Topic 5:

Implementation Data Models

A Wee Bit O’ History

There are four data models of note:

1. Hierarchical (early 1960s, IBM)
2. Network (early 1970s, CODASYL/DBTG)
3. Relational (early 1970s, Codd @ IBM)
4. Object (???)

Hierarchical Model: Background and Ideas

Background:

- John F. Kennedy, May 25, 1961: ‘. . . man on the moon . . . ’
- Rockwell needed to organize parts for the Apollo CM & SM
- IBM created IMS (Information Management System) in 1968
  - ○ original name: ICS/DL/I; thankfully renamed in ’69
  - ○ both used and introduced the Hierarchical Model
  - ○ still sold today!

Hierarchical Model: Terminology

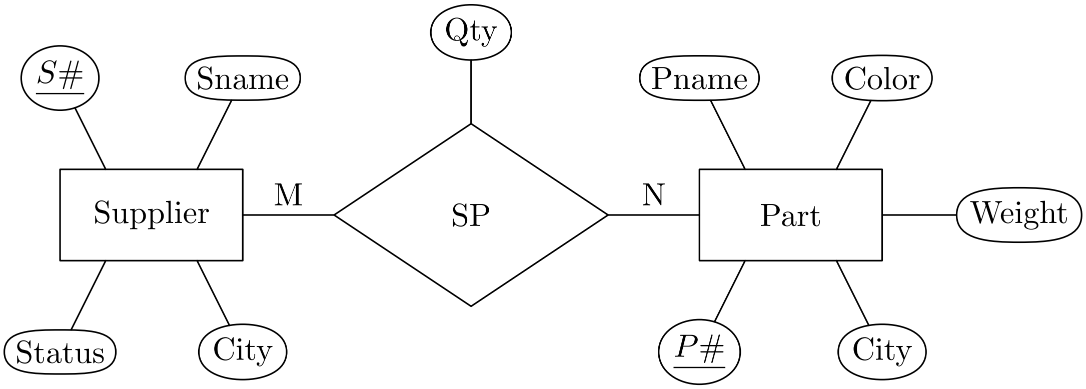
Sample Logical Schema:

Terminology:

E-R Model		Hierarchical Model
Entity Set	≡	
Entity	≡	
Attributes	≡	
Relationships	≡	

Hierarchical Model: Supplier–Part Schema

Consider this subset of Codd's SPJ schema:



Hierarchical Model: M:N Relationships (1 / 3)

Still a Logical Schema, but augmented with fields:

Hierarchical Model: M:N Relationships (2 / 3)

Physical Schema:

Hierarchical Model: M:N Relationships (3 / 3)

The physical schema for Supplier–Part w/ sample data:

Network Model: Background and Ideas

- Created in the early 1970s by CODASYL's

  (Conference/Committee on Data Systems Languages)

  DBTG (Database Task Group)
- Goal: A standard theory of DB systems.

  Origin of the ideas of DML and DDL
- Became an ISO standard in 1987 (ISO 8907:1987)

  (And was withdrawn in 1998!)
- Graph–based instead of tree–based

Network Model: Terminology

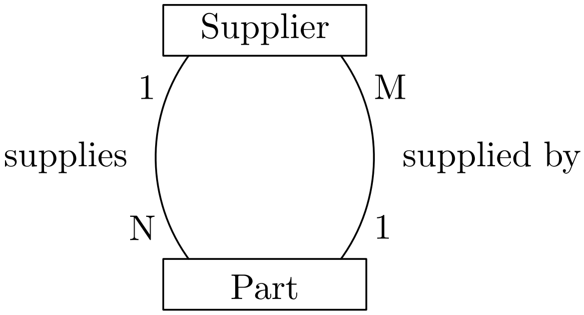
A Sample Logical Schema:

Terminology:

E-R Model		Network Model
Entity Set	≡	
Entity	≡	
Attributes	≡	
Relationships	≡	

Network Model: M:N Relationships (1 / 2)

Logical Schema (of the M:N Supplier - Part relationship):



Network Model: M:N Relationships (2 / 2)

Physical Schema (and *just* a little messy . . .):

Relational Model: Background and Ideas

- Created by Edgar F. Codd. Famous paper:

  “A relational model of data for large shared data banks,” 1970.

- Theoretical foundation: Set Theory
- Uses foreign keys instead of pointers
- No distinction between logical and physical schemas

Relational Model: DMLs

Codd proposed two types of DMLs:

Relational Model: Terminology (1 / 2)

Sample Supplier - Part Schema:

S

S#	Sname	Status	City
S1	Acme	10	Omro
S2	Fubar	10	Fisk
S3	Snafu	20	Ring

P

P#	Pname	Color	Weight	City
P1	Nut	Pink	0.2	Anton
P2	Bolt	Blue	0.9	Borea

Terminology:

E-R Model		Relational Model
Entity Set	≡	
Entity	≡	
Attributes	≡	
Relationships	≡	

Relational Model: Terminology (2 / 2)

Sample Department – Employee Schema:

DEPARTMENT

DeptNum	DeptName	ManagerID	ManagerStartDate
1	Shipping	364	2001-04-01
2	Payroll	NULL	NULL
3	Billing	298	2000-11-17

EMPLOYEE

Surname	GivenName	EmpNum	DeptID	Salary
Spade	Sam	786	1	48000
Trune	Joe	410	2	49500
Smith	Megan	364	1	75000
Maher	Mary	298	3	72000

Relational Model: Misc. Notes

- Order of tuples in a relation is logically irrelevant (Why?)
- Fields are single-valued by default (vs. set-valued)
- Relationships are supported by foreign keys

Relational Model: 1:N Relationships

We've already seen how to do this! (Just two slides ago!)

DEPARTMENT

DeptNum	DeptName	ManagerID	ManagerStartDate
1	Shipping	364	2001-04-01
2	Payroll	NULL	NULL
3	Billing	298	2000-11-17

EMPLOYEE

Surname	GivenName	EmpNum	DeptID	Salary
Spade	Sam	786	1	48000
Trune	Joe	410	2	49500
Smith	Megan	364	1	75000
Maher	Mary	298	3	72000

Relational Model: 1:1 Relationships

- Just a restriction of 1:N relationships:
  - We still store a FK in the ‘many’ relation
  - Must constrain the field’s values to be unique; two options:

Relational Model: M:N Relationships

S

S#	Sname	Status	City
S1	Acme	10	Omro
S2	Fubar	10	Fisk
S3	Snafu	20	Ring

P

P#	Pname	Color	Weight	City
P1	Nut	Pink	0.2	Anton
P2	Bolt	Blue	0.9	Borea

SP

S#	P#	Qty
S1	P1	50
S1	P2	150
S2	P2	25
S3	P1	300

Object Model: Ideas

- OO programming languages have existed since Simula in 1967
- We'd like to be able to store objects in a DBMS
  - ○ provides object persistence
  - ○ can do it by mapping object instance fields to relational tuples, but that's clunky
- The Object Data Management Group (ODMG) defined an object-based DBMS standard
  - ○ finished ODMG 3.0 in 2001 (and then disbanded!)

Object Model: Object DBMS Types

Two major varieties:

1. Object Oriented DBMS (OODBMS)
   - Marriage of an OOPL and a DBMS
2. Object Relational DBMS (ORDBMS)
   - A relational DBMS with added objects