The Relational Model

Abdu Alawini

University of Illinois at Urbana-Champaign

CS411: Database Systems

Some slides are adopted from D. Maier, S. Davidson and A. Parameswaran

Learning Objectives

After this lecture, you should be able to:

- Define a **data model**
- Define the **relational data model**
- Articulate the **basic terminologies** of the **relational** data **model** (from a practical perspective)
- Define **Primary and Foreign keys**

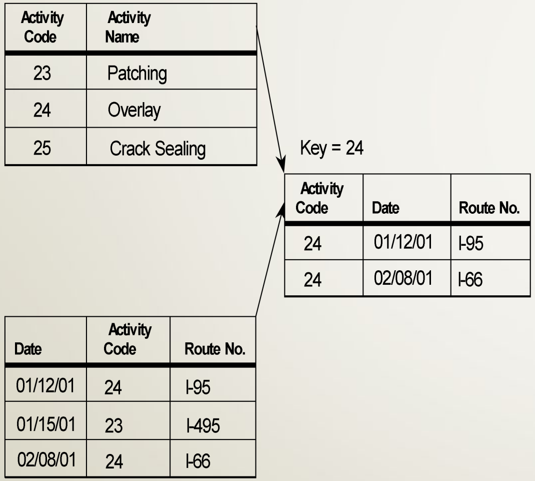
What is a Data Model?

A data model is a notation for **describing data or information**. The description generally consists of three parts:

1. *Structure of the data:*
   - data structures used to implement data in the computer (physical data model)
2. *Operations on the data:*
   - limited set of queries (operations that retrieve information) and modifications (operations that change the database).
3. *Constraints on the data:*
   - ways to describe limitations on what the data can be. These constraints often come from the real-world application requirements

Data Model Examples

Relational Model



Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

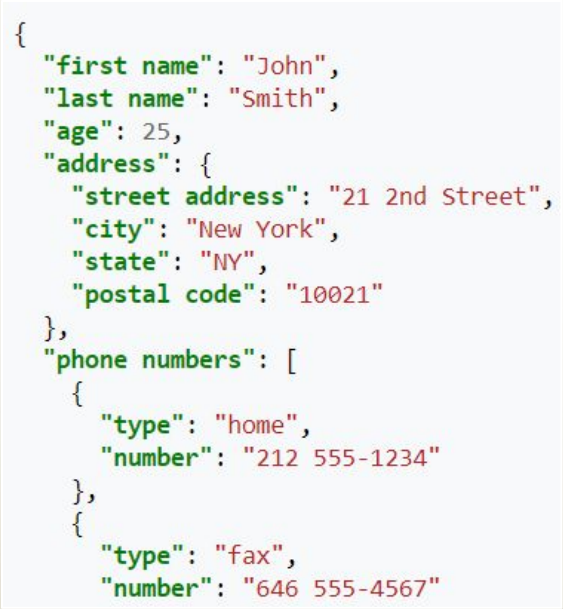
Key = 24

Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495
02/08/01	24	I-66

Src: Wikipedia

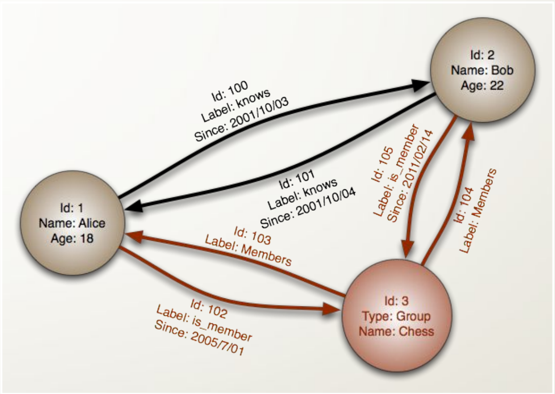
Document (e.g., JSON)



```
{
  "first name": "John",
  "last name": "Smith",
  "age": 25,
  "address": {
    "street address": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postal code": "10021"
  },
  "phone numbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "fax",
      "number": "646 555-4567"
```

Src: Wikipedia

Graph Model



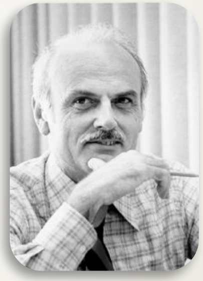
Src: Wikipedia

Outline

✓ Data Models

- Relational Database Model
  - Basic Concepts and Terminology
  - Keys and Foreign Keys
  - Schema Specifications

The Relational Data Model



It all began with a breakthrough paper by E.F. Codd in 1970:
"A relational model of data for large shared data banks".
Communications of the ACM 13 (6): 377

Codd's insights:

- Separate physical implementation from logical.
- Describe the data and operations mathematically.

Database from a user Perspective

- We'll assume that a DB has been already been implemented and loaded with data
- Our roles is to query/modify the data using SQL
- But before that, we need to learn the basics of relational model (from a practical point of view)

Introduction to Relational Databases from a Practical Point of View

Account

Number	Owner	Balance	Type
101	J. Smith	1000.00	checking
102	W. Wei	2000.00	checking
103	J. Smith	5000.00	savings
104	M. Jones	1000.00	checking
105	H. Martin	10,000.00	checking

Imagine that this table (or relation) has been defined to help keep track of bank accounts.

Table Structure

The *name* of the table

Account

The name of the *columns* (*attributes*)

Number	Owner	Balance	Type
101	J. Smith	1000.00	checking
102	W. Wei	2000.00	checking
103	J. Smith	5000.00	savings
104	M. Jones	1000.00	checking
105	H. Martin	10,000.00	checking

Table Schema

The *schema* for the table

Account

Number	Owner	Balance	Type
101	J. Smith	1000.00	checking
102	W. Wei	2000.00	checking
103	J. Smith	5000.00	savings
104	M. Jones	1000.00	checking
105	H. Martin	10,000.00	checking

The schema sets the structure of the table. You can think of the schema as the *definition* of the table. (Note, the schema specifies more information than what is shown.)

Table Rows

Account

Number	Owner	Balance	Type
101	J. Smith	1000.00	checking
102	W. Wei	2000.00	checking
103	J. Smith	5000.00	savings
104	M. Jones	1000.00	checking
105	H. Martin	10,000.00	checking

Each entry in the table is called a ***row*** (***tuple***).

Sometimes an entry in the table is called a record.

Table Instance

An *instance* of the table...

the current contents or data in the table.

Account

Number	Owner	Balance	Type
101	J. Smith	1000.00	checking
102	W. Wei	2000.00	checking
103	J. Smith	5000.00	savings
104	M. Jones	1000.00	checking
105	H. Martin	10,000.00	checking

Another Table Instance

Another *instance* of the table

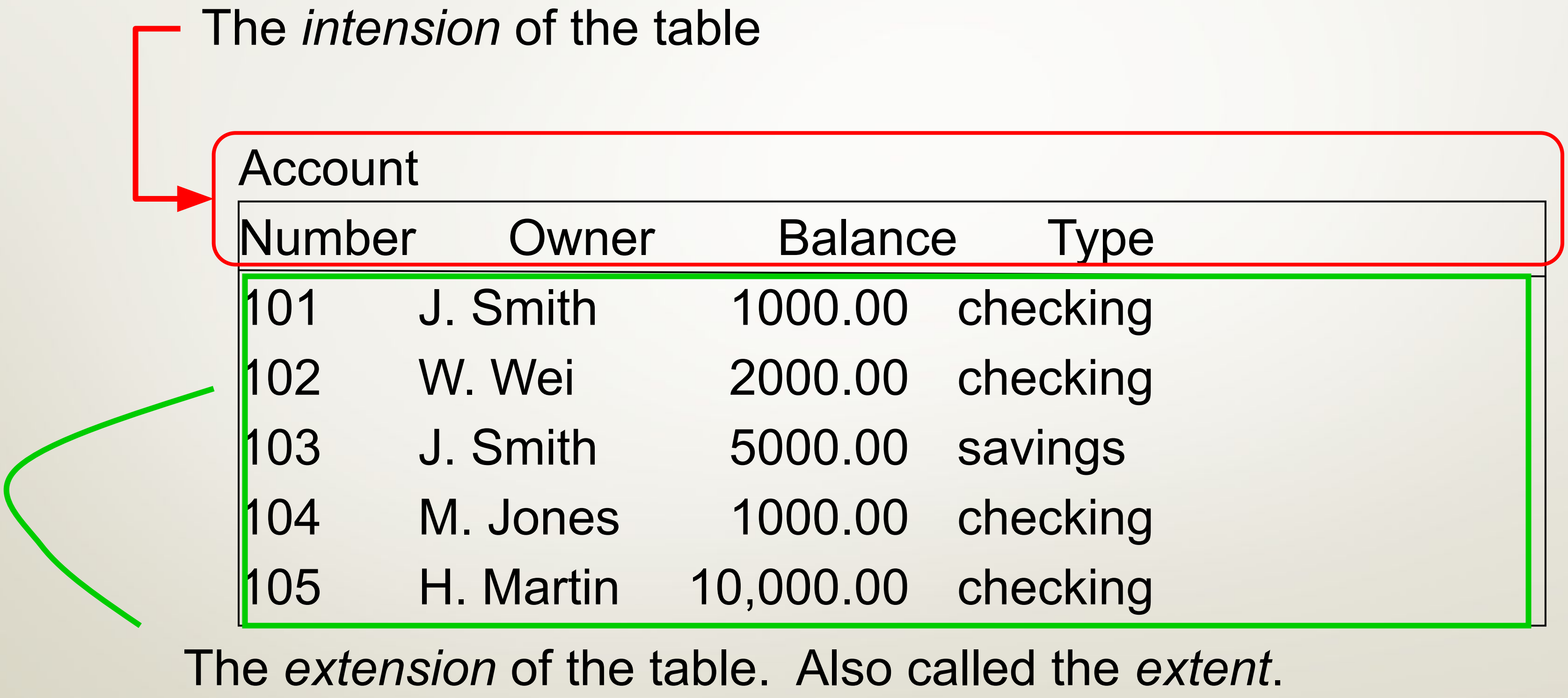
(two rows added, one (103) deleted)

Account

Number	Owner	Balance	Type
101	J. Smith	1,000.00	checking
102	W. Wei	2,000.00	checking
104	M. Jones	1,000.00	checking
105	H. Martin	10,000.00	checking
107	W. Yu	7,500.00	savings
109	R. Jones	432.55	checking

new

Intension vs. Extension



The *intension* of the table

Account

Number	Owner	Balance	Type
101	J. Smith	1000.00	checking
102	W. Wei	2000.00	checking
103	J. Smith	5000.00	savings
104	M. Jones	1000.00	checking
105	H. Martin	10,000.00	checking

The *extension* of the table. Also called the *extent*.

“Size” of a Table

Degree or arity of a table is the number of columns

Degree of this relation (or table) is 4 because there are 4 attributes

Account

Number	Owner	Balance	Type
101	J. Smith	1000.00	checking
102	W. Wei	2000.00	checking
103	J. Smith	5000.00	savings
104	M. Jones	1000.00	checking
105	H. Martin	10,000.00	checking

Cardinality of this instance is 5 (because there are 5 rows)

Cardinality of a table = the number of rows in the current instance

Outline

- ✓ Data Models
- Relational Database Model
  - ✓ Basic Concepts and Terminology
  - Keys and Foreign Keys
  - Schema Specifications

Database (One or More Tables)

Account	Number	Owner	Balance	Type
	101	J. Smith	1000.00	checking
	102	W. Wei	2000.00	checking
	103	J. Smith	5000.00	savings
	104	M. Jones	1000.00	checking
	105	H. Martin	10,000.00	checking

Deposit	AcctNo	Transaction-id	Date	Amount
	102	1	10/22/00	500.00
	102	2	10/29/00	200.00
	104	3	10/29/00	1000.00
	105	4	11/02/00	10,000.00

Check	AcctNo	Check-number	Date	Amount
	101	924	10/23/00	125.00
	101	925	10/24/00	23.98

Table Keys

Account	Number	Owner	Balance	Type
	101	J. Smith	1000.00	checking
	102	W. Wei	2000.00	checking
	103	J. Smith	5000.00	savings
	104	M. Jones	1000.00	checking
	105	H. Martin	10,000.00	checking

Deposit	AcctNo	Transaction-id	Date	Amount
	102	1	10/22/00	500.00
	102	2	10/29/00	200.00
	104	3	10/29/00	1000.00
	105	4	11/02/00	10,000.00

Check	AcctNo	Check-number	Date	Amount
	101	924	10/23/00	125.00
	101	925	10/24/00	23.98

Each table has a key…. where the values must be unique.

Table Keys (cont.)

Account	Number	Owner	Balance	Type
	101	J. Smith	1000.00	checking
	102	W. Wei	2000.00	checking
	103	J. Smith	5000.00	savings
	104	M. Jones	1000.00	checking
	105	H. Martin	10,000.00	checking

Deposit	AcctNo	Transaction-id	Date	Amount
	102	1	10/22/00	500.00
	102	2	10/29/00	200.00
	104	3	10/29/00	1000.00
	105	4	11/02/00.	10,000.00

Check	AcctNo	Check-number	Date	Amount
	101	924	10/23/00	125.00
	101	925	10/24/00	23.98

Key may consist of one column or two (or more) columns.

Connections between Tables

Account	Number	Owner	Balance	Type
	101	J. Smith	1000.00	checking
	102	W. Wei	2000.00	checking
	103	J. Smith	5000.00	savings
	104	M. Jones	1000.00	checking
	105	H. Martin	10,000.00	checking

Deposit	AcctNo	Transaction-id	Date	Amount
	102	1	10/22/00	500.00
	102	2	10/29/00	200.00
	104	3	10/29/00	1000.00
	105	4	11/02/00	10,000.00
	106	5	12/05/00	555.00

Is this legal?

If not, how do we prevent it from happening?

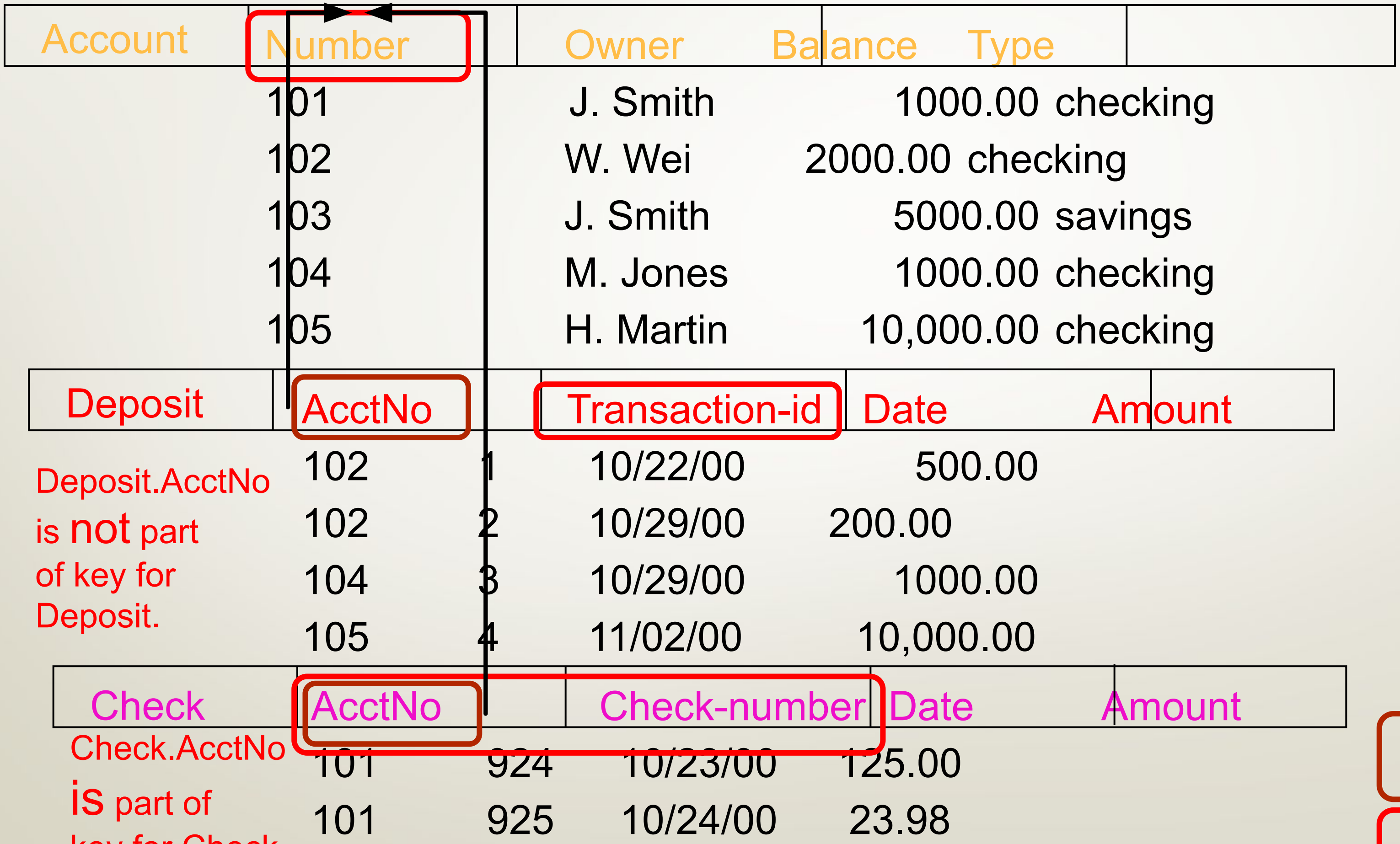
Foreign Key

Account	Number	Owner	Balance	Type
	101	J. Smith	1000.00	checking
	102	W. Wei	2000.00	checking
	103	J. Smith	5000.00	savings
	104	M. Jones	1000.00	checking
	105	H. Martin	10,000.00	checking

Deposit	AcctNo	Transaction-id	Date	Amount
	102	1	10/22/00	500.00
	102	2	10/29/00	200.00
	104	3	10/29/00	1000.00
	105	4	11/02/00	10,000.00
	~~106~~	~~5~~	~~12/05/00~~	~~555.00~~

We say that Deposit.AcctNo is a *foreign key* that *references* Account.Number. If the DBMS enforces this constraint, we have ***referential integrity***.

Foreign keys might or might not be part of the key for the referring table



Account	Number	Owner	Balance	Type
	101	J. Smith	1000.00	checking
	102	W. Wei	2000.00	checking
	103	J. Smith	5000.00	savings
	104	M. Jones	1000.00	checking
	105	H. Martin	10,000.00	checking

Deposit	AcctNo	Transaction-id	Date	Amount
	102	1	10/22/00	500.00
	102	2	10/29/00	200.00
	104	3	10/29/00	1000.00
	105	4	11/02/00	10,000.00

Deposit.AcctNo is **not** part of key for Deposit.

Check	AcctNo	Check-number	Date	Amount
	101	924	10/23/00	125.00
	101	925	10/24/00	23.98

Check.AcctNo **is** part of key for Check.

Foreign Key

Primary Key

Outline

- ✓ Data Models
- Relational Database Model
    - ✓ Basic Concepts and Terminology
    - ✓ Keys and Foreign Keys
    - Schema Specifications

Specification of a Database Schema

- Select the tables, with a name for each table.
- Select columns for each table and give the domain for each column.
- Specify the key(s) for each table.
- Specify all appropriate foreign keys.

There can be more than one key for a table.

Database Domains for Columns

Account	Number	Owner	Balance	Type
	101	J. Smith	1000.00	checking
	102	W. Wei	2000.00	checking
	...			

For every column of every table, the schema specifies allowable values. For example,

Number must be a 3-digit number
Owner must be a 30-character string
Type must be “checking” or “savings”

The set of allowable values for a column is called the domain of the column.

Example Database Schema

(Keys are underlined. Each table has one key.)

Student

sid	name

Takes

sid	exp-grade	cid	sem

Course

cid	subj	sem

Professor

fid	name

Teaches

fid	cid	sem

In relational DBs, we use *relation(attribute:domain)*

STUDENT(sid:int, name:string)

Takes(sid:int, exp-grade:char[2], cid:string, sem:char[3])

COURSE(cid:string, subj:string, sem:char[3])

Teaches(fid:int, cid:string, sem:char[3])

PROFESSOR(fid:int, name:string)

Popularity of the Relational Data Model

- Most popular database systems use the relational model.
  - Oracle
  - MS SQL Server
  - MySQL
  - PostgreSQL
  - IBM DB2
  - SQLLite
  - Microsoft Access
- Check: https://db-engines.com/en/ranking
- Learning about the relational model (and SQL) is a wise investment.

Outline

- ✓ Data Models
- Relational Database Model
  - ✓ Basic Concepts and Terminology
  - ✓ Keys and Foreign Keys
  - ✓ Schema Specifications