What Is a DBMS?

- Database: A collection of information. Eg ?
- Examples: Library, University
- Database Management System (DBMS) : software package designed to store and manage databases.
  - Eg: Oracle, SQL server, MySQL, Access
- Files vs DBMS : Why bother with databases ?
  - Why not just just store all the data in a big file and write C or Java programs to manipulate the data.

Why Use a DBMS?

- Naïve users sheltered from messy details
- Data integrity:
  - Eg: if Bob works in Marketing, make sure there is a dept called Marketing.
- Reduced application development time: Avoid writing special programs from scratch each time to access data.
- Standard Application Interface: increased reliability

Why Use a DBMS?

- Data independence: easier to make changes
  - If how data is stored changes
- Security: easier to control how data is shared
- Concurrent access: allow multiple users to access simultaneously
  - But in a controlled way !

Different people involved

- DBMS implementers: who build the DBMS like Oracle, MS SQL server
- End users: forms, reports, SQL queries
- DB application programmers: write programs to make life easier for end users.
  - Eg: person who creates forms for library.
  - Must know how databases work
- DB administrator (DBA):
  - Handles security and authorization
  - Crash recovery
  - Database tuning as needs evolve

Overview of course: Relational Model: Student Database, Fig 1.2

STUDENT

Name	StudentNumber	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Overview of course:

- Data Models:
  - High level : E.R. model
  - Intermediate level : relational model
    - Student database
  - Low level: physical database: 622B/524
- Relational databases:
  - Integrity constraints
  - Good design : normalization
- Query languages: Relational algebra, SQL
- Views, Assertions, Triggers

Relational Data Model

- Relation: 2-dimensional table
  - All info stored in tables
- Eg: student, course Elmasri Fig 1.2
  - Rows (or tuples): student : 2 rows
    - Records: a row may correspond with a record in a file
      - Commonly used if we are talking about the physical storage of databases
  - Columns (or attributes): student : 4 columns

Relational Data Model

- Relational model proposed by E. F. Codd 1970
- Dominant model in commercial DBMS products.
  - Eg: Oracle, SQL server, MySQL, Access.
- Compared to previous models (network, hierarchical etc):
  - Easier to understand info in tables
  - Casual user can write simple SQL queries
  - Complex queries much easier to understand compared to previous models.

Basic Terminology

- Relational Schema (or head): set of all the column names i.e. what info is being stored. For student table:
  - Name, StudentNumber, Class, Major
- Relational Database Schema : collection of all the relational schemas
- Relational Instance (or current relational state): what data is currently in the table.
  - In the Eg in Elmasri Fig 1.2 student database, there are two rows (Smith …, Brown …) in the relational instance.
- Schema vs instance:
  - schema changes infrequently
  - instance changes often as data changes, rows added deleted.

Relational Rules

- Attempt to standardize across different products, so can discuss design issues in general
- Have to be true for relation at all times, not just happen to be true for the current relational instance
- Some rules violated in some products. Why ? For efficiency and for ease of use, a “clean mathematical design” may be sacrificed.
- Some rules violated in SQL itself.

First Normal Form rule

- Every value atomic: No multi-values, composite. Followed by "all" DBMS.
  - Not followed in OODBMS, ORDBMS
- No multi-valued field. Eg: if we are looking at locations of projects, can't store Los Angeles, NY in single attribute

project	location
Finance	L.A., N.Y

project	location
Finance	L.A.
Finance	N.Y

- How to fix ?
- Split into different rows

First Normal Form rule

- No composite values. Eg: if we are looking to store names, can't have sub-fields of name as fname, lname

name	
fname	lname
mike	smith

- How to solve? Split into different columns

fname	lname
mike	smith

Unique Row Rule

- No two rows identical: they have to differ in at least one value (one column).

project	location
Finance	L.A.
Finance	N.Y

OK

project	location
Finance	L.A.
Finance	L.A.

NOT OK

- SQL allows duplicate rows: default. Why ?
- Lot of work may be needed to ensure no duplicates: Eg. When combining two tables.
- Duplicates may be useful. Eg. When counting how many employees make > 50k. Enumerate salaries and count

Rows not ordered

project	location
Finance	L.A.
Sales	N.Y

same as

project	location
Sales	N.Y
Finance	L.A.

- To access rows: do by content (where is Sales located) rather than row # (what is the location in the 2nd row). We will assume this is true.
- Commercial DBMS break this rule. Why ?
- Rows ordered: logically no, physically yes
- For performance, because of physical locality, order can matter.

Columns not ordered

project	location
Finance	L.A.
Sales	N.Y

same as

location	project
L.A.	Finance
N.Y	Sales

- Property of relations, but rule violated in SQL:
  Eg: when inserting a new row into the table above left, can just insert Marketing, Chicago.
  - Don't have to specify project = Marketing, location = Chicago. How ?
- Since Marketing written before Chicago, DBMS will assume Marketing goes into 1st column, Chicago goes into 2nd column.

Domain

- The values an attribute can take: Eg: string, integer, real
- Domain of each attribute is specified when creating a table in SQL
- Enforced by the DBMS when making changes or adding a new row.
- Eg: age: integer
  - If try to set age to 7.3, DBMS will not allow

Null value

- An attribute can take null value. How to interpret ?
- Doesn't exist: Eg ?
- Supervisor attribute for employees, but some employees may not have a supervisor.
- Don't know the value: Eg ?
- Birth date
- Don't know if it exists or unknown value: Eg ?
- Work phone for customers in online store database: customer may not have, or may choose not to give

Keys

- Super key of a relation: one or more attributes that uniquely identifies which rows we are talking about.
- Eg: for COURSE table which are super keys?
- {C#}, {CName}, {C#, CName}, {C#,Dept}…
- Has to be time invariant: always true, not just happens to be true for current instance
- Eg: in STUDENT, Name is unique, but may not always be true

Keys

- Have to look at semantics: to decide if superkey, is not enough to look at instance
- Key: a super key which is minimal i.e. if we remove any attribute from a key, will no longer uniquely identify a row.
- Eg: {C#, CName} for COURSE is a super key, but not a key because C# is a key.
- Composite keys: more than one attribute. Eg?
- Eg: GRADEREPORT, only key is {St#,SecId}

Keys

- Can have multiple keys: Eg: C#, Cname keys
- Primary Key: One key is picked as primary key. Shown by underlining.
- Which is picked can have an impact
  - Foreign keys can refer only to primary keys
  - File may be sorted by primary key (or index), so searches may be more efficient (622B/524).
- Pick natural and simple one. Eg: C#
- Can also be done through auto-number
- All primary keys for course database ?

Foreign Keys

- Suppose added a new row in GRADEREPORT : (14,117,B). Is this OK ?
- No: because no 117 in section:
  - any section is in GRADEREPORT also has to be in SECTION table.
- Section_id in GRADEREPORT is **foreign key** to Section_id in SECTION :
  - a value for Section_id in GRADEREPORT has to also occur in SECTION
- How to show: using arrows.
- Foreign key, primary keys are glue for diff. tables
- All foreign keys for course database

Foreign Keys

- F. Key may be composite:
  - Eg: if in student table: (fname, lname) P. Key
  - In Gradereport (fname, lname) stored instead of student_number
  - (fname, lname) composite F. Key. Is this different to saying fname F. Key and lname F. Key ?
- F. Key has to be Time invariant: always true
  - Not enough that it is true for one instance
  - Eg: student_number from Student to Gradereport
    - Is this a valid F. Key ?
- No : in other instances may not be the case

Foreign Keys

- F. Key can be null:
  - Eg: suppose had an instructor table also
  - Section 145 has no instructor
  - So either F. Key should be null or exist in referenced table
- F. Key could itself be part of P. key: Eg ?
- In Prerequisite table, Prerequisite_number
- Self referential: F. Key can refer to a P. Key in the *same relation*.
  - Eg: if prerequisite info being kept in Course table

Elmasri Company Database

- The company is organized into DEPARTMENTS.
- Each department has a name, number and an employee who *manages* the department
  - We keep track of the start date of the department manager
  - Departments can have multiple locations
- Each department *controls* a number of PROJECTs.
- Each project has a name, number and is located at a single location.

Elmasri Company Database

- For each EMPLOYEE, we store the social security number, address, salary, sex, and birthdate.
- Employees may have a *supervisor*
  - **DIFFERENT FROM (and no connection to) *manager***
- Each employee *works for* one department but may *work on* several projects.
- We keep track of the number of hours per week that an employee currently works on each project.
- Each employee may have a number of DEPENDENTs.
  - For each dependent, we keep track of their name, sex, birthdate, and relationship to employee.

Elmasri Company Database

- What will be the different tables ?
- What will be the primary keys ?
- What will be the foreign keys ?
- An example instance

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 3.5
Schema diagram for the COMPANY relational database schema.

Elmasri database with FK: Figure 3.7

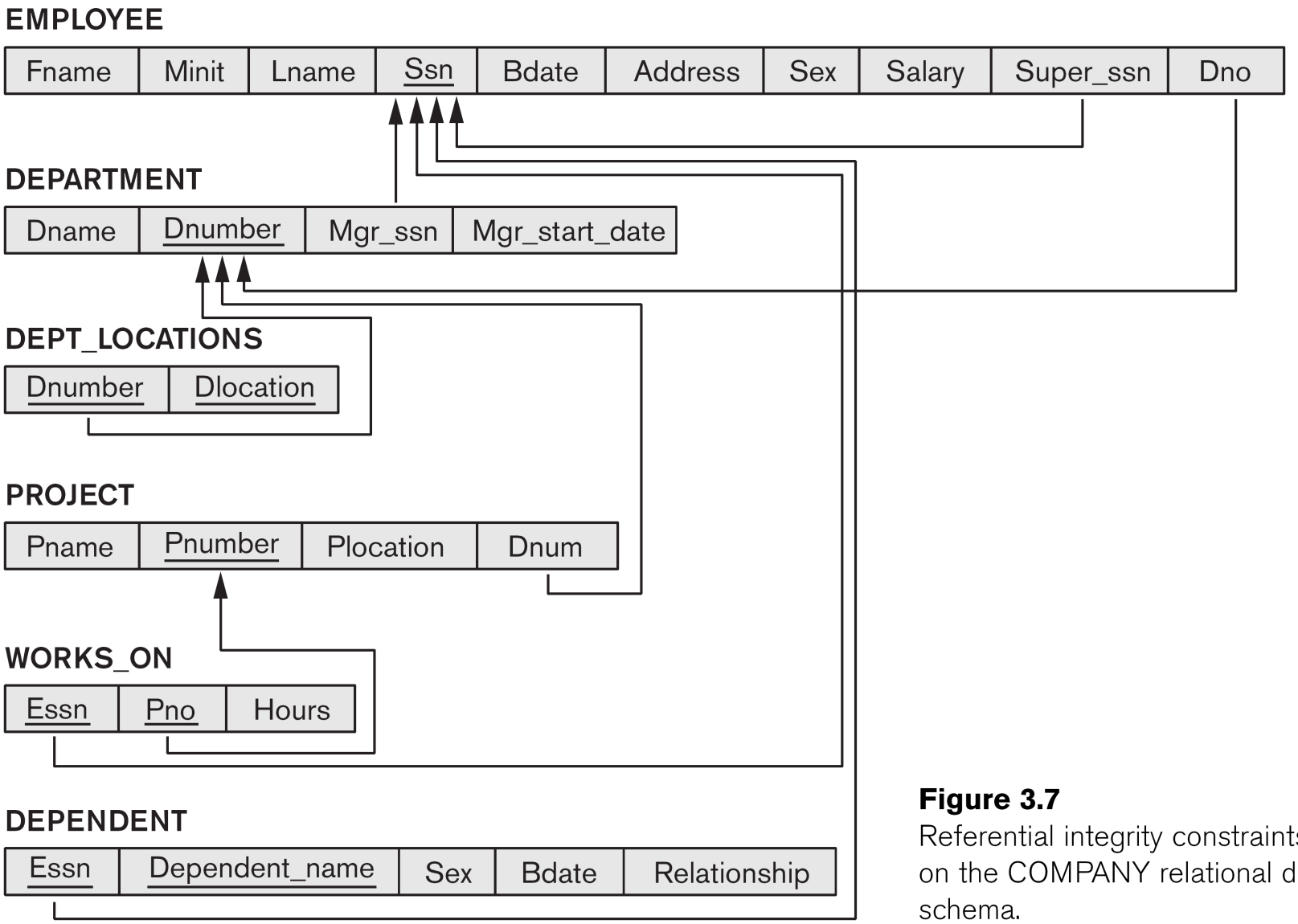


Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

Elmasri Figure 3.6: Relational Instance

One possible database state for the COMPANY relational database schema.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

In class problem: relational schemas

Database keeps track of student enrollment in courses and the books adopted for each course:

STUDENT(SSN, Name, Major, Bdate)

COURSE(Course#, Cname, Dept)

ENROLL(SSN, Course#, Quarter, Grade)

BOOK_ADOPTION(Course#, Quarter, Book_ISBN)

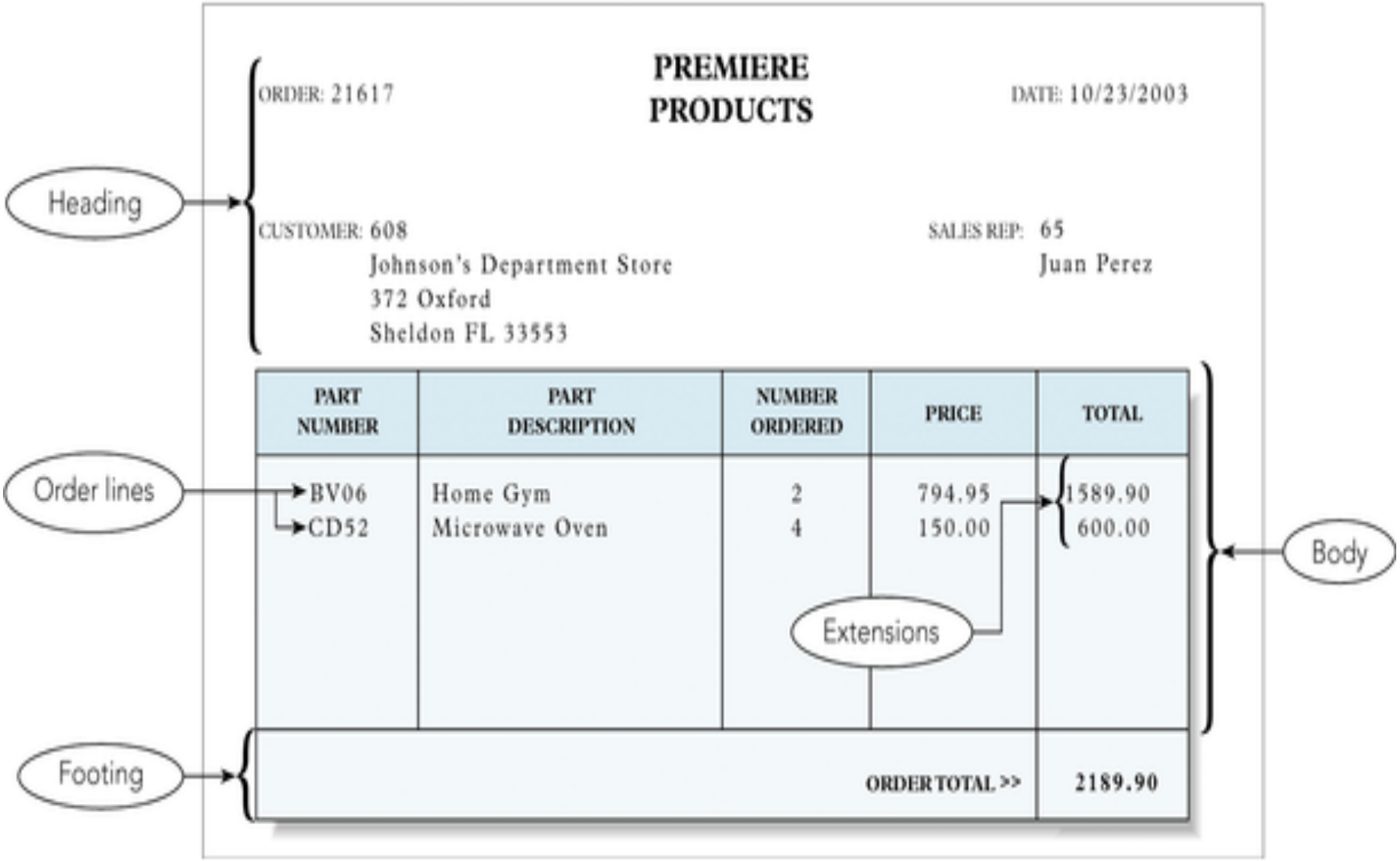
TEXT(Book_ISBN, Book_Title, Publisher, Author)

- Why is Quarter part of P.Key in Enroll ?
- Suppose Book_ISBN was part of P.K. in the BOOK_ADOPTION table
  - How would we interpret that ?
- Draw relational schema specifying foreign keys.

Premiere Products Database: from Pratt and Adamski

- Sales Reps
  - Sales rep number, last name, first name, address, total commission, commission rate
- Customers
  - Customer number, name, address, current balance, credit limit, customer sales rep
- Parts Inventory
  - Part number, description, number units on hand, item class, warehouse number, unit price

Figure 1.2: Premiere Products Sample Order



Premiere Products Customer Order

- Order
  - Order number, order date, customer number
- Order line
  - Order number, part number, number units ordered, unit price
- Overall order total
  - Not stored since it can be calculated

Premiere Products Sample Data

Rep

RepNum	LastName	FirstName	Street	City	State	Zip	Commission	Rate
20	Kaiser	Valerie	624 Randall	Grove	FL	33321	$20,542.50	0.05
35	Hull	Richard	532 Jackson	Sheldon	FL	33553	$39,216.00	0.07
65	Perez	Juan	1626 Taylor	Fillmore	FL	33336	$23,487.00	0.05

Customer

CustomerNum	CustomerName	Street	City	State	Zip	Balance	CreditLimit	RepNum
148	Al's Appliance and Sport	2837 Greenway	Fillmore	FL	33336	$6,550.00	$7,500.00	20
282	Brookings Direct	3827 Devon	Grove	FL	33321	$431.50	$10,000.00	35
356	Ferguson's	382 Wildwood	Northfield	FL	33146	$5,785.00	$7,500.00	65
408	The Everything Shop	1828 Raven	Crystal	FL	33503	$5,285.25	$5,000.00	35
462	Bargains Galore	3829 Central	Grove	FL	33321	$3,412.00	$10,000.00	65
524	Kline's	838 Ridgeland	Fillmore	FL	33336	$12,762.00	$15,000.00	20
608	Johnson's Department Store	372 Oxford	Sheldon	FL	33553	$2,106.00	$10,000.00	65
687	Lee's Sport and Appliance	282 Evergreen	Altonville	FL	32543	$2,851.00	$5,000.00	35
725	Deerfield's Four Seasons	282 Columbia	Sheldon	FL	33553	$248.00	$7,500.00	35
842	All Season	28 Lakeview	Grove	FL	33321	$8,221.00	$7,500.00	20

(continued)

Premiere Products Sample Data

Orders

OrderNum	OrderDate	CustomerNum
21608	10/20/2007	148
21610	10/20/2007	356
21613	10/21/2007	408
21614	10/21/2007	282
21617	10/23/2007	608
21619	10/23/2007	148
21623	10/23/2007	608

OrderLine

OrderNum	PartNum	NumOrdered	QuotedPrice
21608	AT94	11	$21.95
21610	DR93	1	$495.00
21610	DW11	1	$399.99
21613	KL62	4	$329.95
21614	KT03	2	$595.00
21617	BV06	2	$794.95
21617	CD52	4	$150.00
21619	DR93	1	$495.00
21623	KV29	2	$1,290.00

Part

PartNum	Description	OnHand	Class	Warehouse	Price
AT94	Iron	50	HW	3	$24.95
BV06	Home Gym	45	SG	2	$794.95
CD52	Microwave Oven	32	AP	1	$165.00
DL71	Cordless Drill	21	HW	3	$129.95
DR93	Gas Range	8	AP	2	$495.00
DW11	Washer	12	AP	3	$399.99
FD21	Stand Mixer	22	HW	3	$159.95
KL62	Dryer	12	AP	1	$349.95
KT03	Dishwasher	8	AP	3	$595.00
KV29	Treadmill	9	SG	2	$1,390.00