Founder of CS? Alan Turing
- Turing award (Nobel prize in CS equiv.)

Charles Bachman: Invented DBMS
1980 Edgar Codd: Relational model invention
Jim Gray: Transaction
Mike Stonebraker: Postgres

Fundamental Functionality of DBMS

① Persistence of data (similar to a filesystem)
② Fast access to data (unlike filesys)
(indexing)
③ Support for a data model suitable for a class of apps. (more soon) *
④ QL and programming interface *
⑤ Transaction management (e.g. ACID) *
⑥ Security of data and authorization

What's a data model? (DM)

Informal definition:
(i) an abstract (mathematical) highlevel notation for describing the structure of application data
(ii) a set of logical operators apps can use to manipulate the data (e.g. derive new data)

Example data model: Relational model (Edgar Codd, 1980)

① Data consists of "relations" or tables with attributes (columns) and a set of unordered rows (tuples)

② Relational operators: selection (σ)
projection (Π)
join ($\bowtie$)
group by and aggregate

e.g. Fin. Transactions Table

AcctFrom	AcctTo	Type	Amount	Date
Acct1	Acct2	Transfer	500	Jan 1
Acct2	Acct3	Transfer	300	〃
Acct2	ATM	Withdraw	100	〃
ATM	Acct4	Deposit	200	〃

Accounts Table

AcctId	OwnerName	Balance
Acct1	Alice	500
⋮	⋮	⋮

Why is a data model useful? It gives users (app. developers) a view of their data that is higher level than bits (→ data independence)

- Also allows them to understand the querying functionalities provided by the DBMS

3 Fundamental Components of DBMS

CS448

(1) DM	(2) QL	(3) System
Relational	SQL	Postgres, MySQL, Oracle DB, IBM DB2
Property Graph	Cypher, PGQL	Neo4j, DGraph, Oracle Spatial
XML	XPath	Mark Logic
RDF	Sparql	Apache Jena

* Relational model is suitable for a very large class of application

e.g. (suitable): Bank transactions

Suppose Alice withdraws $50 from an ATM

2 very simple queries:

1. INSERT INTO FinTransactions VALUES (Acct1, ATM1, Withdraw, 50, ...)

2. UPDATE Accounts SET balance = (SELECT balance FROM Accounts WHERE AcctId = Acct1) - 50 WHERE AcctId = Acct1

Trans 1	Trans 2
read bal = 1000	
	read bal = 1000
write = 950	
	write = 1050

t ↓

t → 1050 (wrong)

Transaction Concurrency Control

e.g. (unsuitable) Money Laundering / Fraud Detection

one classic pattern: cyclic transfers

4 step	Acct 1	transfers	Acct 2
	Acct 2	transfers	Acct 3
	Acct 3	transfers	Acct 4
	Acct 4	transfers	Acct 1 (launders the $)

Sometimes much longer (10-15 steps). Unknown to the bank. (Often we don't know length.)

For 4-length cycle:

```
SELECT T1.AcctFrom
FROM FT as T1,
     FT as T2,
     FT as T3,
     FT as T4
WHERE T1.AcctTo = T2.AcctFrom
  AND T2.AcctTo = T3.AcctFrom
  AND T2.AcctTo = T4.AcctFrom
```

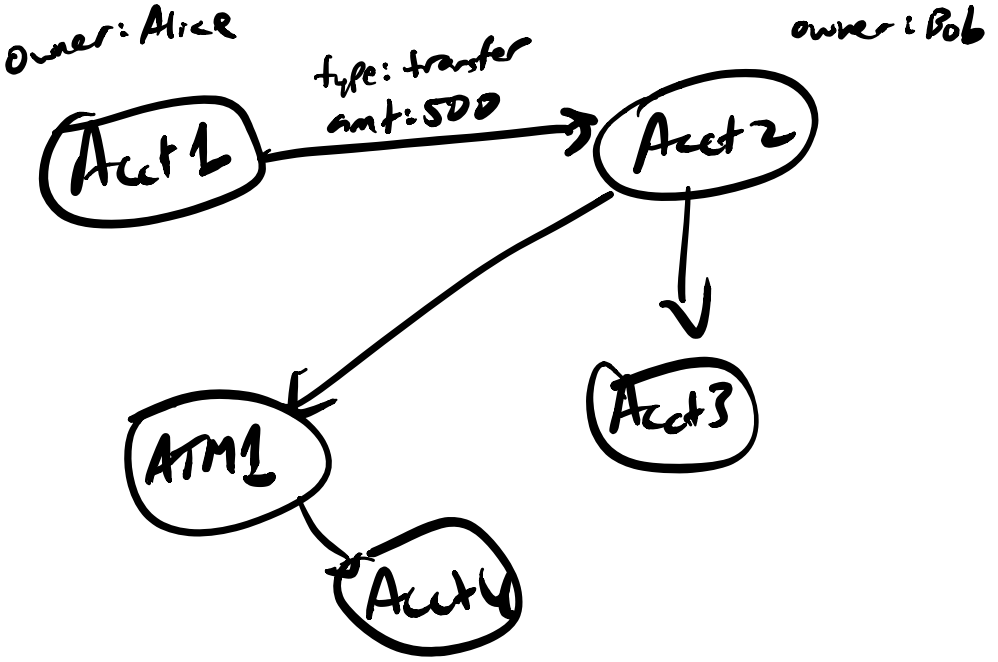
AND T4.AcctTo = T1.AcctFrom

What if cycle length is unknown?
(Not an easy way.)

Alternative Data Model: Property Graph

1. Data is a graph with arbitrary attributes on nodes and edges. Specifically,
   a) Nodes: V
   b) Edges: E
   c) Arbitrary attributes on V and E (arb. key-value pairs)

2. Ops: Basic traversal operators and path finding



Finding cycles in neo4j's Cypher language:

4-length cycle

MATCH (a) -[transfer]→ (b) -[transfer]→ (c) -[transfer]
→ (a)
RETURN a.id

Variable length
MATCH (a) -[:transfer 2..10]→ (a)
RETURN a.ID

Charles Bachman (GE engineer, ca. 1961-62)
invented databases.

1970 - Codd's Rel'n Model
1973 - Bachman won Turing award (IDS)
- not relational - network data model (close to prop graph model)
1980s - First Rsystems built