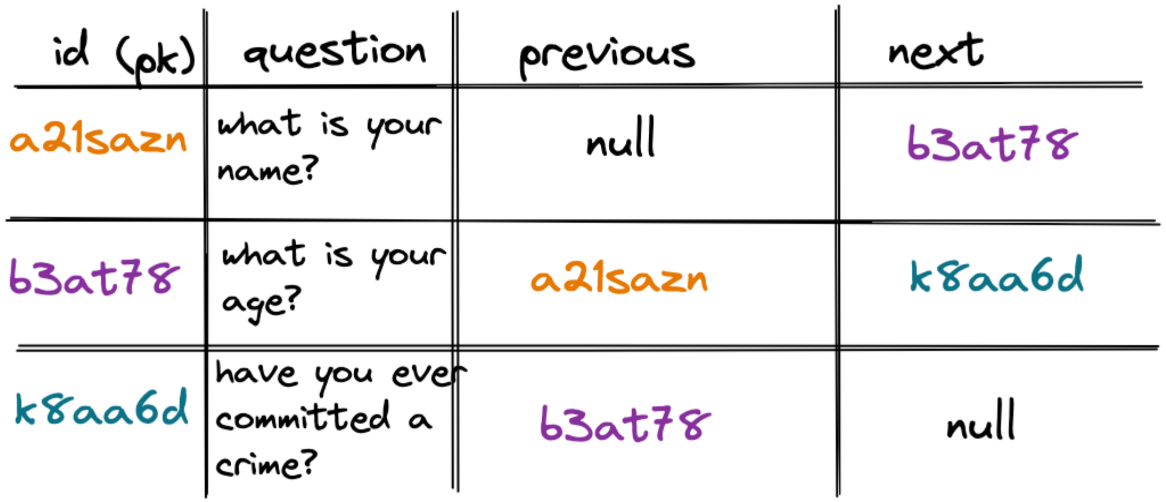
Relational Databases - In Practice

- You'd rarely find a company using "raw" SQL queries as it is considered error prone
- Most likely a company would use an ORM framework of some sorts
- ORM - Object (to) Relational Mapping - Libraries that abstract away the SQL expressions and allow to use a more programmatic access to data
  - Python - Django, SQLAlchemy
  - Java - Hibernate
- Python Django (similar to the left join demonstrated above):

```
Positions.objects.filter(people__id='KbIW0l..')
```

When the Relational Model Breaks

Questions

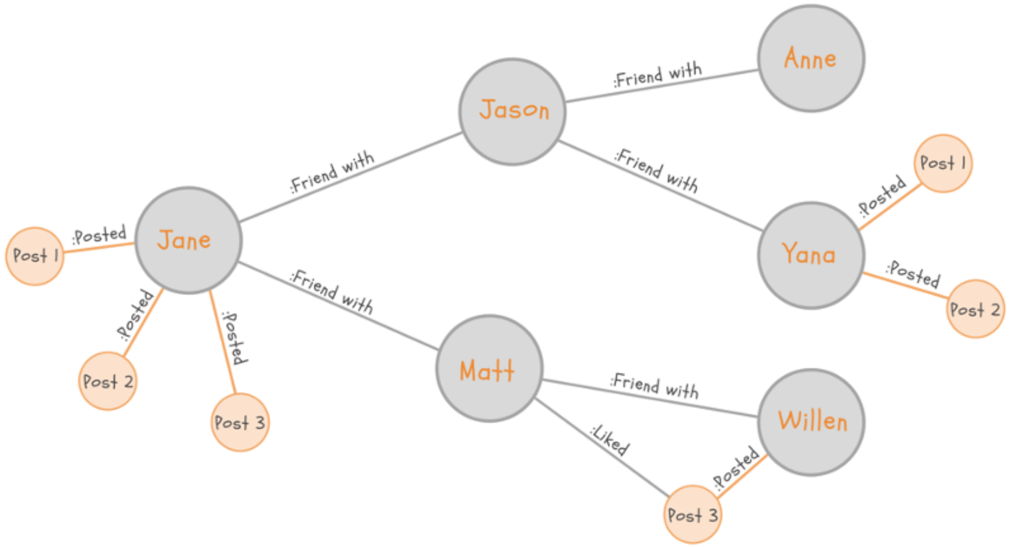


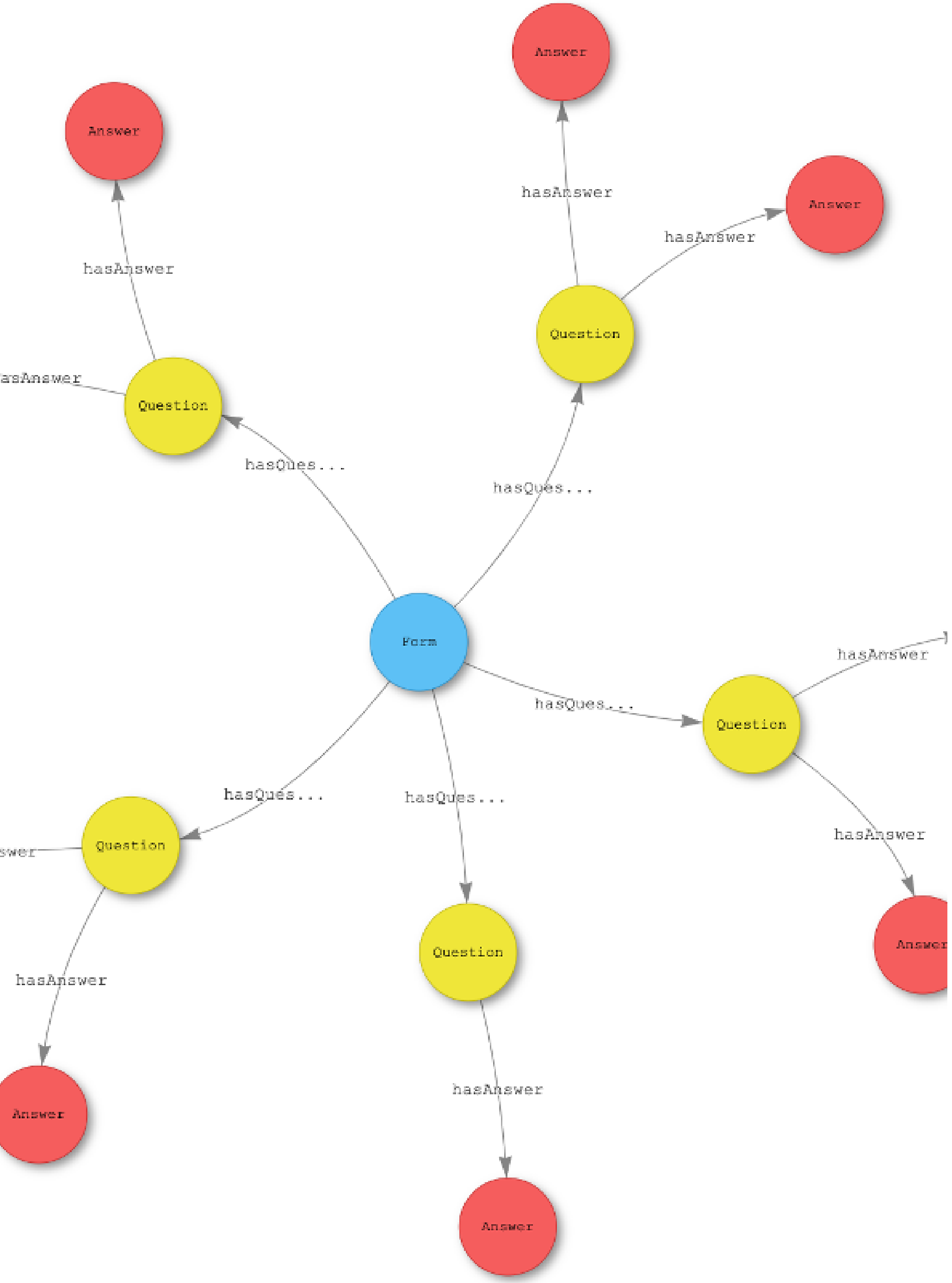
id (pk)	question	previous	next
a21sazn	what is your name?	null	b3at78
b3at78	what is your age?	a21sazn	k8aa6d
k8aa6d	have you ever committed a crime?	b3at78	null

Linked list in a relational database

Graph Databases

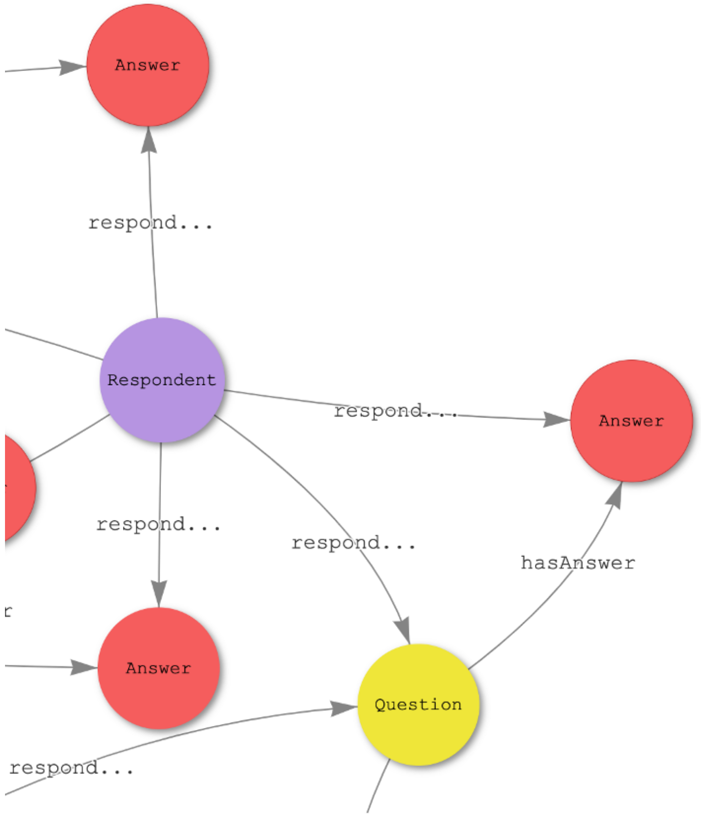
- Databases that uses **graph structures** to store data. The graph is build by nodes and edges (relationship).
- Common query languages are: Cypher, Gremlin and SparQL.
- Offers ACID transaction guarantees
- Schema-less
- Excel at recursive, graph data structures



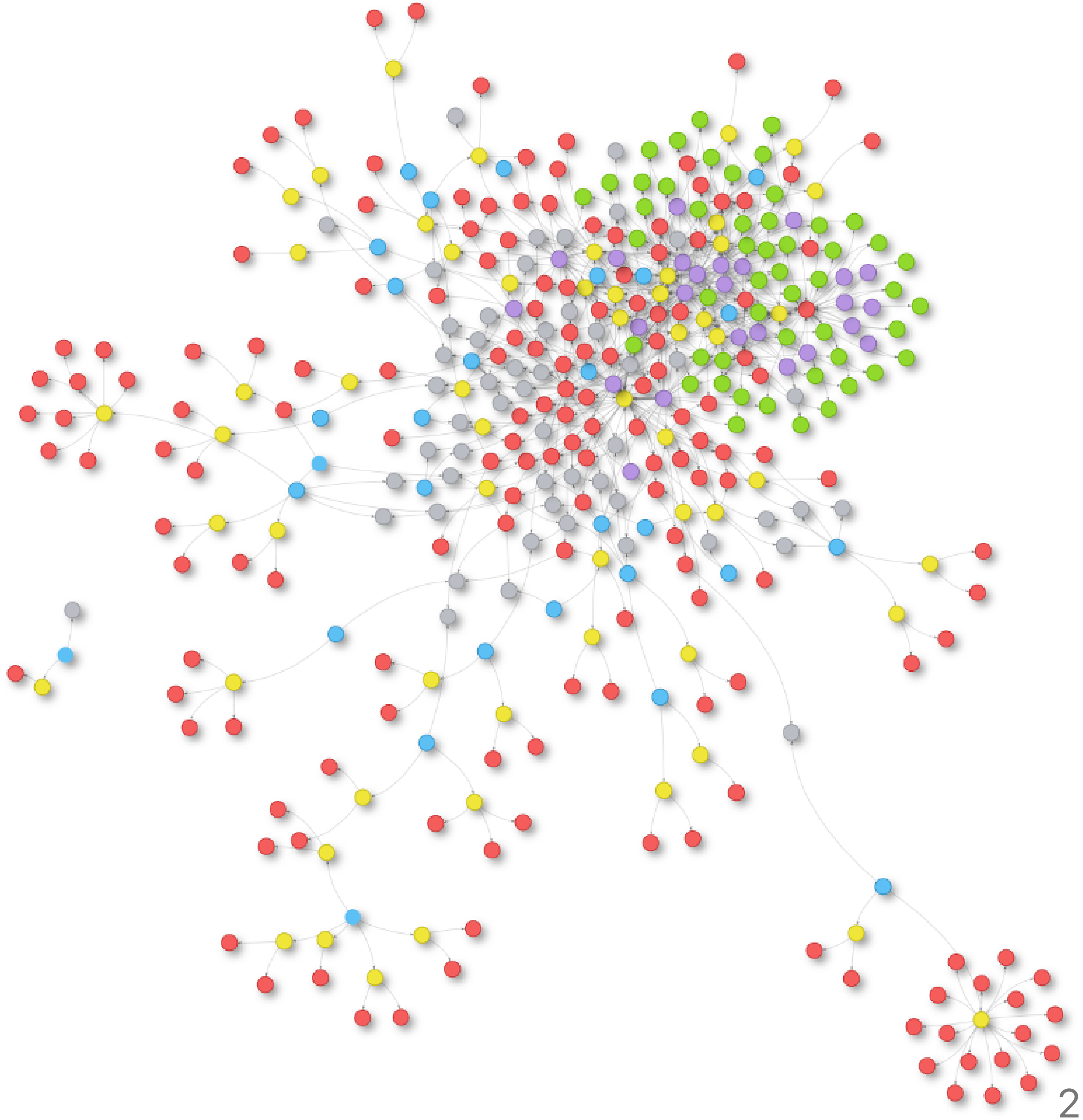


Modeling Questions on a Graph Database

Modeling a Respondent



That's a Questionnaire



At-Bay

- An insure-tech startup (5 yo)
- Raised > 200M$; Evaluated > 1.3B$
- x3 people in 2 years, 76 people in tlv
- working hybrid, people from north, south and center
- very diverse and heterogeneous skills are around
- best phase to do interesting things (got the money, customers and scaling the tech)
- trying to do good
- looking for back/front/full-end devs,cyber researchers, data/ml engineers

Literature

- wikipedia
- http://users.ece.utexas.edu/~perry/work/papers/swa-sen.pdf
- https://www.goodreads.com/book/show/296981.Object_Oriented_Software_Engineering
- https://thevaluable.dev/fighting-software-entropy/
- E.F. Codd paper