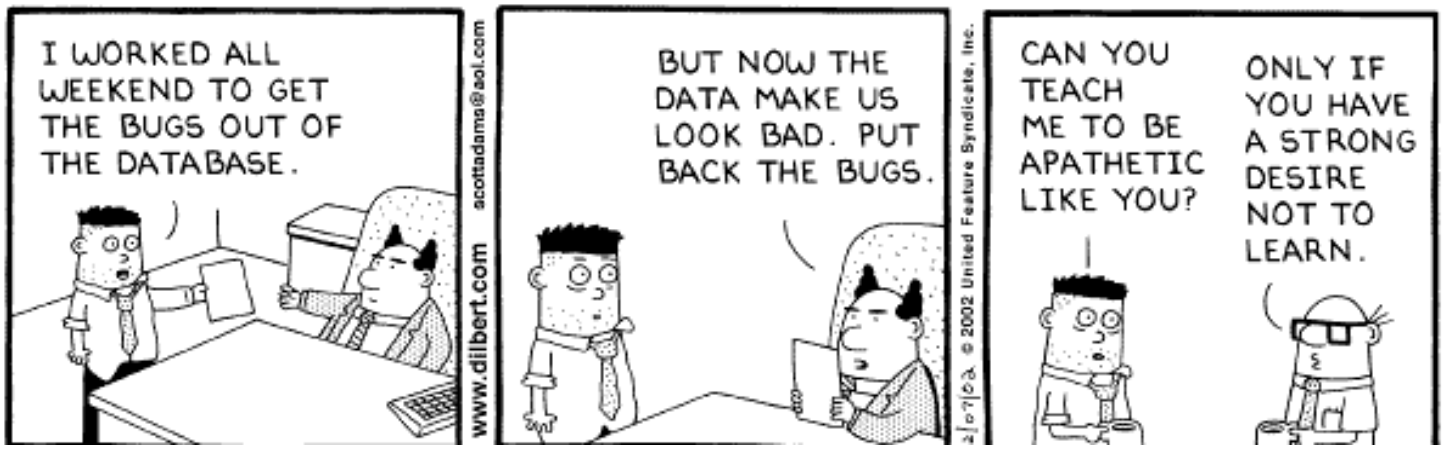
Fresno County office assistant Pat Acevedo reshelves a record amid vast, overflowing files under the old filing system.

Transferring paper files to electronic storage has freed a warehouse, which saves the county money. Switching over to electronic files has improved morale and reduced injuries.

The Fresno County Clerk's Office celebrated a year's effort of reorganizing and transferring files into electronic form.

With the remaining files filed correctly at the county's storage site on Ventura Avenue, the goal is to eliminate the warehouse.

(Fresno Bee, December 16, 2004)

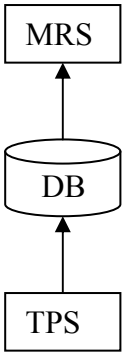

I WORKED ALL WEEKEND TO GET THE BUGS OUT OF THE DATABASE.
www.dilbert.com
scottadams@aol.com
BUT NOW THE DATA MAKE US LOOK BAD. PUT BACK THE BUGS.
2/07/02 © 2002 United Feature Syndicate, Inc.
CAN YOU TEACH ME TO BE APATHETIC LIKE YOU?
ONLY IF YOU HAVE A STRONG DESIRE NOT TO LEARN.


I THINK WE SHOULD BUILD AN SQL DATABASE.
UH-OH
S. Adams
E-mail: SCOTTADAMS@AOL.COM
DOES HE UNDERSTAND WHAT HE SAID OR IS IT SOMETHING HE SAW IN A TRADE MAGAZINE AD?
WHAT COLOR DO YOU WANT THAT DATABASE?
I THINK MAUVE HAS THE MOST RAM.
11/17
© 1995 United Feature Syndicate, Inc.(NYC)

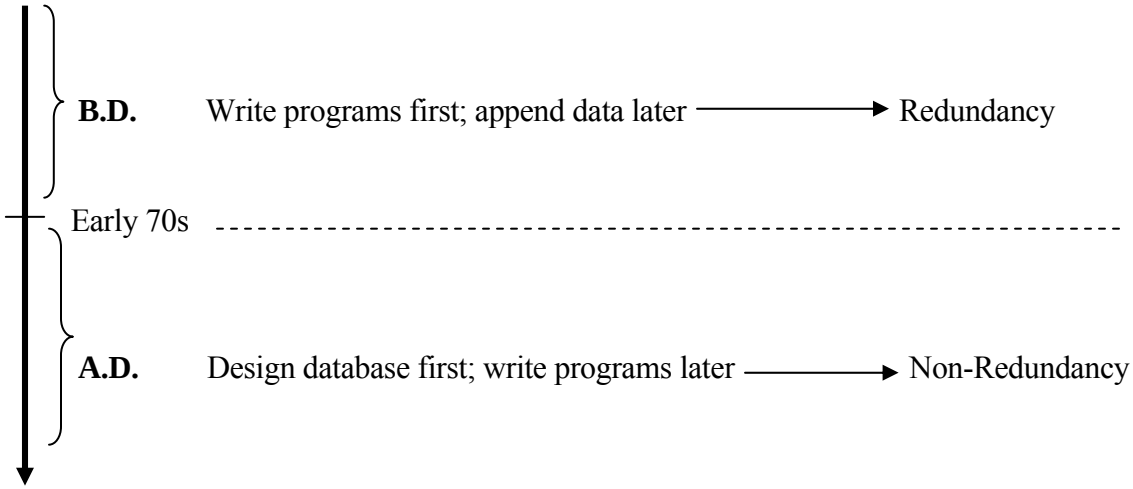
The Database Approach: Making a Case For It

The Buffer Principle:
The reason for storing a resource is that the time of acquiring it is different from the time of consuming it.

Database = Collective, corporate memory; the buffer between data production (TPS) and consumption (MRS).



The database timeline:



The Data-Redundancy Principle:
With humans, redundancy is often desirable in that it reinforces behavior. With computers, redundancy is always undesirable.

Redundant Data:

- Waste storage space
- Generate contradictory information
- Make it difficult to revise data formats
- Make it difficult to produce ad hoc reports

A **FLAT FILE** is a file containing redundant, repetitive data (mixing apples and oranges).

The intuitive justification of flat files from the user viewpoint: The illusion of one-to-one.

What the users think is happening:

- Because data enter the system through one screen, they get stored in one file.
- Because certain information appears on one report, it came from data stored in one file.

What is really happening:

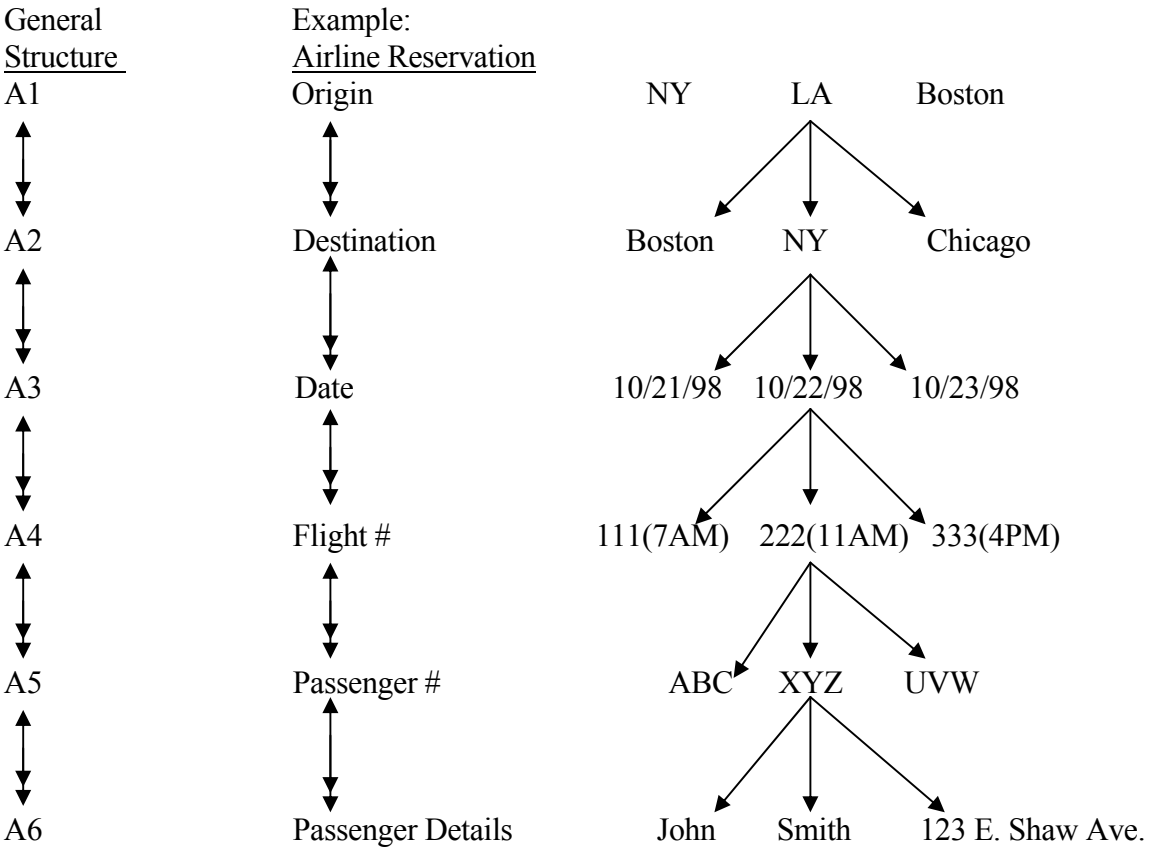
- Data entering the system through one screen may update multiple (related) files.
- Information appearing on one report may be based on data pulled from multiple (related) files.

Advertising agency (customers = corp accounts)
A/R : delinquency ➔ who are our top 5 most delinquent accounts? (name, amount, lateness)
Mktg: customer ➔ who are our top 5 oldest accounts? (name, address, age of relation)
HR: employess ➔ who are our top 5 performing salespeople (sales/name/phone)

Now Combined:
Is there a relationship between delinquency, tenure as customer and salesrep performance?
who are?
For our top 10 most delinquent accounts list:
name, amount, lateness, address, age of relation, performance name/phone of salesperson

Selecting the Appropriate Data Model

The Hierarchical Model



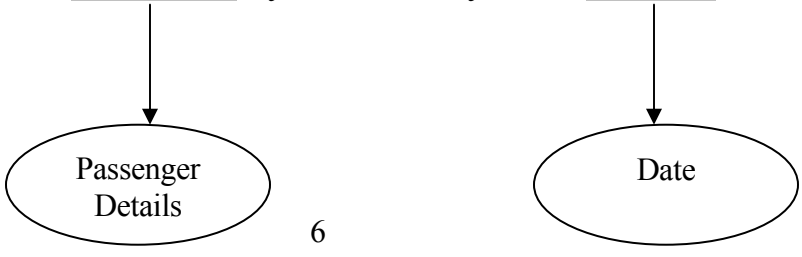
Possible Number of Branches:
10 origins * 10 destinations * 365 days/year * 30 years * 2 flights * 100 passengers * 10 data/passenger ➔ 2,190,000,000

Transaction Processing:

Issuing a ticket involves traversing a single branch.

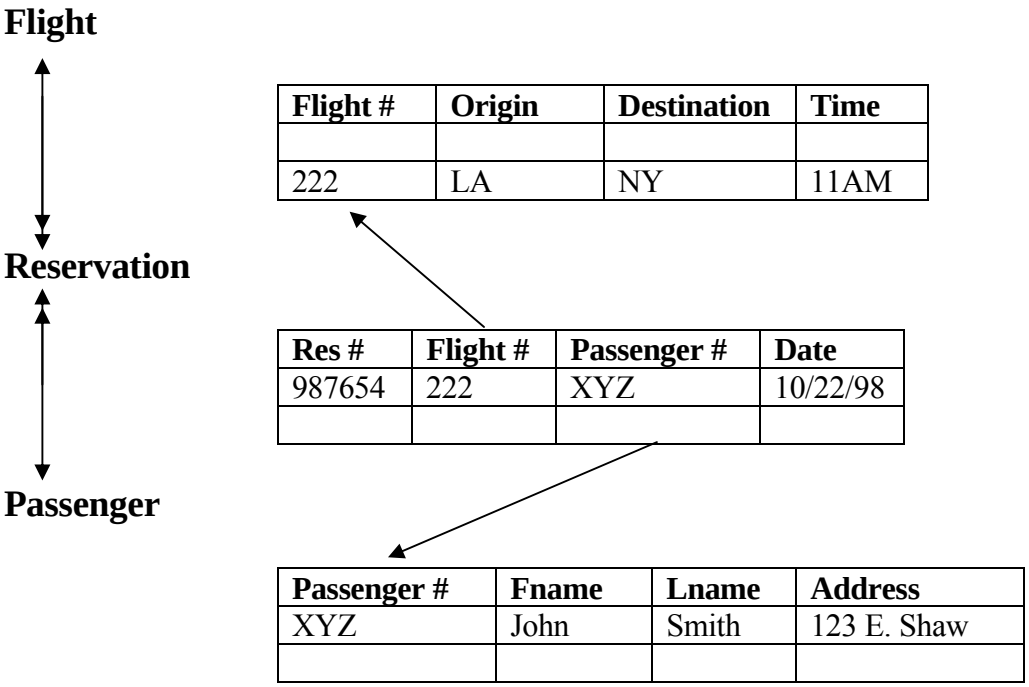
Query Processing:

Answering a question involves traversing all branches until the right one is found.
Example: What cities did John Smith fly from, and fly to, on 10/22/98?



The Relational Model

Information about each entity or event is stored in a table, and the tables are linked through common fields.



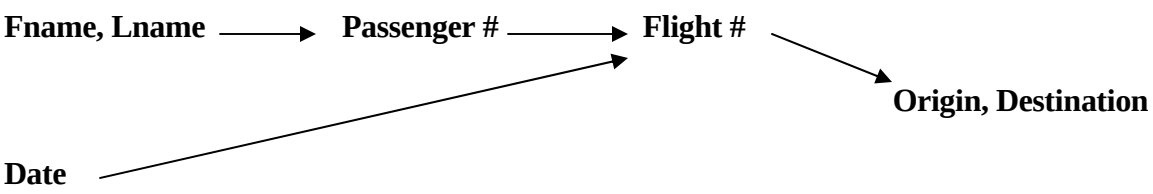
Transaction Processing:

Issuing a ticket involves reading one table while updating two others.

Query Processing:

Answering a question involves navigating the relevant tables via linking fields to find the right record.

Example: What cities did John Smith fly from, and fly to, on 10/22/98?



Comparison of the Two Models

	Transaction Processing (Issuing thousands of tickets per day)	**Query Processing** (What cities did John Smith fly from/to on 10/22/98?)
Hierarchical	*Strong*	*Weak*
Relational	*Weak*	*Strong*

Relational Database Design

Objective: the systematic removal of all data redundancy

- *Repetitive processes should be automated.*
- *Repetitive data should be eliminated.*

Methodology: Breaking up a flat file into smaller, non-redundant data representing the fundamental business entities/events being tracked (separating apples from oranges)

A public library records data about the details of each transaction in one large file, shown below. The file is populated with sample data. No separate record is kept of patrons, books, and book borrowing details other than in this file.

Patron Name	Patron Address	Book ID	Book Title	Book Author	Borrow Date	Due Date	Return Date
J. Smith	12 Elk	AAA	Peace	A. Bart	03/04/94	03/18/94	03/15/94
M. Jones	25 Sun	BBB	War	M. Hine	03/04/94	03/18/94	03/19/94
G. Hart	73 Sera	CCC	System	N. Vang	03/05/94	03/19/94	03/23/94
V. Hicks	22 Mann	AAA	Peace	A. Bart	03/19/94	04/02/94	04/02/94
E. Rice	69 Witt	DDD	Spring	F. Lyon	03/06/94	03/20/94	03/08/94
M. Jones	25 Sun	CCC	System	N. Vang	03/26/94	04/09/94	04/08/94

Explain what problems arise in performing the following tasks.

1. Adding the details of a new book to the file.
   BOOK ID: EEE BOOK TITLE: Fate BOOK AUTHOR: K. Chen

2. Changing the address of a patron.
   The address of M. Jones, from "25 Sun" to "26 Moon."

3. Deleting a patron record.
   E. Rice discontinued membership. His record needs to be deleted.

4. Querying a record.
   A prospective patron calls to find out if the library carries the book entitled Spring.

Lessons to Be Learned

Databases designed "intuitively" by amateurs tend to incorporate redundancy.

Redundancy is stating a fact more than once.

A fact is not a cell/value in the table, such as

- M. Jones
- 25 Sun
- *War*

A fact is the linking of two cells/values, such as

- M. Jones lives at 25 Sun
- M. Jones has borrowed the book *War*
- The book *War* has been borrowed by a resident of 25 Sun

Each of the above facts has been stated more than once.

There are generally two (A/B), and specifically four (1/2/3/4), things a database can do.

A. Update (write) ➔ TPS
   1. *Add* a new record
   2. *Change* an existing record
   3. *Delete* a record

B. Access (read) ➔ MRS
   4. *Retrieve* a record

There are four database anomalies

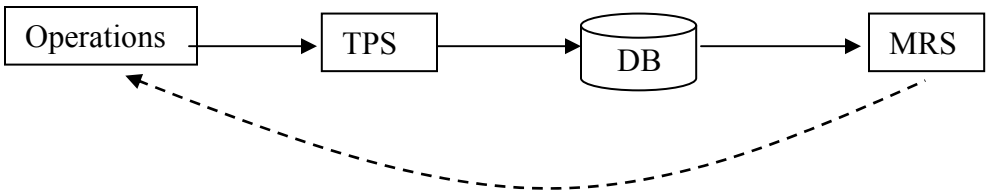
1. *Addition* Anomaly
2. *Change* Anomaly
3. *Deletion* Anomaly
4. *Retrieval* Anomaly

Data redundancy is the root of all database anomalies.

The goal of database design: The elimination of all redundancy.

Database structure must reflect business practices.

Database Management Systems



Database Operations
1. Update (write) ➔ TPS
   - Add record
   - Change record
   - Delete record
2. Access (read) ➔ MRS

The Data Hierarchy

BIT : the amount of information conveyed when, between two equally likely outcomes, one is specified. The two outcomes are often represented by "0" and "1."

BYTE : a set of related bits, defining a character. In the ASCII, 1001010 = "J"

FIELD : a set of related bytes, defining an attribute of something. = "table cell"
JOHN = the first name of a student.

RECORD : a set of related fields, defining all the relevant attributes of something. = "table row"
{John, Smith, 4548 First Street, Fresno, CA, 93710, Junior, English} = the record of a student

FILE : a set of related records, comprising all the relevant information about a type of thing. The records of all students at a university = STUDENT file. = "table"

DATABASE : a set of related files, comprising all the relevant information about all the relevant entities or events.
University database = {STUDENT, INSTRUCTOR, COURSE, REGISTRATION, GRADE, …}

Database Design Methodology: Eliminate Redundancy!

1. Identify and name all the business "things" that need to be tracked (= Entities). Each entity becomes the name of a table (= Master File)
   CAR, CUSTOMER, EMPLOYEE

   Examples are all from the car rental assignment

2. For each entity, identify all the relevant attributes that need to be tracked. Each attribute becomes a column heading in the table.
   CAR = make, model, color, size, mileage, fuel
   CUSTOMER = fname, lname, street, city, state, zip, phone

3. For each entity, create an attribute that uniquely identifies every instance of that entity (= Primary Key, or simply Key). The key becomes the first column heading in the table.
   CAR: car-id
   CUSTOMER: customer-id

4. Establish the relationships among the various entities (= cardinality), and draw them in a diagram (= Entity-Relationship Diagram)

Alternate Symbols
One: 1
Many: M, ∞

√ - **ONE-TO-ONE**: A ⟷ B
For every A there is only one B, and for every B there is only one A.
Rental ⟷ Return

√ - **ONE-TO-MANY** A ⟷ B
For every A there is only one B, but for every B there are multiple As.
Rental ⟷ Car

X - **MANY-TO-MANY**: A ⟷ B
For every A there are multiple Bs, and for every B there are multiple As.
Customer ⟷ Car Not acceptable! Why?!
http://zimmer.csufresno.edu/~sasanr/Teaching-Material/MIS/DBMS/Where-Does-Rental-Data-Go.pdf

5. For each many-to-many relationship between two entities, decompose it into two simpler one-to-many relationships by inserting a table in the middle, representing a transaction (event) linking the two entities. This table (= Transaction File) captures data about the business event that links a single instance of one entity to a single instance of the other entity.

Customer ⟷ Car is decomposed into

Customer ⟷ **Rental** ⟷ Car

[*Database structure reflects business practices*]

6. For each new event thus discovered, identify all the relevant attributes that need to be tracked. Each attribute becomes a column heading in the table.
RENTAL = rental-date, rental-time, expected-return-date, location

7. For each new event thus discovered, create an attribute that uniquely identifies every instance of that event (= Primary Key, or Key). The key becomes the first column heading in the table.
RENTAL: rental-id

8. Assure that the primary keys of all the files linked together are included in the transaction file linking them (= Foreign Key)
RENTAL = customer-id, car-id
[*The primary key of one table becomes a foreign key in another table*]

foreign keys

primary key

RENTAL = rental-id, rental-date, rental-time, expected-return-date, location, customer-id, car-id

A Sample Database

Table (AKA "file")

Foreign Key

Primary Key

Entity

Field

Attribute (AKA "field")

Character (AKA "byte") eg: 11010010

BIT (BInary digiT)

Record

CUSTOMER

Customer Number	Customer Name	Customer Address	Balance	Credit Limit	Salesrep Number
C1	Peter Grant	111 First Street	100	500	S1
C2	George Mason	222 Second Street	230	700	S2
C3	Arlene Smith	333 Third Street	560	800	S2
C4	Tim Wendon	444 Fourth Street	900	700	S3
C5	Susan Yates		200	500	S1
C6	Peter Grant	666 Sixth Street	810	800	S3
C7	Joanna Leet	777 Seventh Street	500	600	S1

SALES REP

Salesrep Number	Salesrep Name	Salesrep Address	YTD Commission
S1	Benny	234 James Street	34,000
S2	Jane	40 Lindt Ave.	78,000
S3	Eric	569 Tenth St.	100,000

PART

Part Number	Description	Quantity on Hand	Unit Price
P1	Trophy	45	12.95
P2	Dar t	100	2.30
P3	Bals	230	1.90
P4	Brace	12	6.95
P5	Wheel	85	7.80
P6	Timer	88	29.50
P7	Chair	3	5.60
P8	Drum	123	67.11

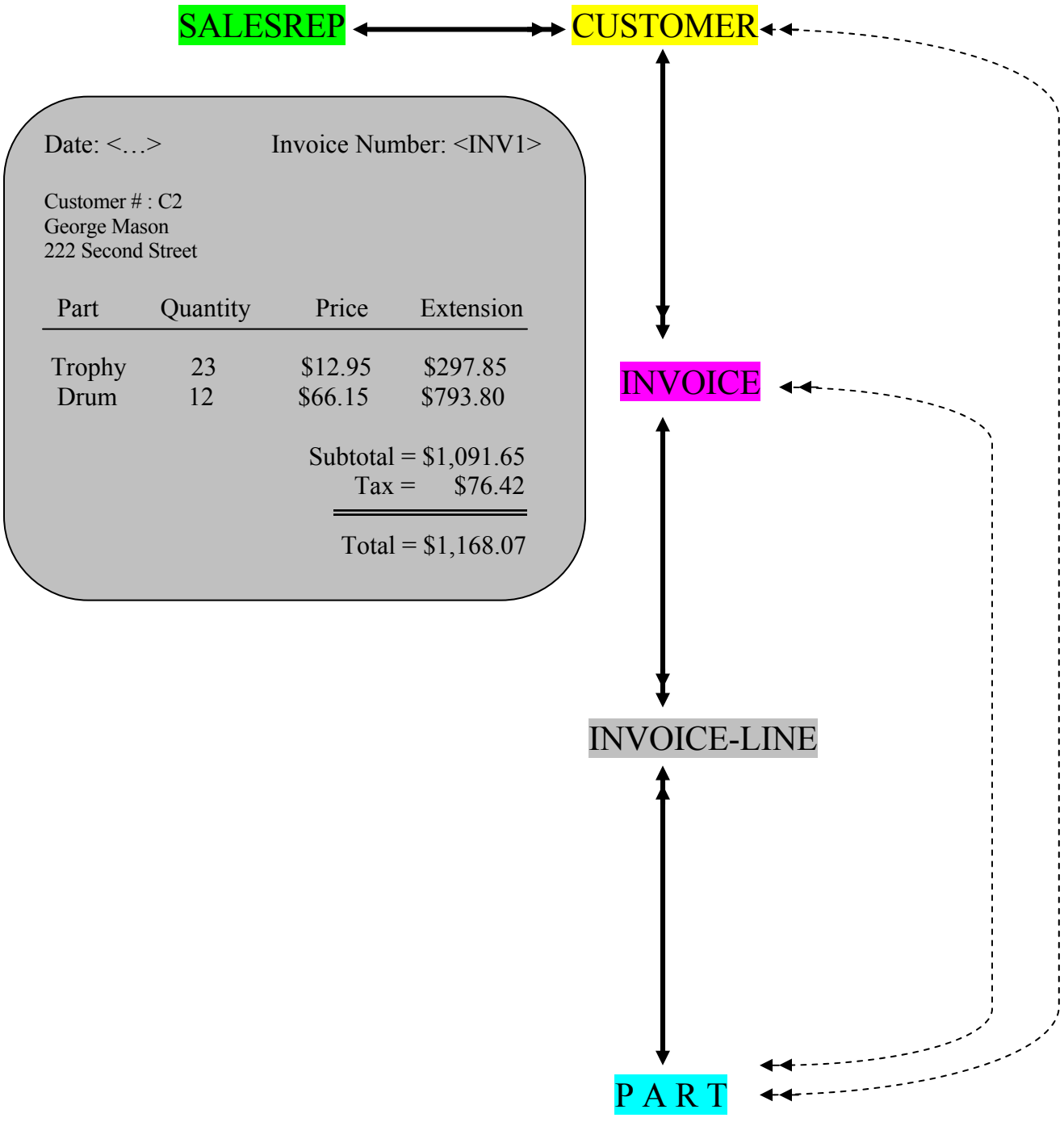
INVOICE

Invoice Number	Date	Customer Number
INV1	06/15/94	C2
INV2	06/22/94	C1
INV3	06/23/94	C5

INVOICE-LINE

Invoice Number	Part Number	Quantity	Price
INV1	P1	23	12.95
INV1	P8	12	66.15
INV2	P2	5	2.25
INV2	P5	6	7.80
INV3	P3	29	1.50

Entity-Relationship Diagram



Many-to-many relationships (◂◂- - - - - -▸▸) are impermissible!

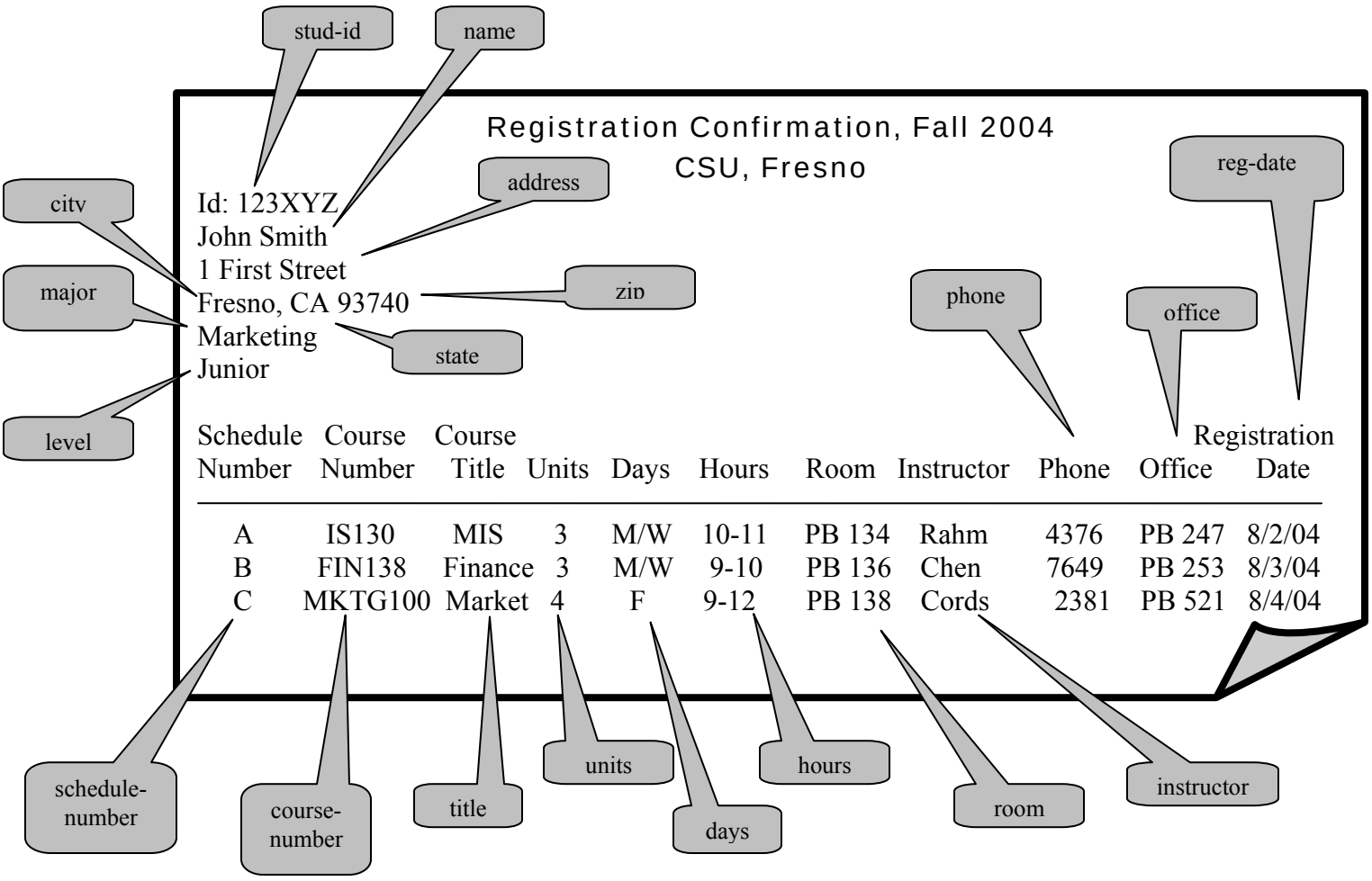
- What is the YTD commission of the salesperson for the customer on invoice INV1?
- How much have all of Eric's customers bought over the past two weeks?

DB Design Exercise #1

Registration Confirmation, Fall 2004
CSU, Fresno

Id: 123XYZ
John Smith
1 First Street
Fresno, CA 93740
Marketing
Junior

Schedule Number	Course Number	Course Title	Units	Days	Hours	Room	Instructor	Phone	Office	Registration Date
A	IS130	MIS	3	M/W	10-11	PB 134	Rahm	4376	PB 247	8/2/04
B	FIN138	Finance	3	M/W	9-10	PB 136	Chen	7649	PB 253	8/3/04
C	MKTG100	Market	4	F	9-12	PB 138	Cords	2381	PB 521	8/4/04


stud-id
name
Registration Confirmation, Fall 2004
CSU, Fresno
reg-date
address
city
Id: 123XYZ
John Smith
1 First Street
major
zip
Fresno, CA 93740
phone
office
Marketing
state
Junior
level
Schedule Number
Course Number
Course Title
Units
Days
Hours
Room
Instructor
Phone
Office
Registration Date
A IS130 MIS 3 M/W 10-11 PB 134 Rahm 4376 PB 247 8/2/04
B FIN138 Finance 3 M/W 9-10 PB 136 Chen 7649 PB 253 8/3/04
C MKTG100 Market 4 F 9-12 PB 138 Cords 2381 PB 521 8/4/04
schedule-number
course-number
title
units
days
hours
room
instructor

DB Design Exercise #2

You run the Puppies For Yuppies business. Each puppy has a number, a name, and is kept in a kennel with its own number, name, and location. A kennel houses multiple puppies. Every puppy knows multiple tricks, and knows each trick at a certain skill level. For example, Fifi can roll over at level 6, but can fetch the paper at level 2. Each trick has an ID and a name of its own. The following fields are therefore relevant:

PNO (puppy number)
PNAME (puppy name)
KNO (kennel number)
KNAME (kennel name)
KLOC (kennel location)
TID (trick ID)
TNAME (trick name)
LEVEL (skill level)

Design the database.

When you are done, check (for yourself) to see if your design allows you to perform the following queries.

- At what level does Shaggy fetch the paper?
- Can Fifi roll over?
- What are the names of the tricks Shaggy can do?
- How many puppies are in kennel 3?
- How many puppies in the Daisy Hill kennel can perform a trick (any trick) at level 10?
- What is the highest level trick Bugger can perform?

DB Design Exercise #3

You are in the business of making and repairing vending machines. You have many customers, each an organization. Each customer has a number of vending machines at their site. A repair call from a customer initiates a repair order which is then assigned to a repair person to handle. Only one repair person handles a repair. Each repair uses certain parts, and possibly multiple units of each part. So, for instance, a certain repair may use 6 units of part AAA and 4 units of part BBB.

Management would like to be able to enter some identifying information about a machine and have the system bring up:

- the name and address of the customer where the machine is located
- the year the machine was manufactured, and its mean-time-between-repairs to date (i.e., the average time elapsed between two consecutive repairs)
- for each of the last three repairs of that machine:
    - its duration (how long the repair took)
    - the name of the repair person, and the number of years they have been working for your firm
    - the part numbers, part descriptions, and number of units of each part used in that repair

Design the database.

DB Design Exercise #4

A physician has to inoculate his patients based on a certain schedule. There are different shots to be given to patients. Each shot is to be given to all patients, but at different times, depending on the shot's frequency. On first signing up with this physician, a patient receives all the shots at that time. Each future shot is then given based on the shot's frequency. So, for instance, the new patient will receive another shot (the "twice-a-year" shot) six months after the initial sign-up, and another shot (the "annual" shot) a year after the sign-up, and so on and so forth. No two shots have the same name. Two patients may have the same name. Historic data (of every shot given to every patient in the past) are to be kept for government auditing purposes.

If something serious goes wrong with a shot, the physician would like to know which staff member gave that shot to that patient. The physician would like to be able to identify the patient and the shot for the computer, and have the computer show the name, hiring date, and address of the staff member who last administered that shot to that patient, in addition to the name and telephone number of that patient.

Design the database.