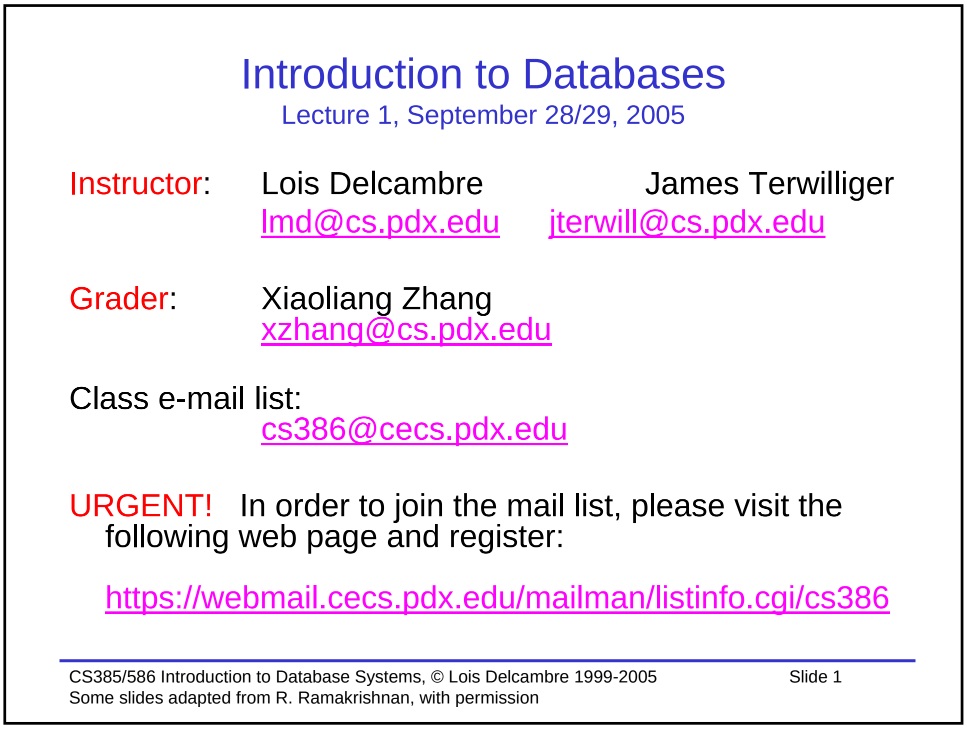


Introduction to Databases

Lecture 1, September 28/29, 2005

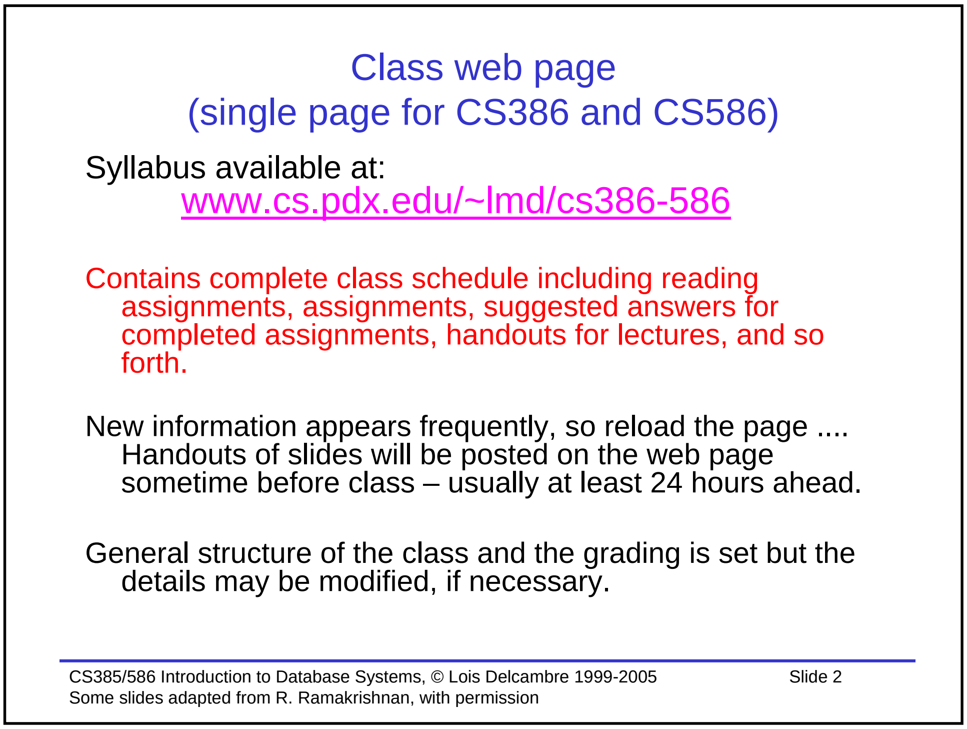
Instructor: Lois Delcambre James Terwilliger
lmd@cs.pdx.edu jterwill@cs.pdx.edu

Grader: Xiaoliang Zhang
xzhang@cs.pdx.edu

Class e-mail list:
cs386@cecs.pdx.edu

URGENT! In order to join the mail list, please visit the following web page and register:

https://webmail.cecs.pdx.edu/mailman/listinfo.cgi/cs386



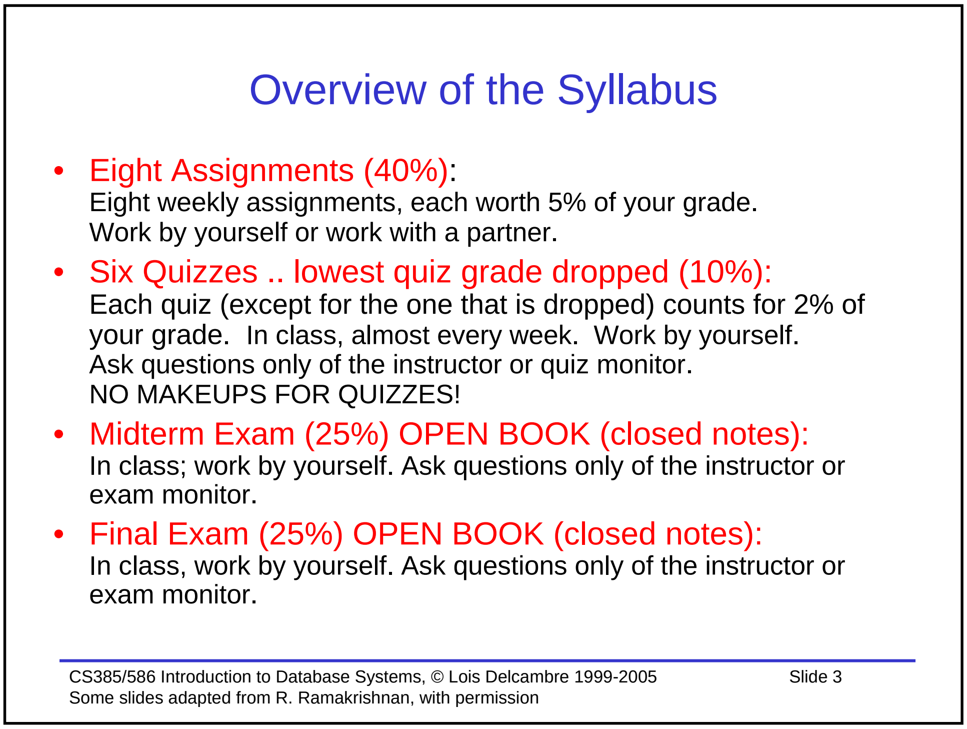
Class web page (single page for CS386 and CS586)

Syllabus available at:
www.cs.pdx.edu/~lmd/cs386-586

Contains complete class schedule including reading assignments, assignments, suggested answers for completed assignments, handouts for lectures, and so forth.

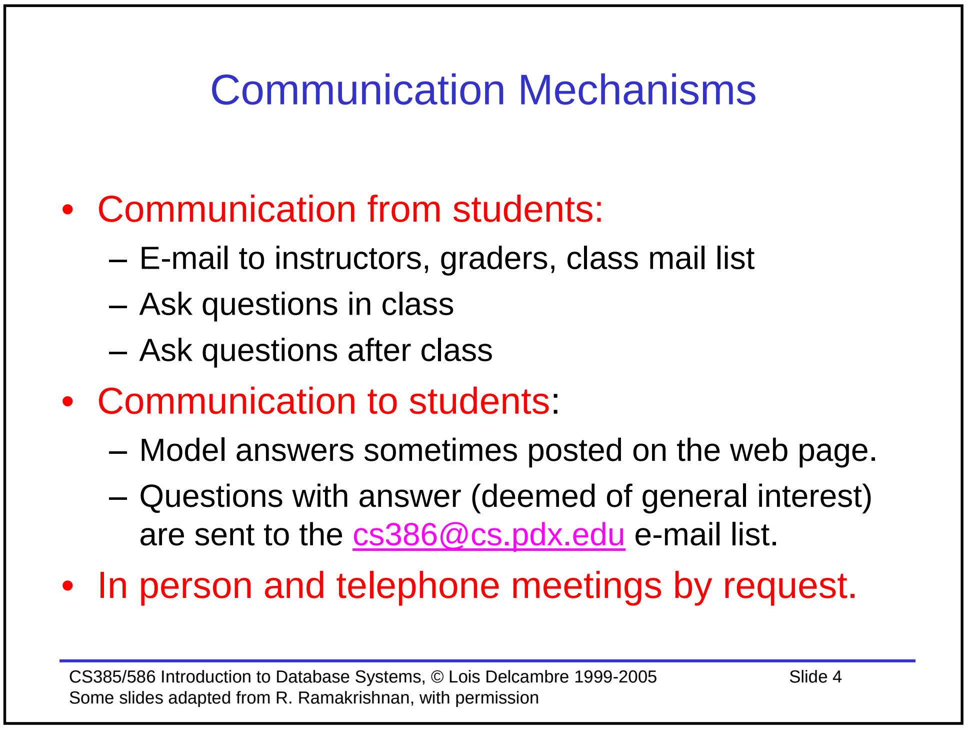
New information appears frequently, so reload the page .... Handouts of slides will be posted on the web page sometime before class – usually at least 24 hours ahead.

General structure of the class and the grading is set but the details may be modified, if necessary.



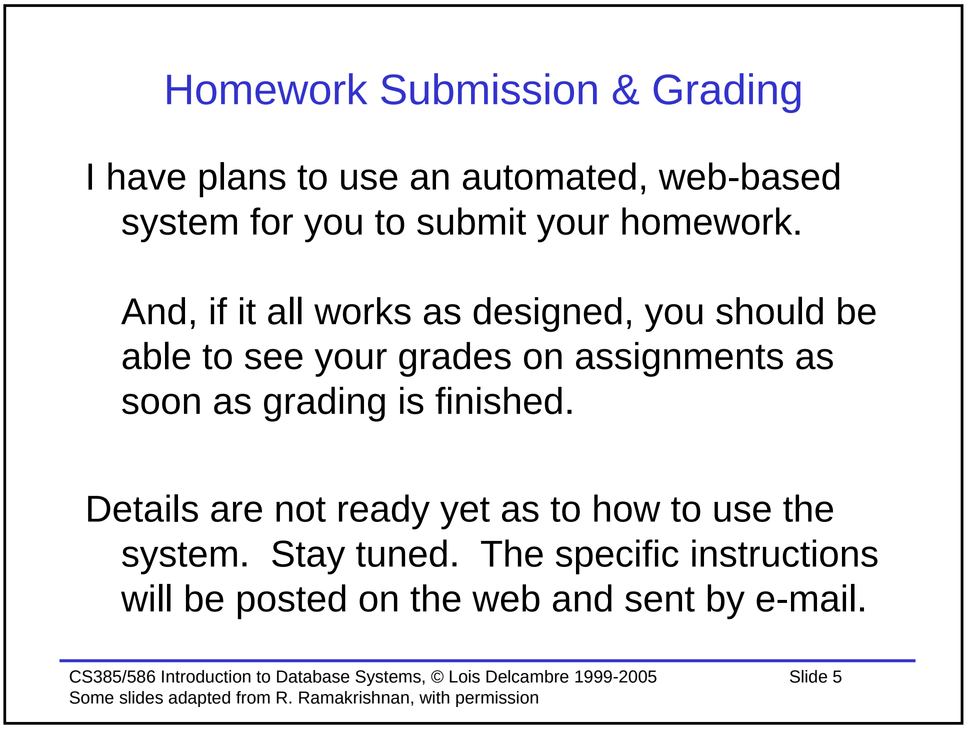
Overview of the Syllabus

- Eight Assignments (40%):
  Eight weekly assignments, each worth 5% of your grade.
  Work by yourself or work with a partner.
- Six Quizzes .. lowest quiz grade dropped (10%):
  Each quiz (except for the one that is dropped) counts for 2% of your grade. In class, almost every week. Work by yourself.
  Ask questions only of the instructor or quiz monitor.
  NO MAKEUPS FOR QUIZZES!
- Midterm Exam (25%) OPEN BOOK (closed notes):
  In class; work by yourself. Ask questions only of the instructor or exam monitor.
- Final Exam (25%) OPEN BOOK (closed notes):
  In class, work by yourself. Ask questions only of the instructor or exam monitor.



Communication Mechanisms

- Communication from students:
  - E-mail to instructors, graders, class mail list
  - Ask questions in class
  - Ask questions after class
- Communication to students:
  - Model answers sometimes posted on the web page.
  - Questions with answer (deemed of general interest) are sent to the cs386@cs.pdx.edu e-mail list.
- In person and telephone meetings by request.

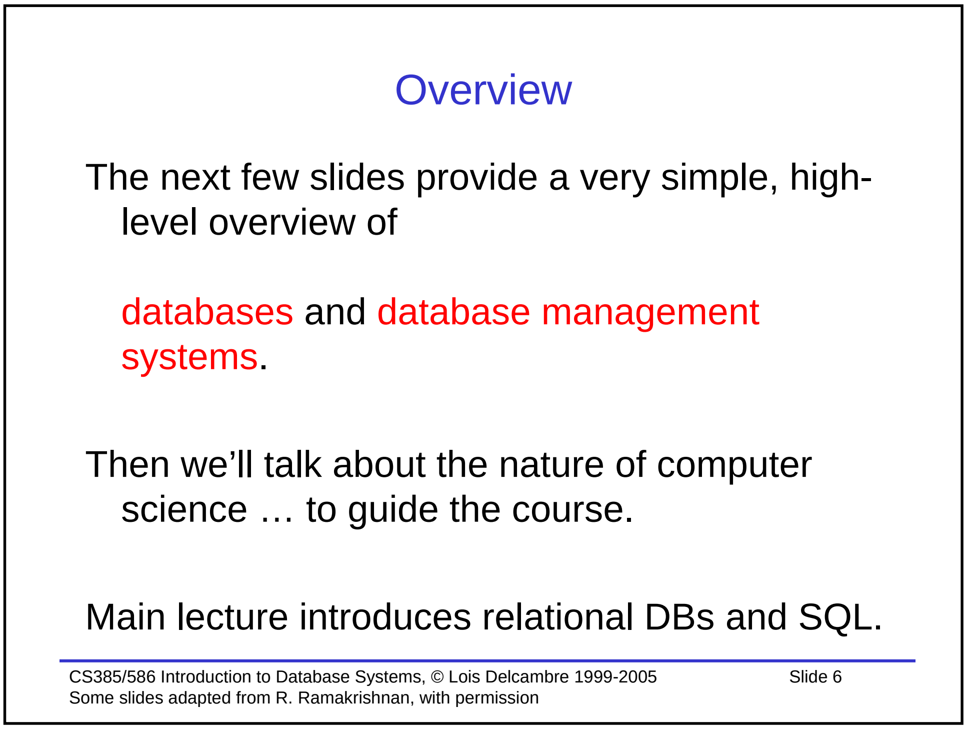


Homework Submission & Grading

I have plans to use an automated, web-based system for you to submit your homework.

And, if it all works as designed, you should be able to see your grades on assignments as soon as grading is finished.

Details are not ready yet as to how to use the system. Stay tuned. The specific instructions will be posted on the web and sent by e-mail.



Overview

The next few slides provide a very simple, high-level overview of

databases and database management systems.

Then we’ll talk about the nature of computer science … to guide the course.

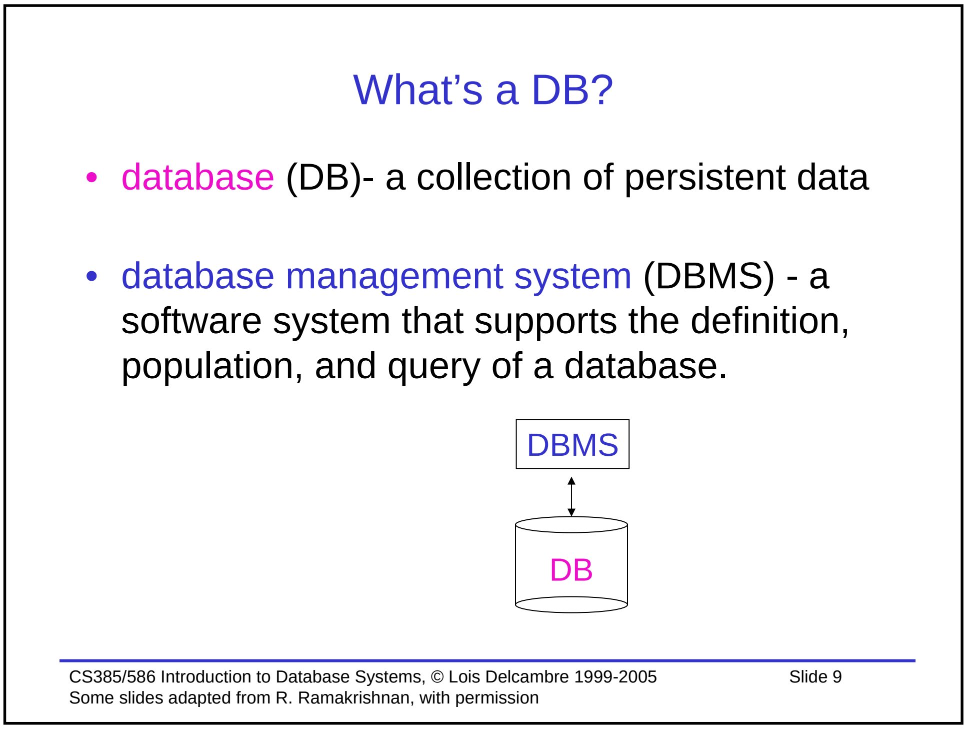
Main lecture introduces relational DBs and SQL.

Why study databases?

- Because data is valuable:
  - often more valuable than the software e.g., bank account records, tax records, …
  - it must be protected - no matter what happens whether we have machine crashes, disk crashes, hurricanes/floods, …
  - It can be combined and summarized in many ways – to serve many different purposes

Why study databases?

- Because the database field has made a number of contributions to basic computer science:
  - because of its focus on data...and disks...
  - because of the formalization of concepts
- Because DBMS software is highly successful as a commercial technology (Oracle, Informix, MS Access…)
- Because DB research is highly active

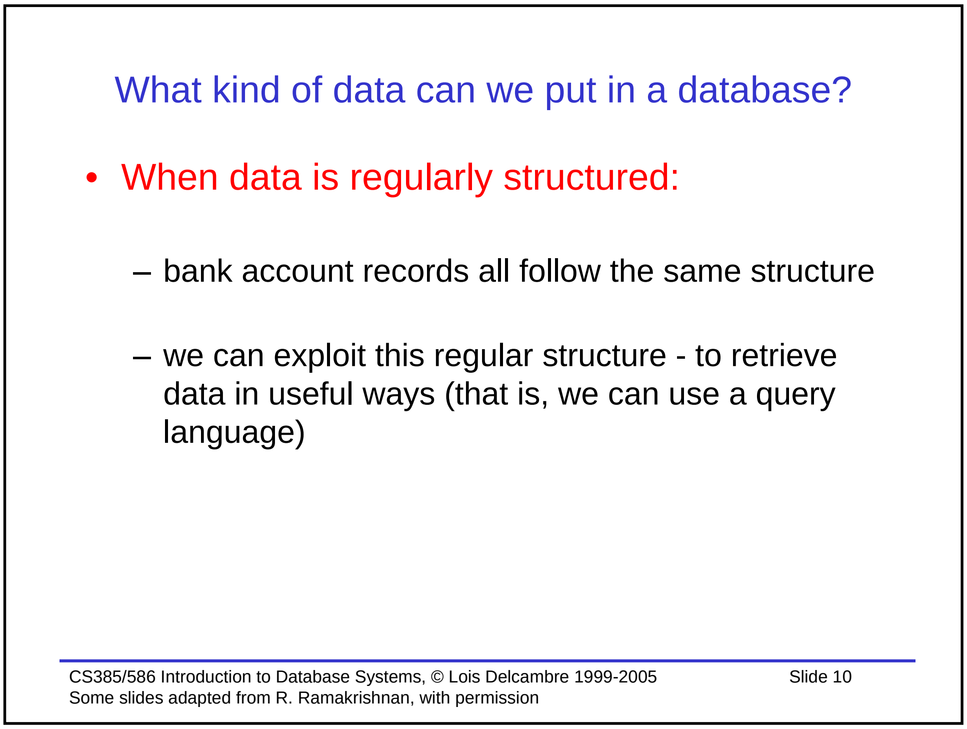


What’s a DB?

- database (DB)- a collection of persistent data
- database management system (DBMS) - a software system that supports the definition, population, and query of a database.

DBMS

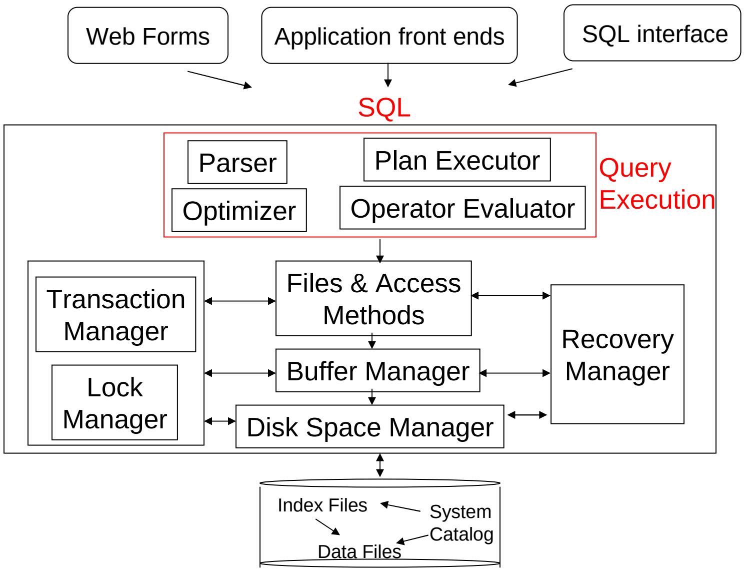
DB



What kind of data can we put in a database?

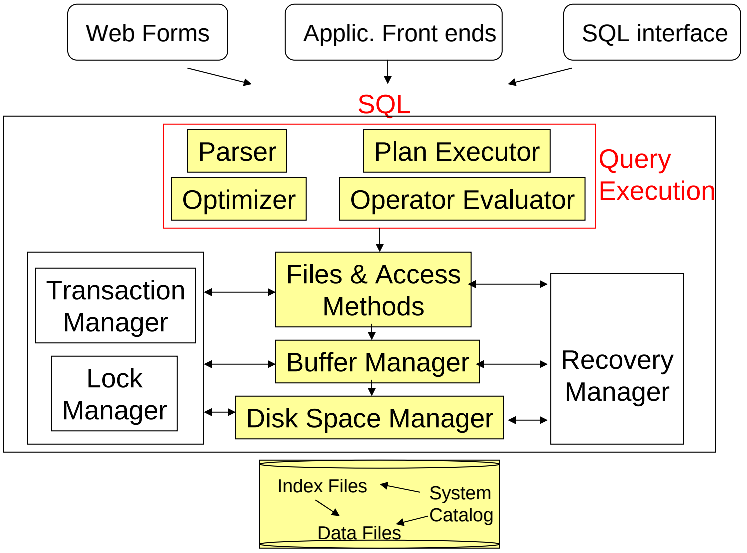
- When data is regularly structured:
  - – bank account records all follow the same structure
  - – we can exploit this regular structure - to retrieve data in useful ways (that is, we can use a query language)

Database Architecture (Figure 1.3, p. 20)

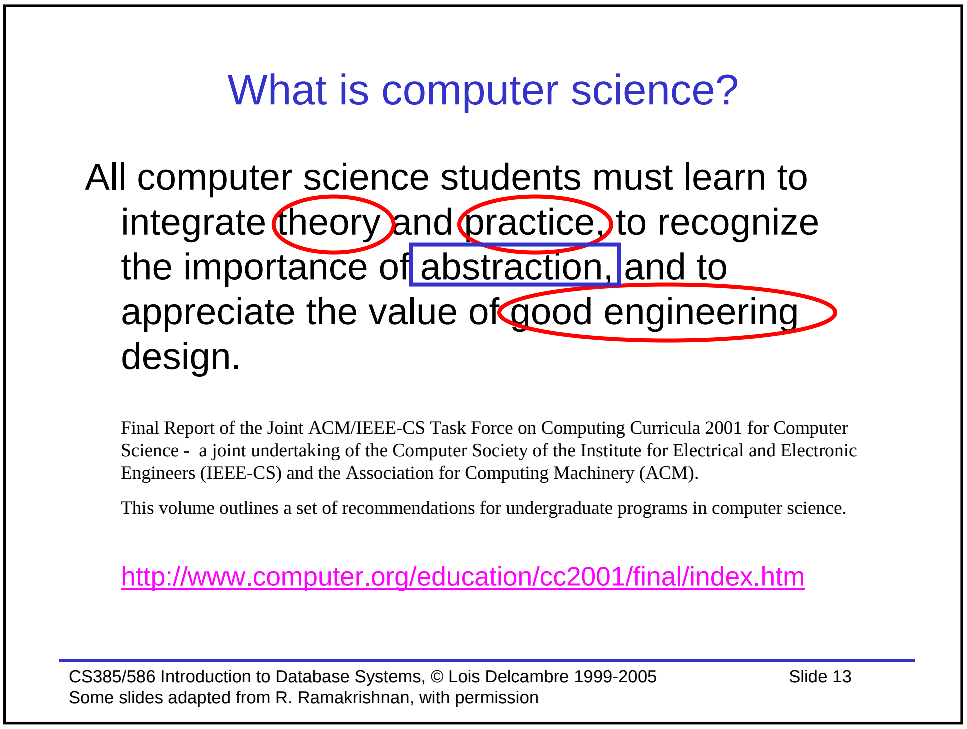


CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 11
Some slides adapted from R. Ramakrishnan, with permission

Query Processing! (shown in yellow)



CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 12
Some slides adapted from R. Ramakrishnan, with permission



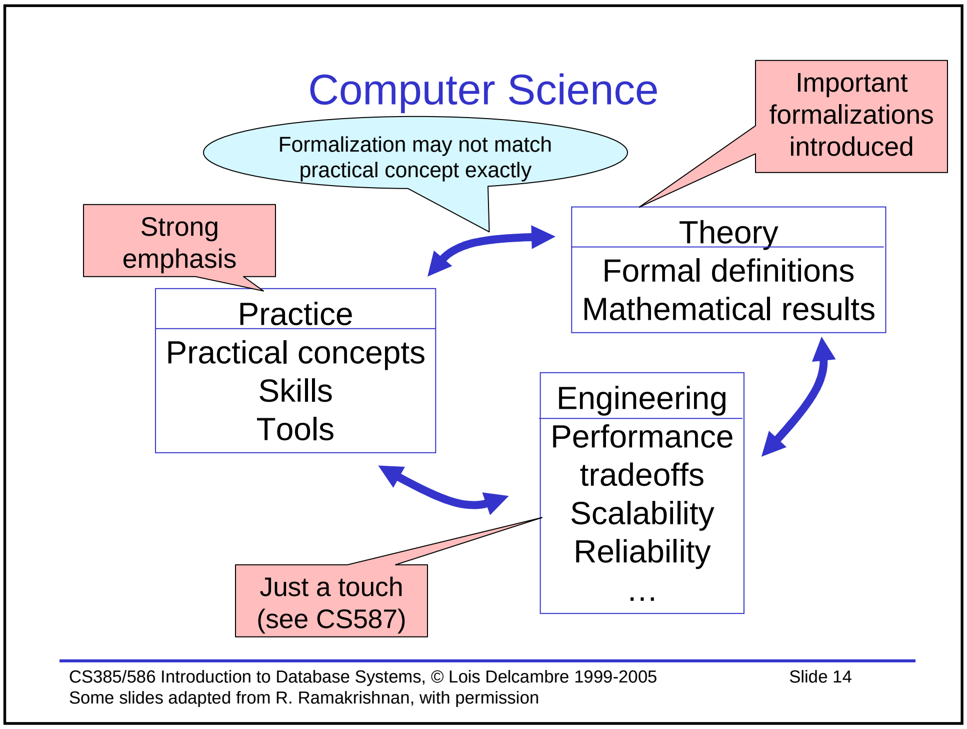
What is computer science?

All computer science students must learn to integrate theory and practice, to recognize the importance of abstraction, and to appreciate the value of good engineering design.

Final Report of the Joint ACM/IEEE-CS Task Force on Computing Curricula 2001 for Computer Science - a joint undertaking of the Computer Society of the Institute for Electrical and Electronic Engineers (IEEE-CS) and the Association for Computing Machinery (ACM).

This volume outlines a set of recommendations for undergraduate programs in computer science.

http://www.computer.org/education/cc2001/final/index.htm



Computer Science

Important formalizations introduced

Formalization may not match practical concept exactly

Strong emphasis

Theory
Formal definitions
Mathematical results

Practice
Practical concepts
Skills
Tools

Engineering
Performance tradeoffs
Scalability
Reliability
…

Just a touch (see CS587)

Tonight's Lecture

- Introduce:
  - Database terminology
  - Difference between schema and data
  - SQL query language
  - Relational data model

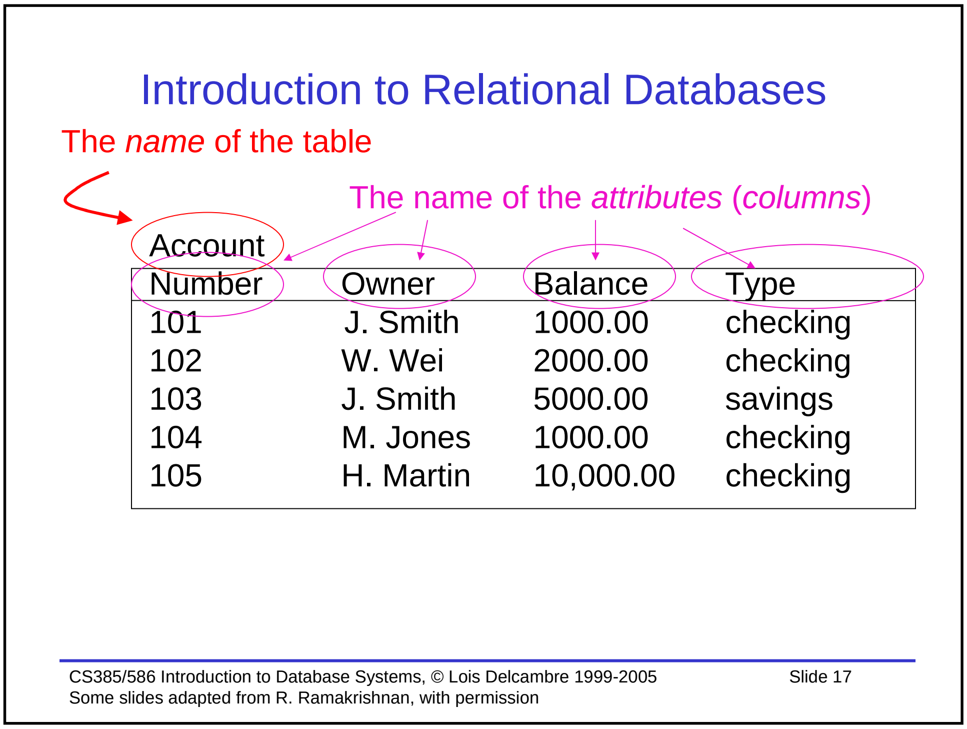
  from a practical point of view (only, for tonight)!

Introduction to Relational Databases

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 has been defined to help keep track of bank accounts.



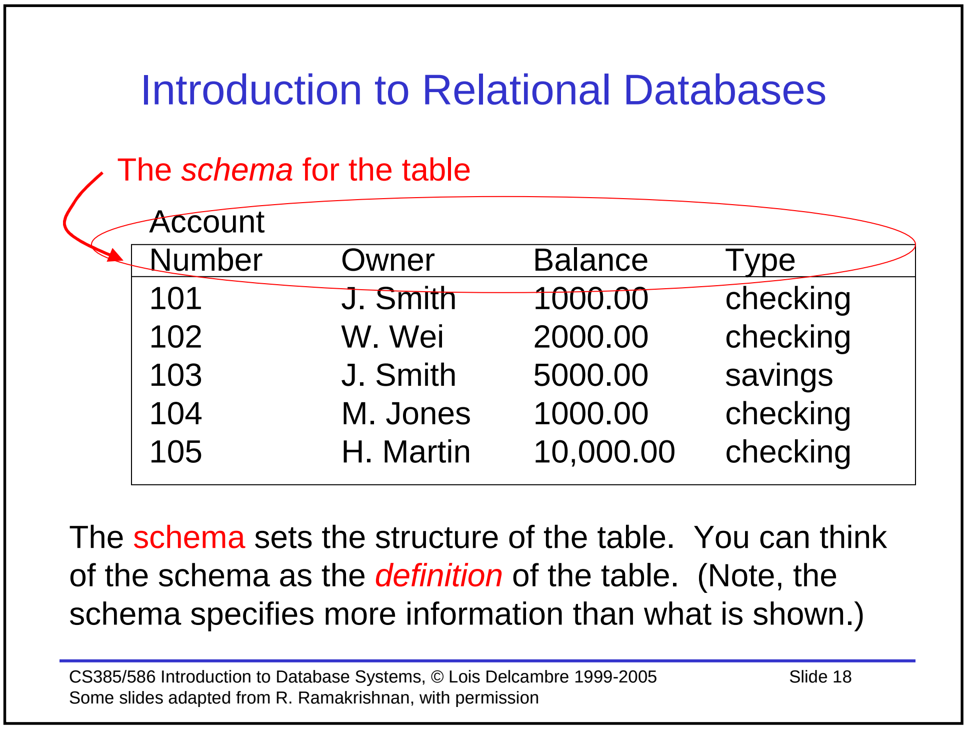
Introduction to Relational Databases

The *name* of the table

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

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

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 17
Some slides adapted from R. Ramakrishnan, with permission



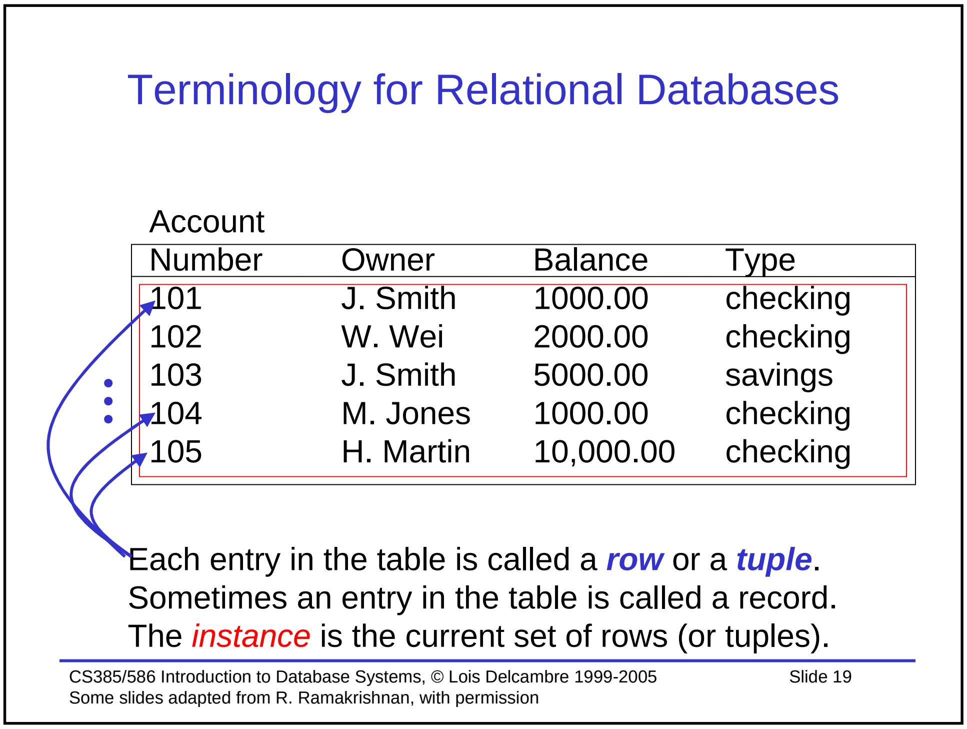
Introduction to Relational Databases

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.)

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 18
Some slides adapted from R. Ramakrishnan, with permission

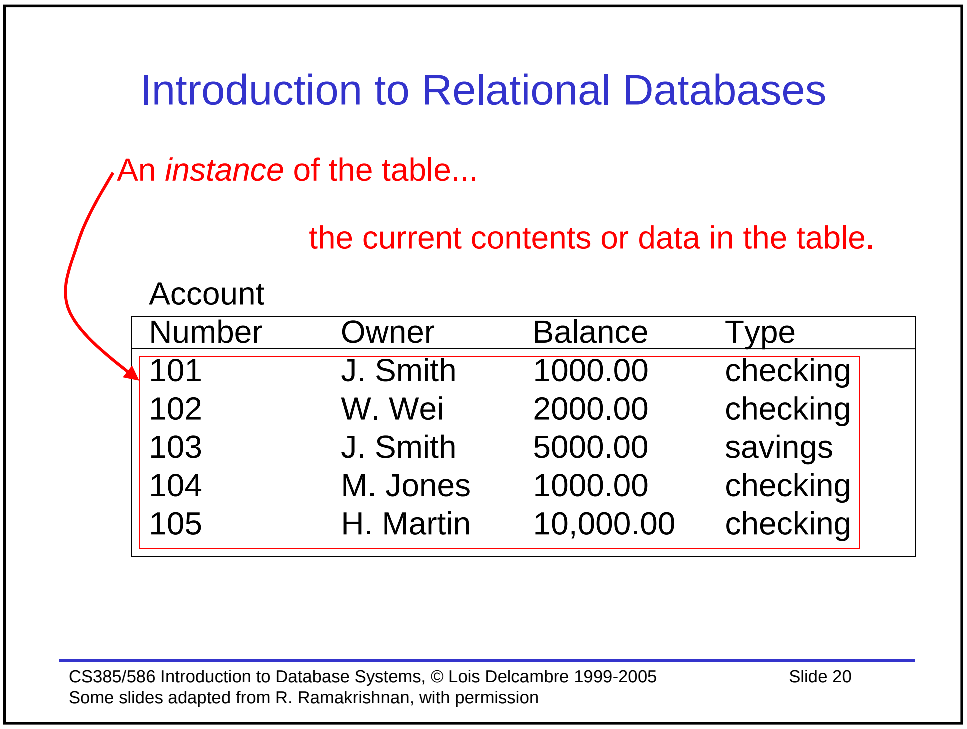


Terminology for Relational Databases

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*** or a ***tuple***.
Sometimes an entry in the table is called a record.
The *instance* is the current set of rows (or tuples).

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 19
Some slides adapted from R. Ramakrishnan, with permission



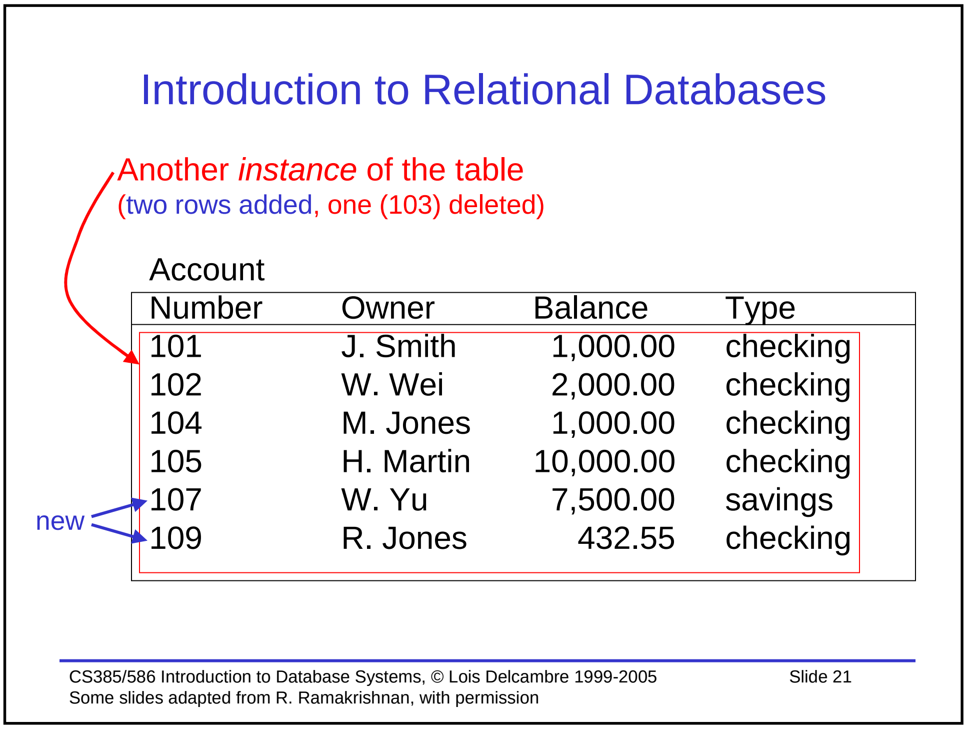
Introduction to Relational Databases

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

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 20
Some slides adapted from R. Ramakrishnan, with permission



Introduction to Relational Databases

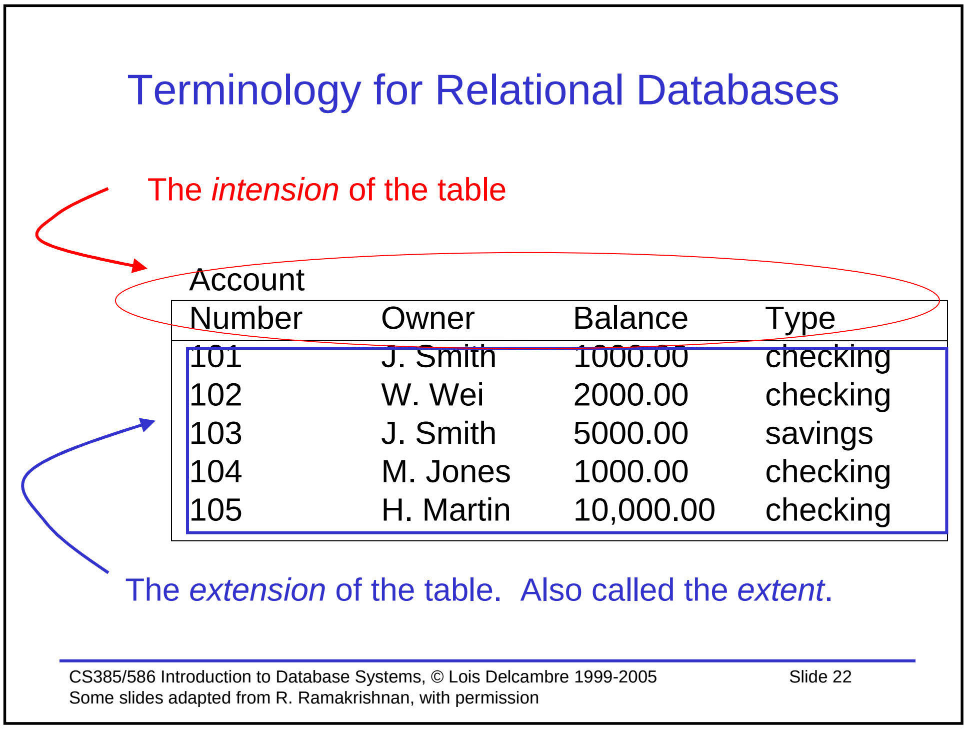
Another *instance* of the table
(two rows added, one (103) deleted)

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

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 21
Some slides adapted from R. Ramakrishnan, with permission



Terminology for Relational Databases

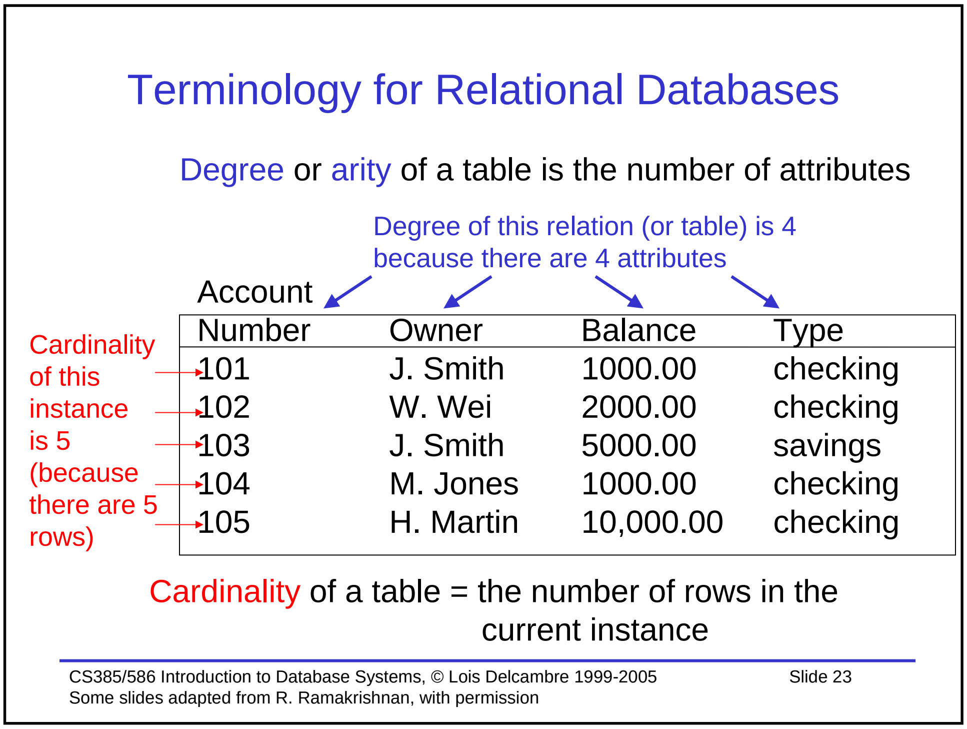
The *intension* of the table

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*.

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 22
Some slides adapted from R. Ramakrishnan, with permission



Terminology for Relational Databases

Degree or arity of a table is the number of attributes

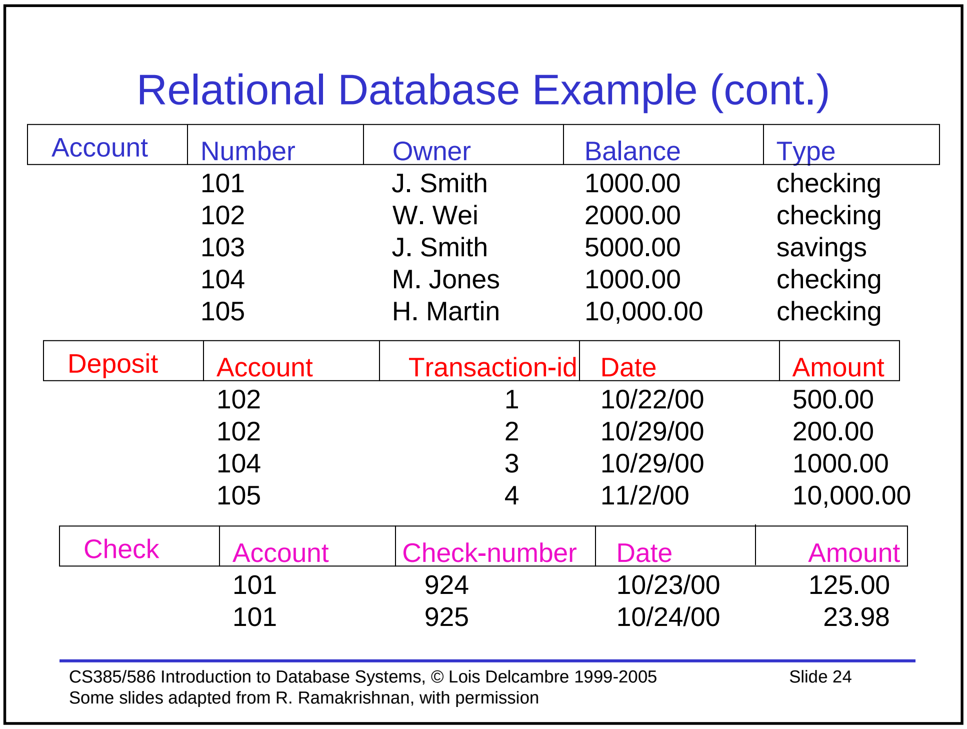
Degree of this relation (or table) is 4
because there are 4 attributes

Cardinality of this instance is 5 (because there are 5 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

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

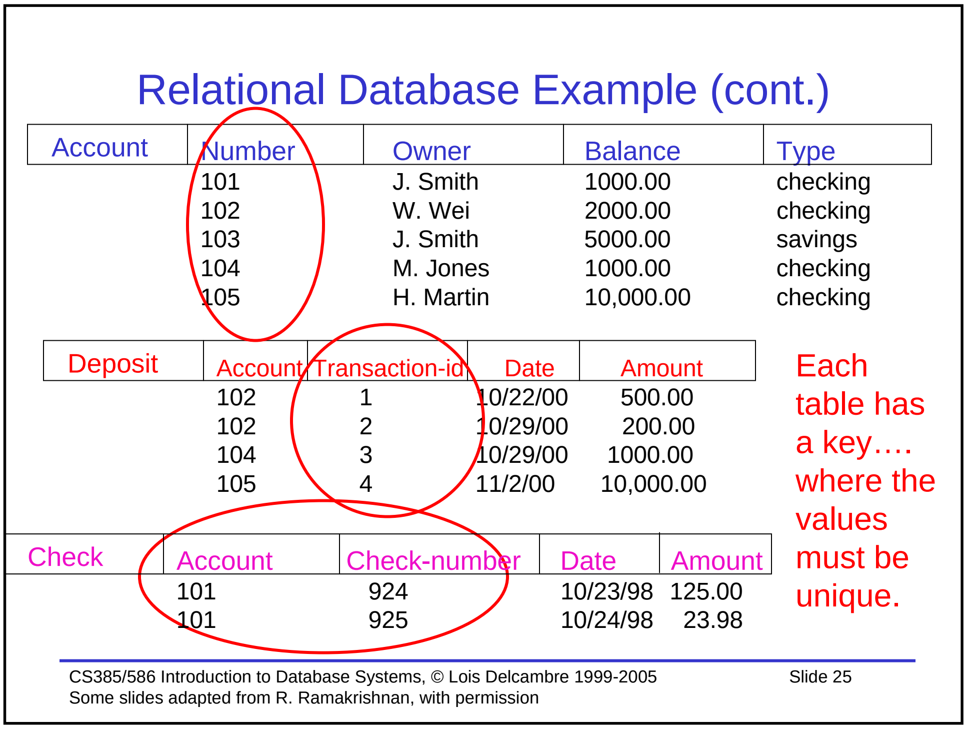


Relational Database Example (cont.)

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	Account	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	11/2/00	10,000.00

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



Relational Database Example (cont.)

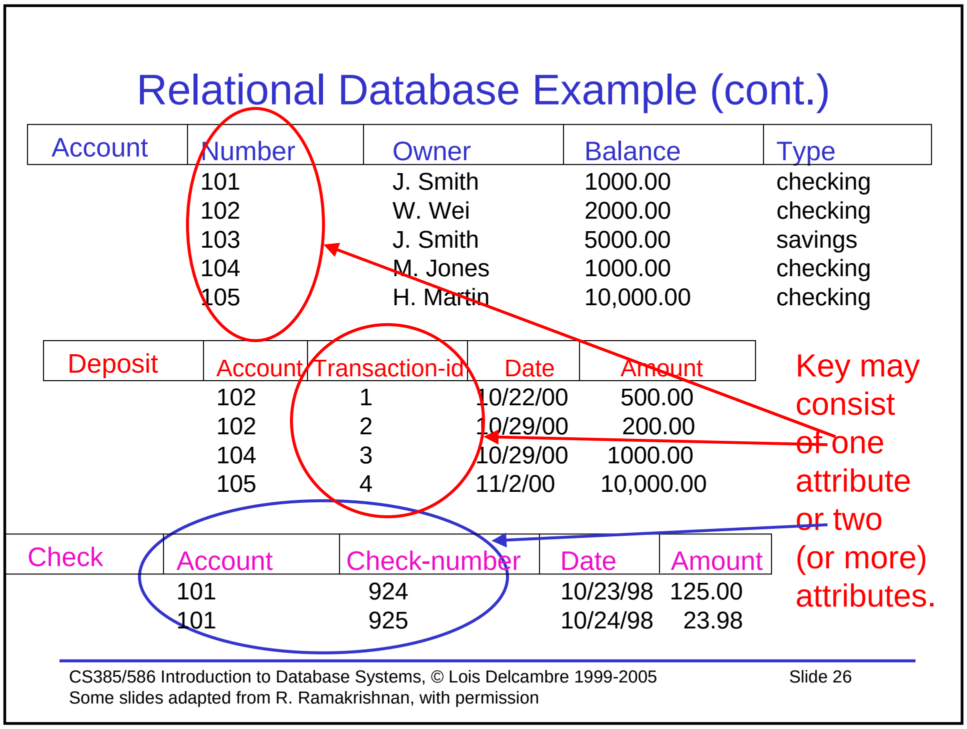
Account	Number	Owner	Balance	Type
	101	J. Smith	1000.00	checking
	102	W. Wei	2000.00	checking
	103	J. Smith	5000.00	savings
	104	M. Jones	1000.00	checking
	105	H. Martin	10,000.00	checking

Deposit	Account	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	11/2/00	10,000.00

Check	Account	Check-number	Date	Amount
	101	924	10/23/98	125.00
	101	925	10/24/98	23.98

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

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 25
Some slides adapted from R. Ramakrishnan, with permission



Relational Database Example (cont.)

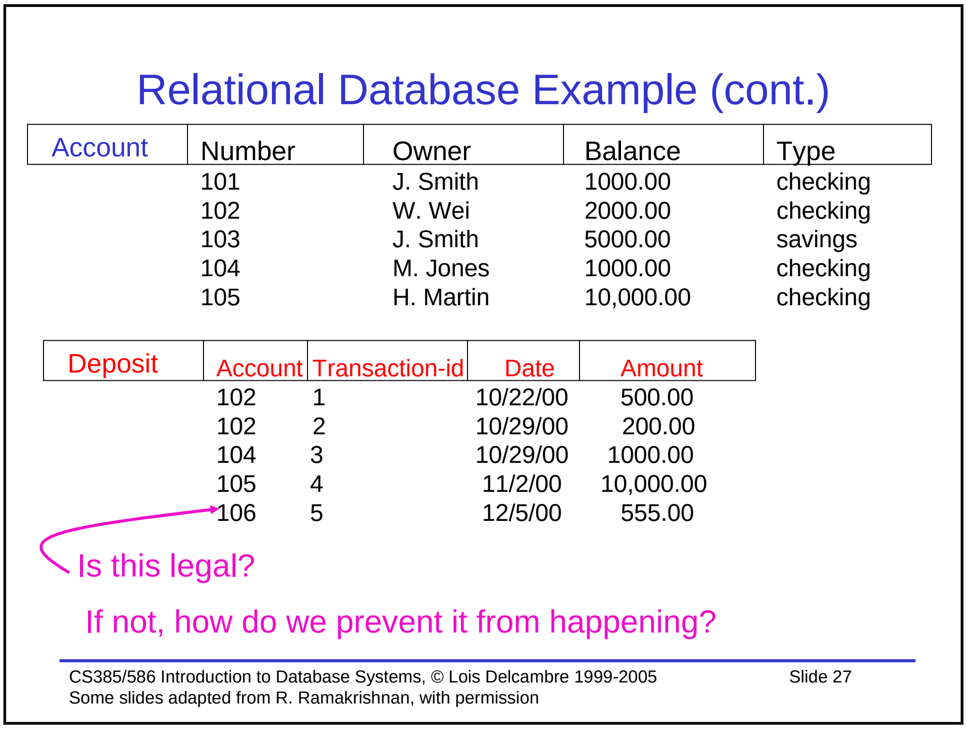
Account	Number	Owner	Balance	Type
	101	J. Smith	1000.00	checking
	102	W. Wei	2000.00	checking
	103	J. Smith	5000.00	savings
	104	M. Jones	1000.00	checking
	105	H. Martin	10,000.00	checking

Deposit	Account	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	11/2/00	10,000.00

Check	Account	Check-number	Date	Amount
	101	924	10/23/98	125.00
	101	925	10/24/98	23.98

Key may consist of one attribute or two (or more) attributes.

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 26
Some slides adapted from R. Ramakrishnan, with permission



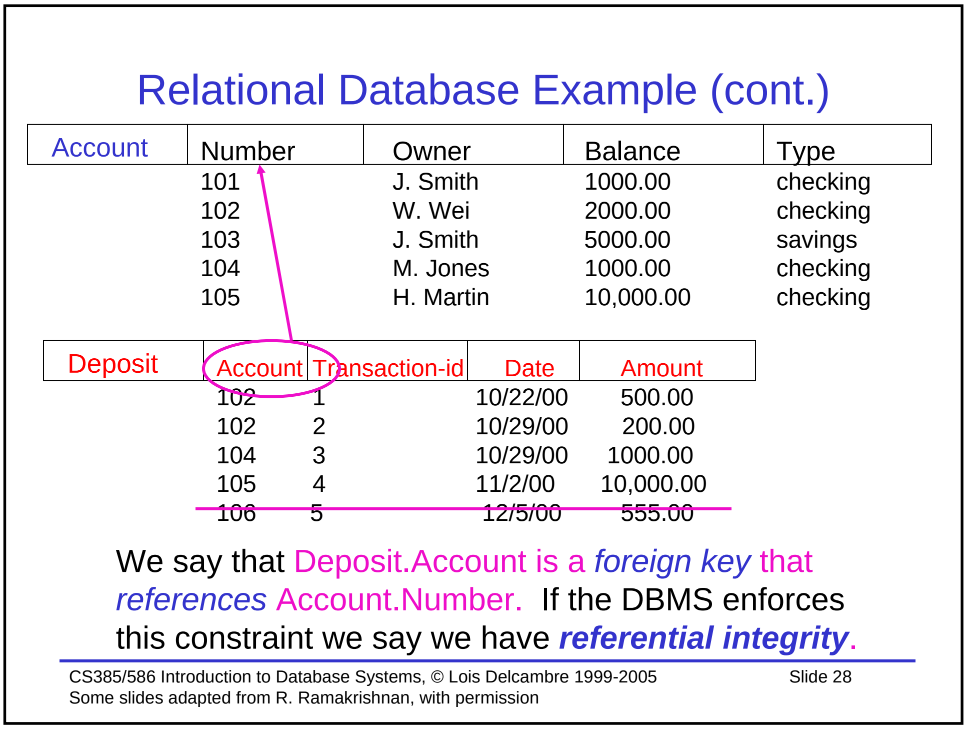
Relational Database Example (cont.)

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	Account	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	11/2/00	10,000.00
	106	5	12/5/00	555.00

Is this legal?

If not, how do we prevent it from happening?

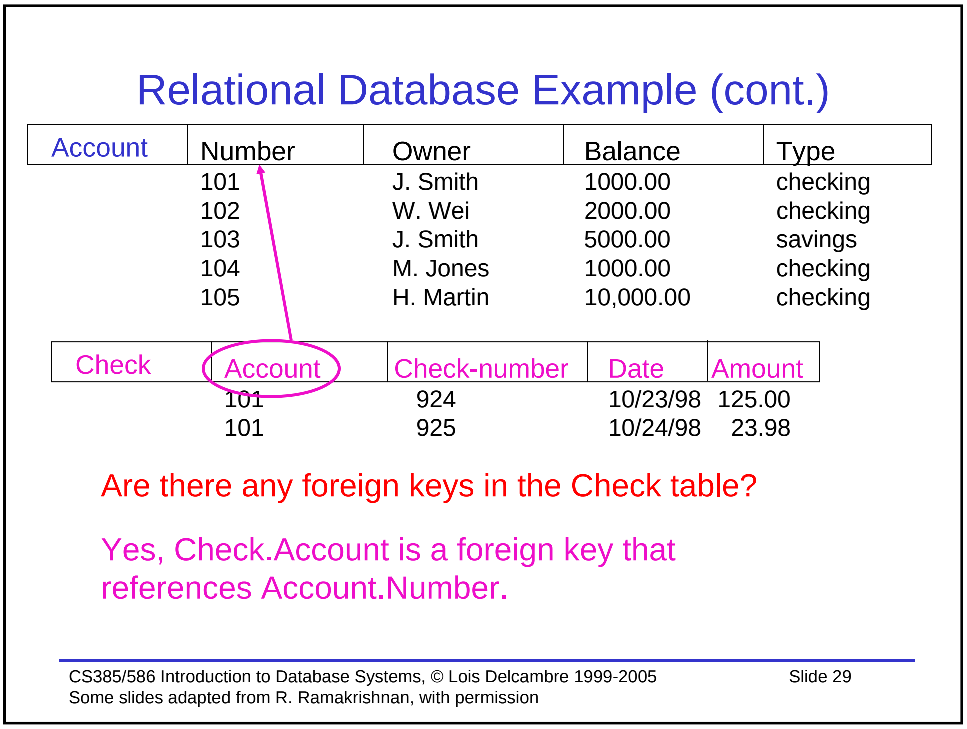


Relational Database Example (cont.)

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	Account	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	11/2/00	10,000.00
	~~106~~	~~5~~	~~12/5/00~~	~~555.00~~

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



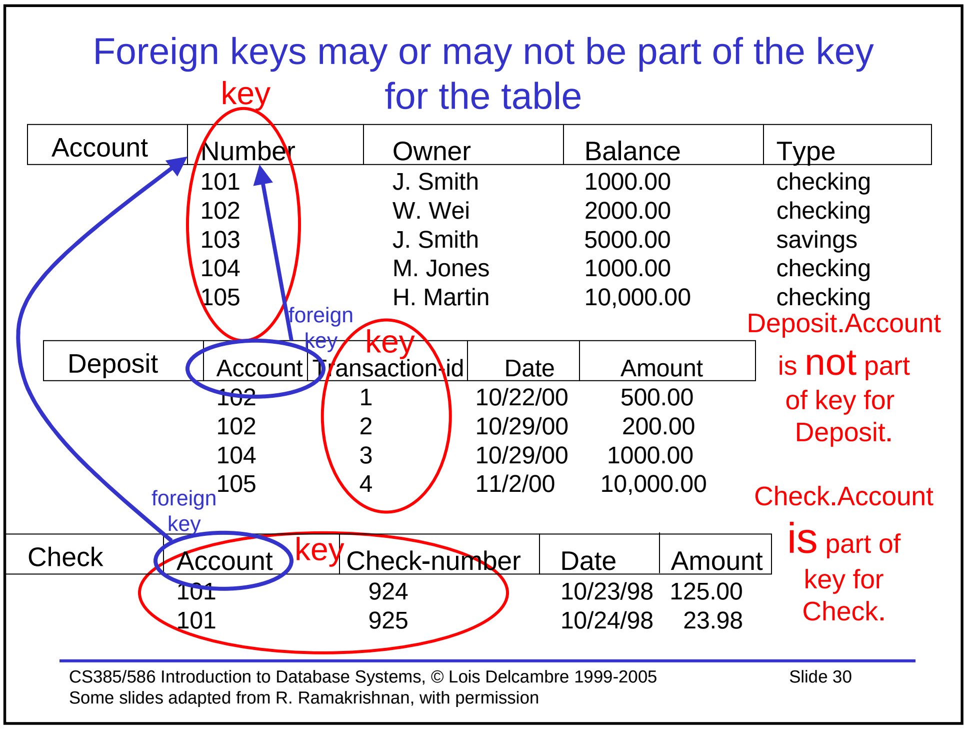
Relational Database Example (cont.)

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

Check	Account	Check-number	Date	Amount
	101	924	10/23/98	125.00
	101	925	10/24/98	23.98

Are there any foreign keys in the Check table?

Yes, Check.Account is a foreign key that references Account.Number.



Foreign keys may or may not be part of the key 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

Deposit	Account	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	11/2/00	10,000.00

Check	Account	Check-number	Date	Amount
	101	924	10/23/98	125.00
	101	925	10/24/98	23.98

Deposit.Account is not part of key for Deposit.

Check.Account is part of key for Check.

Keys for a Table

Consider the following sample data from a table:

1	Jones	28	$50,000.00

Can you tell what the key for this table is?

Keys for a Table

Consider the following sample data from a table:

1	Jones	28	$50,000
2	Smith	28	$60,000

Can you tell what the key for this table is?

Keys for a Table

One possibility:

Person Table with Id as the key

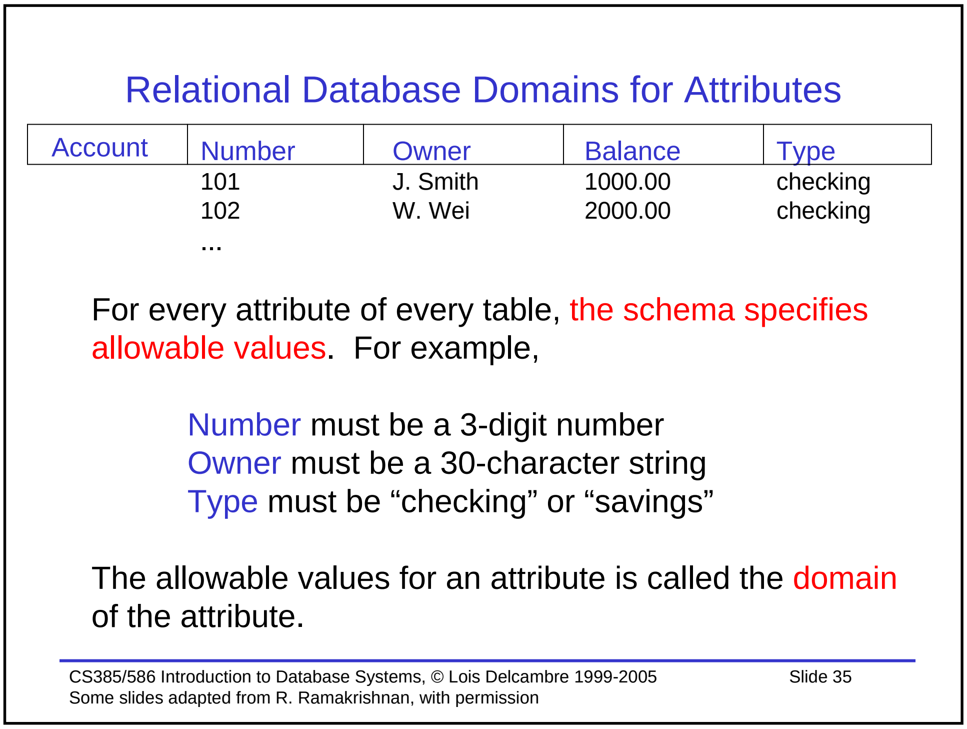
Id	Name	Age	Salary
1	Jones	28	$50,000
2	Smith	28	$60,000

Keys, Table Names, Attribute Names
Tell us what the table is

Another possibility:

Sales Commission Table, by client company, per day

Salesperson	Company	Day	Commission
1	Jones	28	$50,000
2	Smith	28	$60,000



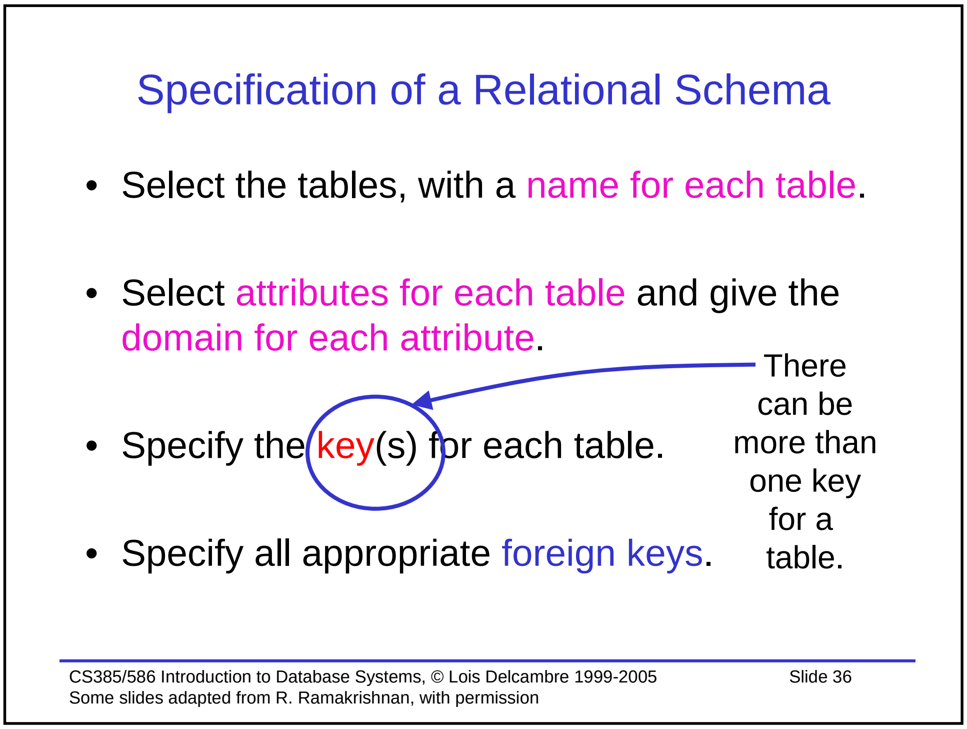
Relational Database Domains for Attributes

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

For every attribute 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”

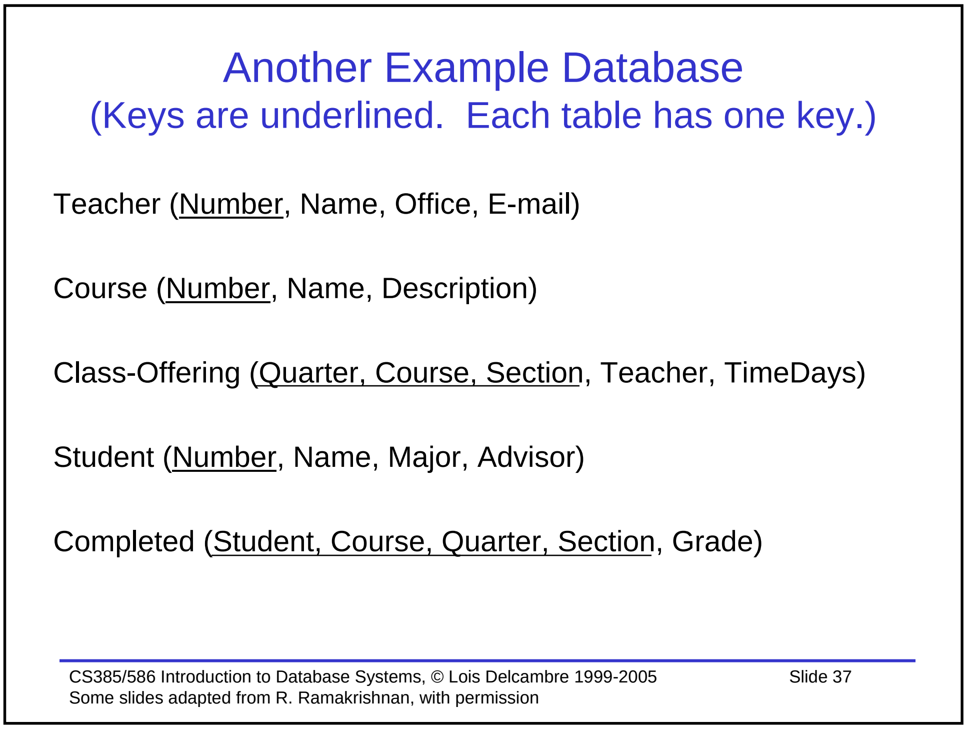
The allowable values for an attribute is called the domain of the attribute.



Specification of a Relational Schema

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

There can be more than one key for a table.



Another Example Database
(Keys are underlined. Each table has one key.)

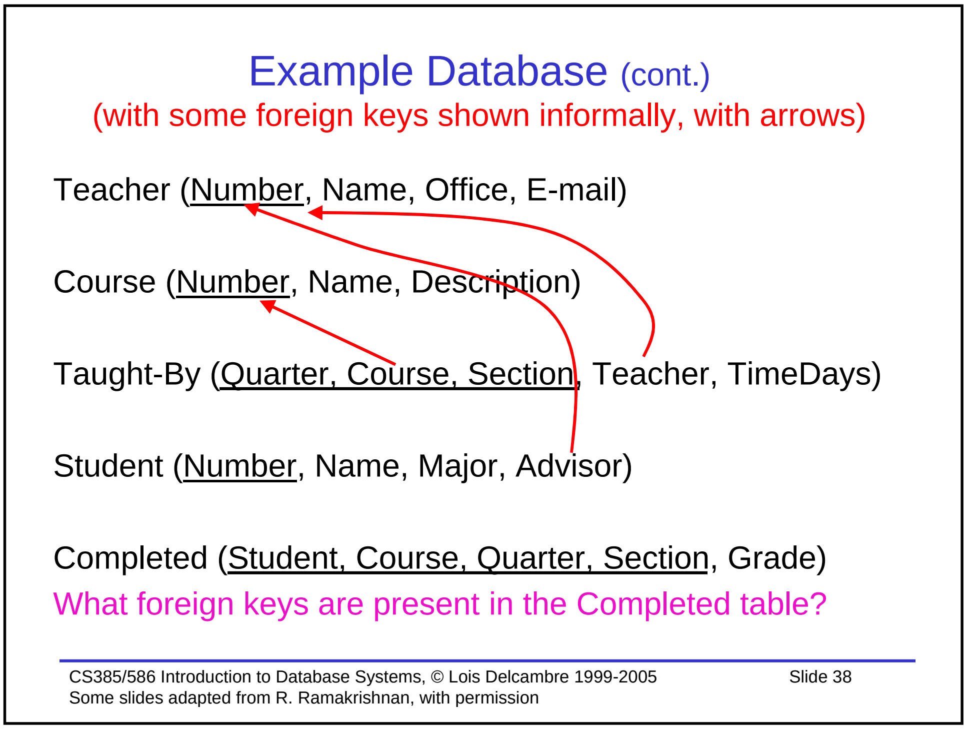
Teacher (Number, Name, Office, E-mail)

Course (Number, Name, Description)

Class-Offering (Quarter, Course, Section, Teacher, TimeDays)

Student (Number, Name, Major, Advisor)

Completed (Student, Course, Quarter, Section, Grade)



Example Database (cont.)
(with some foreign keys shown informally, with arrows)

Teacher (Number, Name, Office, E-mail)

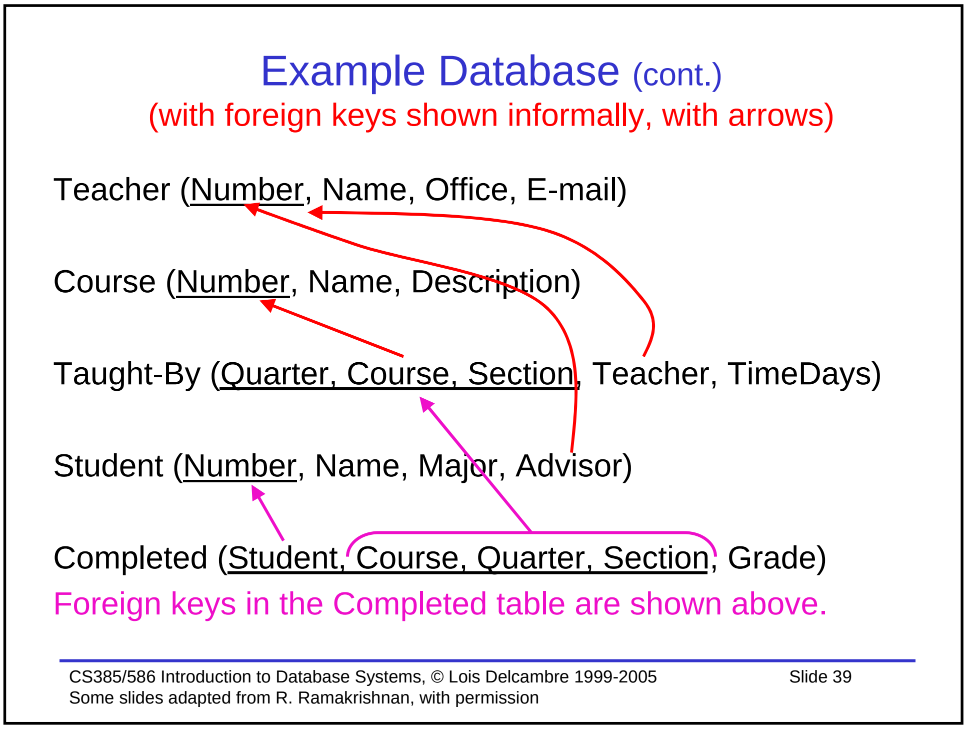
Course (Number, Name, Description)

Taught-By (Quarter, Course, Section, Teacher, TimeDays)

Student (Number, Name, Major, Advisor)

Completed (Student, Course, Quarter, Section, Grade)

What foreign keys are present in the Completed table?



Example Database (cont.)

(with foreign keys shown informally, with arrows)

Teacher (Number, Name, Office, E-mail)

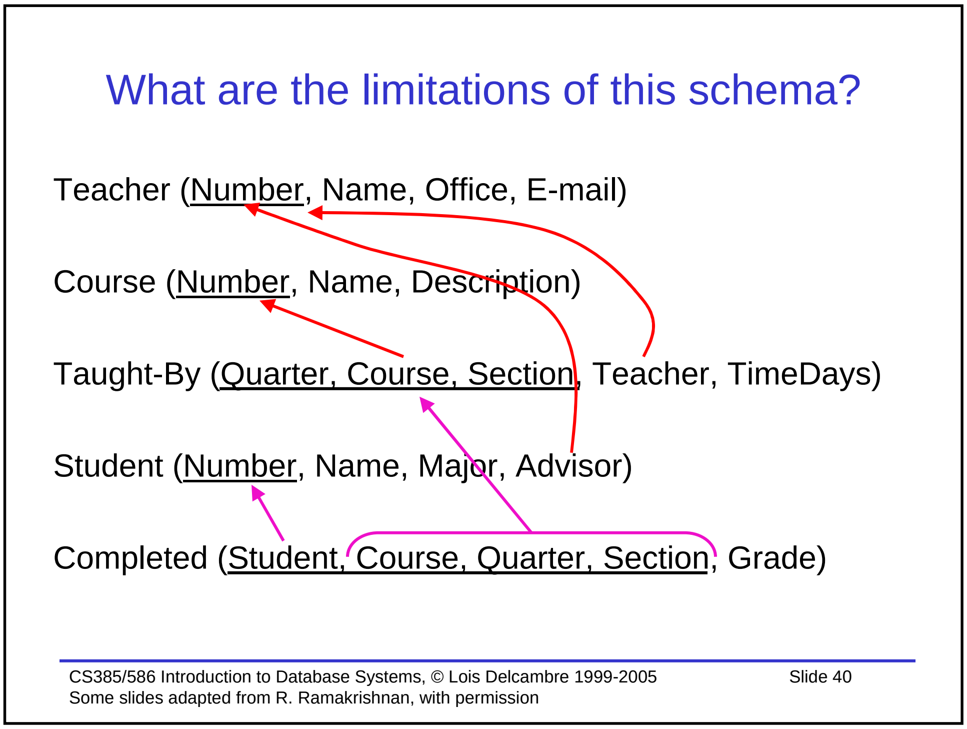
Course (Number, Name, Description)

Taught-By (Quarter, Course, Section, Teacher, TimeDays)

Student (Number, Name, Major, Advisor)

Completed (Student, Course, Quarter, Section, Grade)

Foreign keys in the Completed table are shown above.



What are the limitations of this schema?

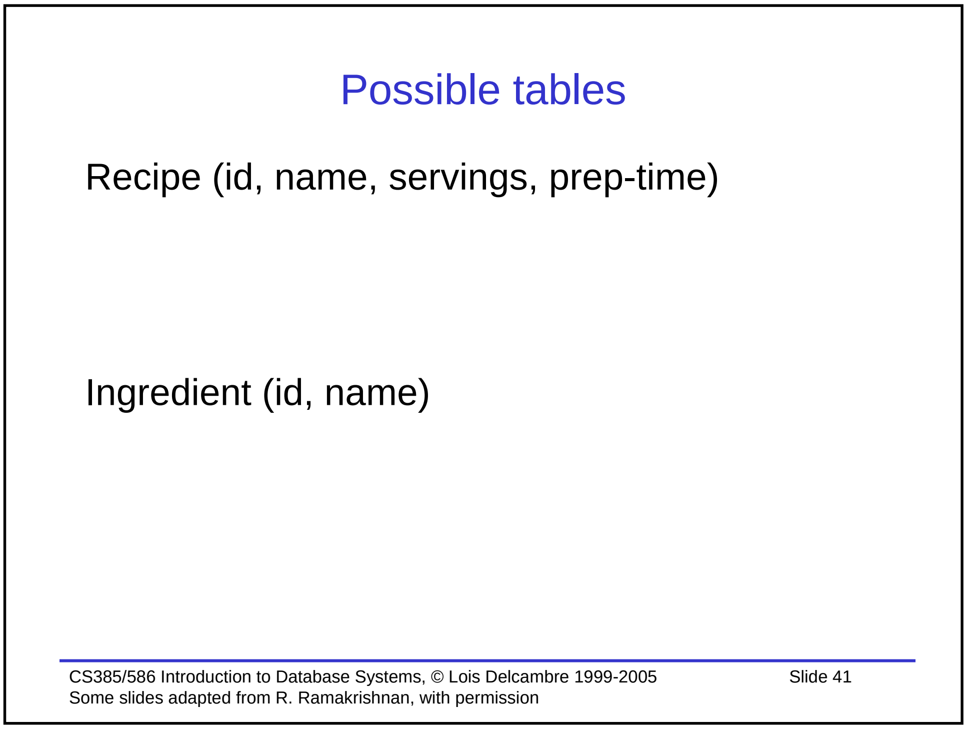
Teacher (Number, Name, Office, E-mail)

Course (Number, Name, Description)

Taught-By (Quarter, Course, Section, Teacher, TimeDays)

Student (Number, Name, Major, Advisor)

Completed (Student, Course, Quarter, Section, Grade)

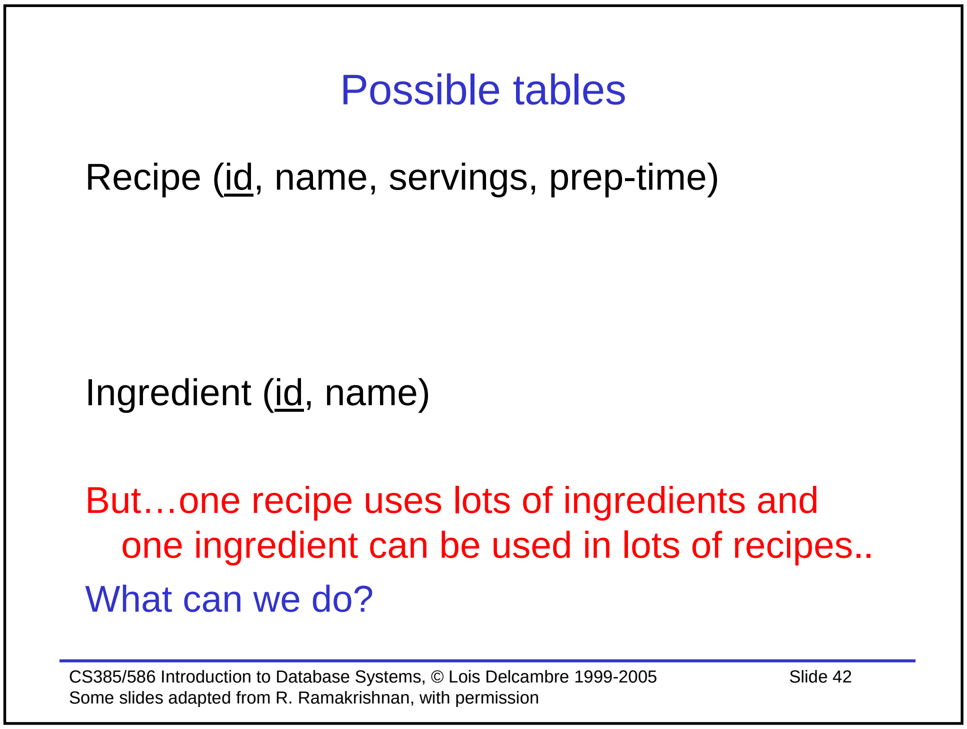


Possible tables

Recipe (id, name, servings, prep-time)

Ingredient (id, name)

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 41
Some slides adapted from R. Ramakrishnan, with permission



Possible tables

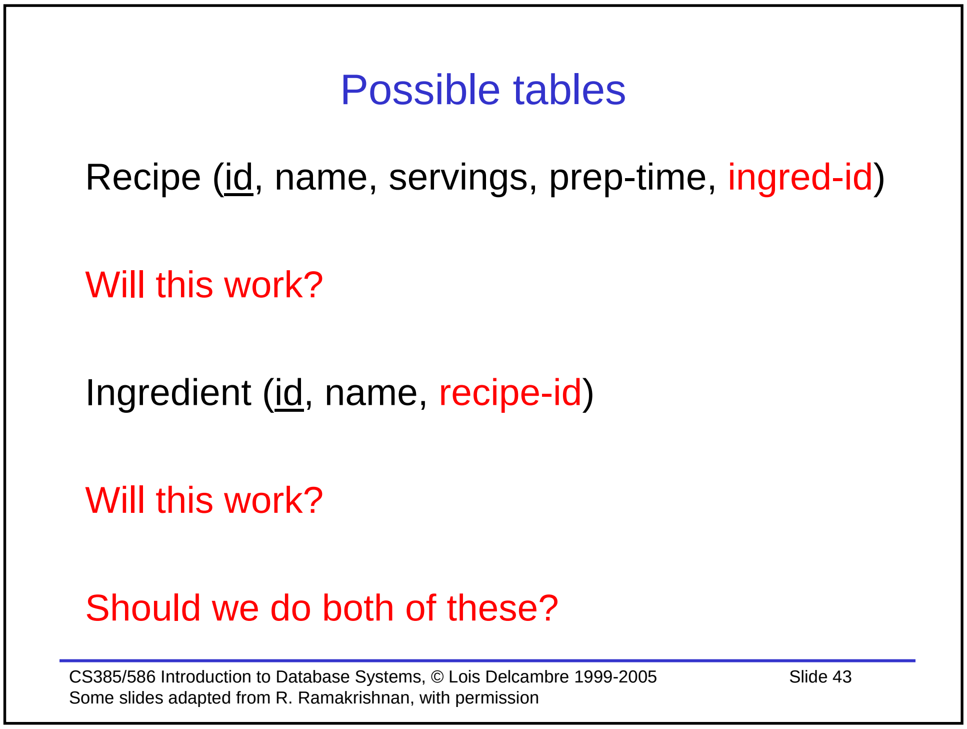
Recipe (id, name, servings, prep-time)

Ingredient (id, name)

But…one recipe uses lots of ingredients and one ingredient can be used in lots of recipes..

What can we do?

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 42
Some slides adapted from R. Ramakrishnan, with permission



Possible tables

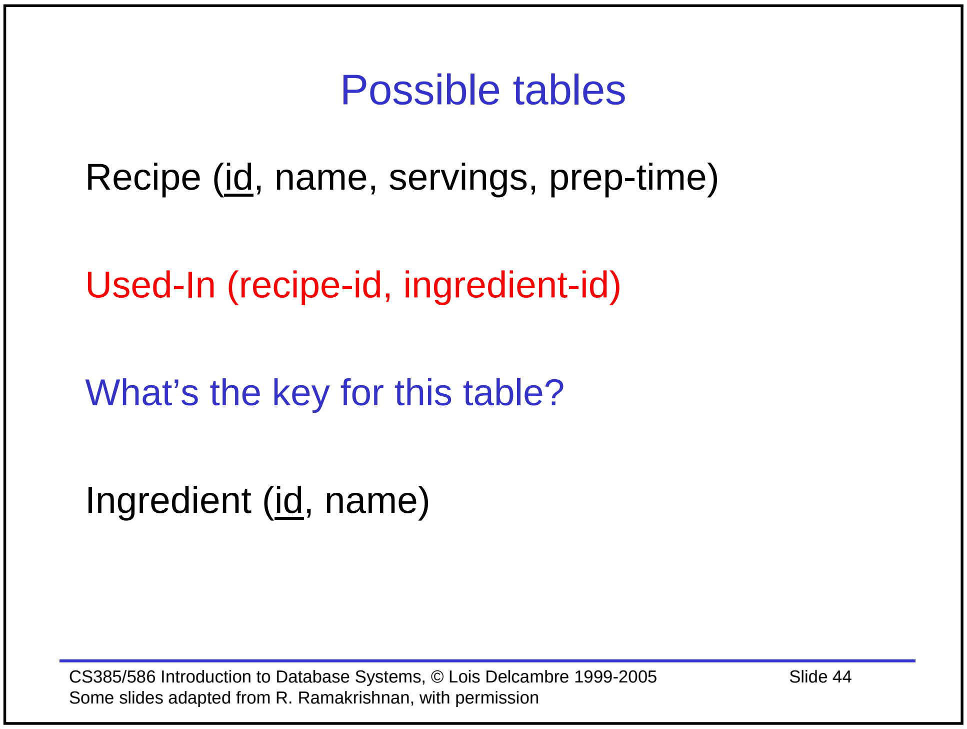
Recipe (id, name, servings, prep-time, ingred-id)

Will this work?

Ingredient (id, name, recipe-id)

Will this work?

Should we do both of these?



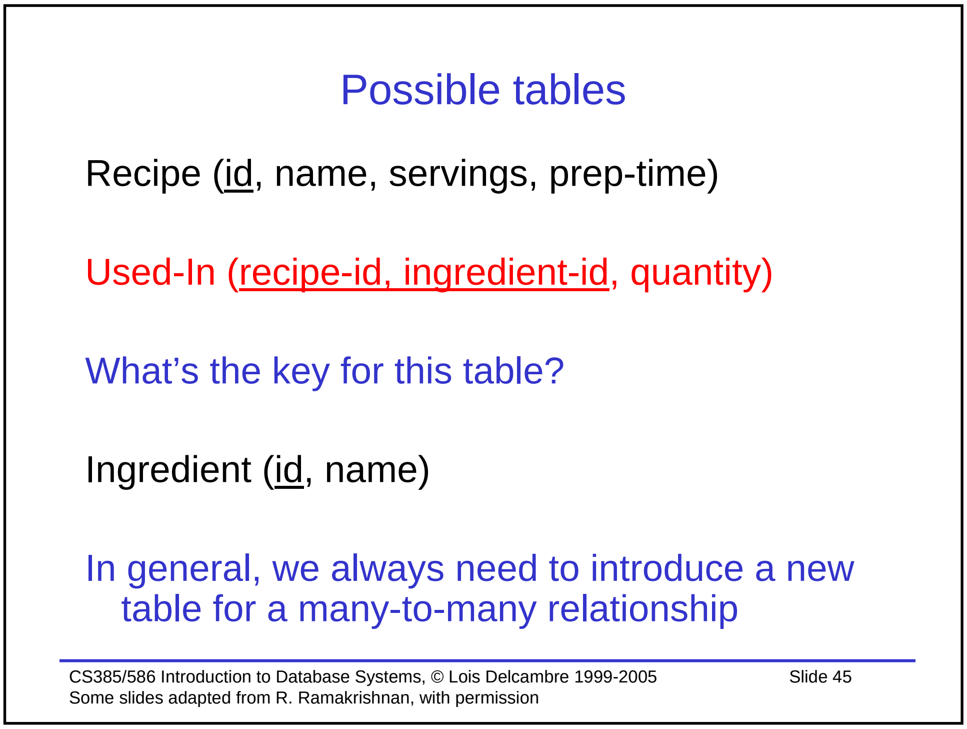
Possible tables

Recipe (id, name, servings, prep-time)

Used-In (recipe-id, ingredient-id)

What’s the key for this table?

Ingredient (id, name)



Possible tables

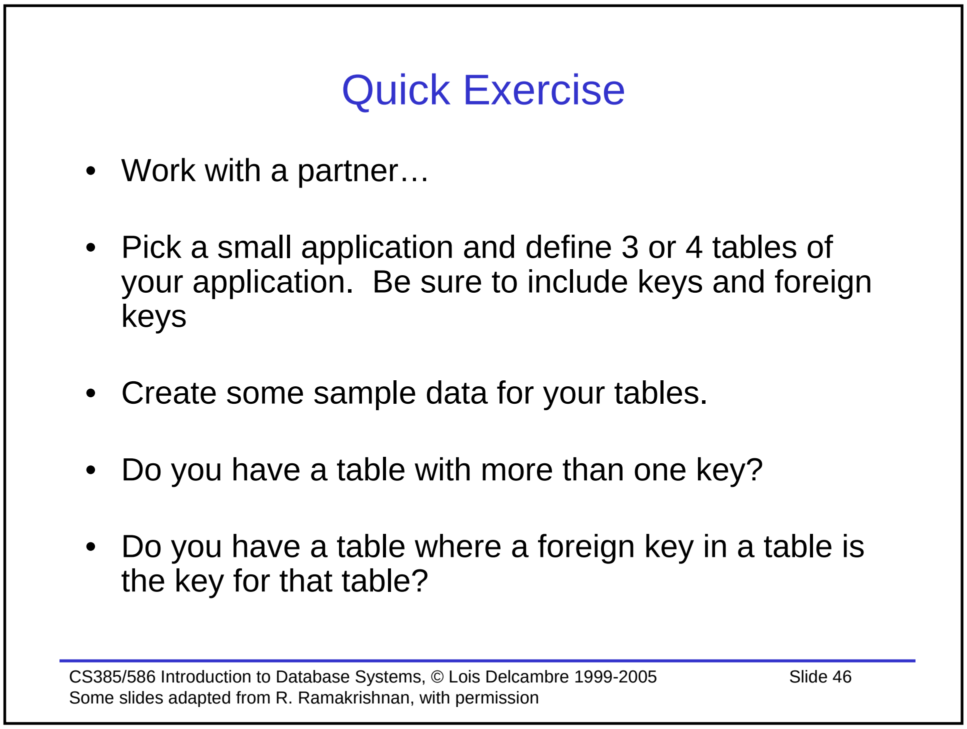
Recipe (id, name, servings, prep-time)

Used-In (recipe-id, ingredient-id, quantity)

What’s the key for this table?

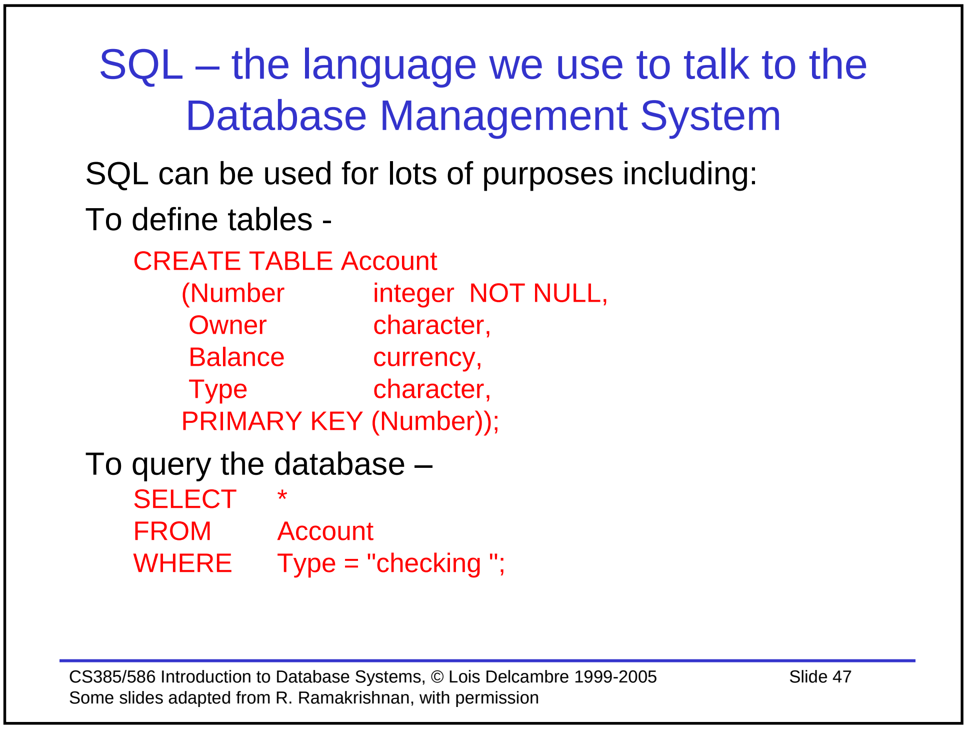
Ingredient (id, name)

In general, we always need to introduce a new table for a many-to-many relationship



Quick Exercise

- Work with a partner…
- Pick a small application and define 3 or 4 tables of your application. Be sure to include keys and foreign keys
- Create some sample data for your tables.
- Do you have a table with more than one key?
- Do you have a table where a foreign key in a table is the key for that table?



SQL – the language we use to talk to the Database Management System

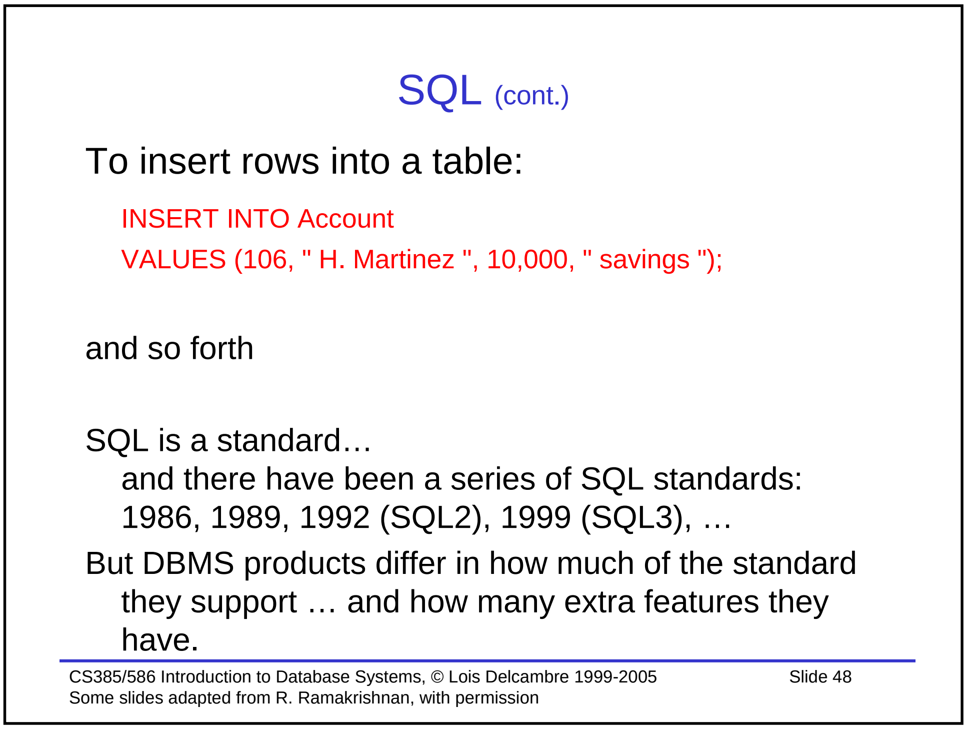
SQL can be used for lots of purposes including:

To define tables -

```
CREATE TABLE Account
    (Number          integer  NOT NULL,
     Owner           character,
     Balance         currency,
     Type            character,
    PRIMARY KEY (Number));
```

To query the database –

```
SELECT     *
FROM       Account
WHERE      Type = "checking ";
```



SQL (cont.)

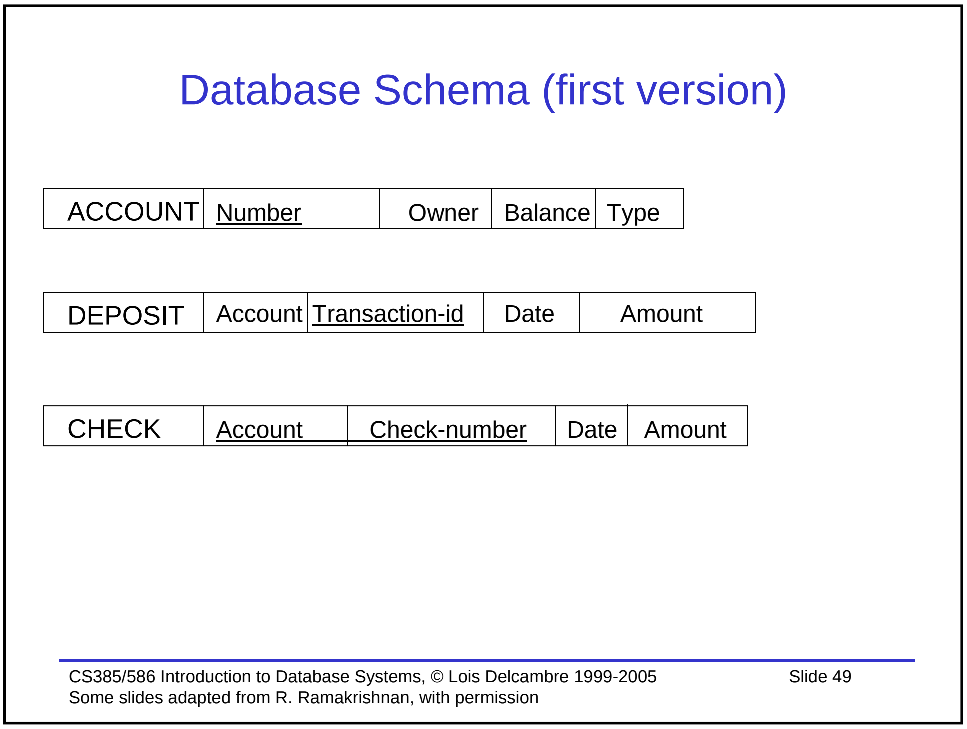
To insert rows into a table:

```
INSERT INTO Account
VALUES (106, " H. Martinez ", 10,000, " savings ");
```

and so forth

SQL is a standard…
and there have been a series of SQL standards: 1986, 1989, 1992 (SQL2), 1999 (SQL3), …

But DBMS products differ in how much of the standard they support … and how many extra features they have.

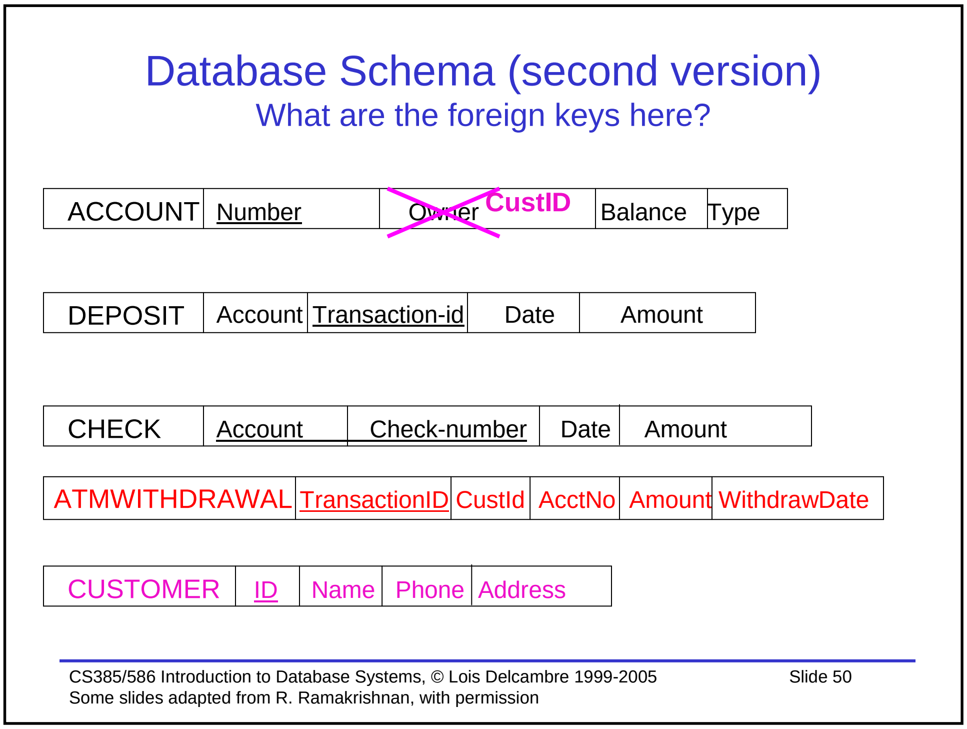


Database Schema (first version)

ACCOUNT	Number	Owner	Balance	Type	

DEPOSIT	Account	Transaction-id	Date	Amount

CHECK	Account	Check-number	Date	Amount



Database Schema (second version)

What are the foreign keys here?

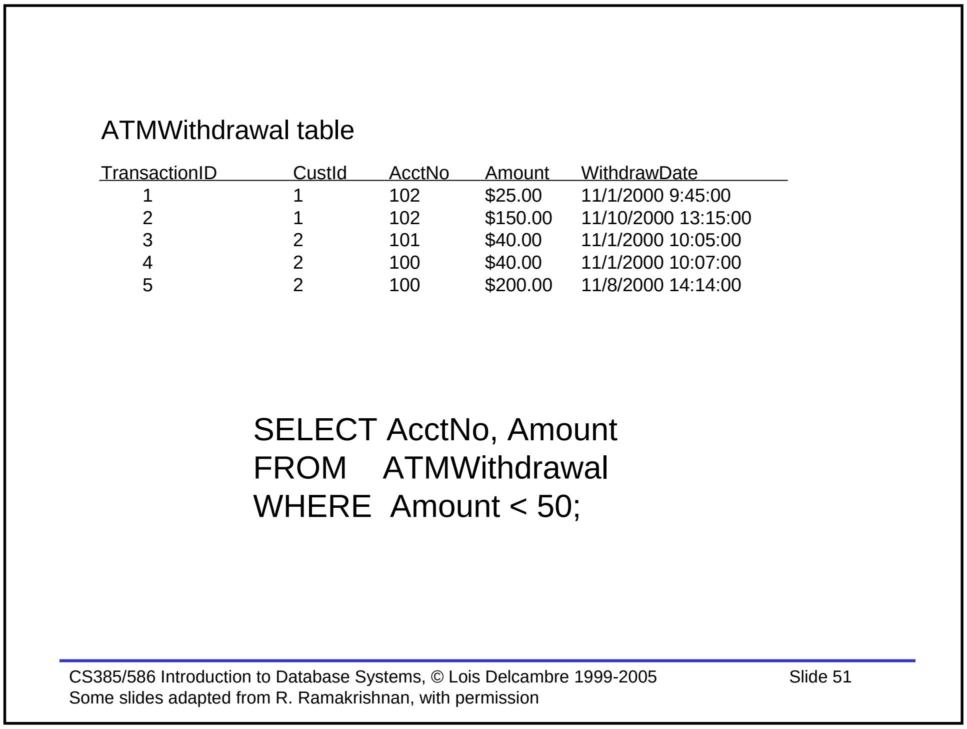
ACCOUNT	Number	~~Owner~~ CustID	Balance	Type

DEPOSIT	Account	Transaction-id	Date	Amount

CHECK	Account	Check-number	Date	Amount

ATMWITHDRAWAL	TransactionID	CustId	AcctNo	Amount	WithdrawDate

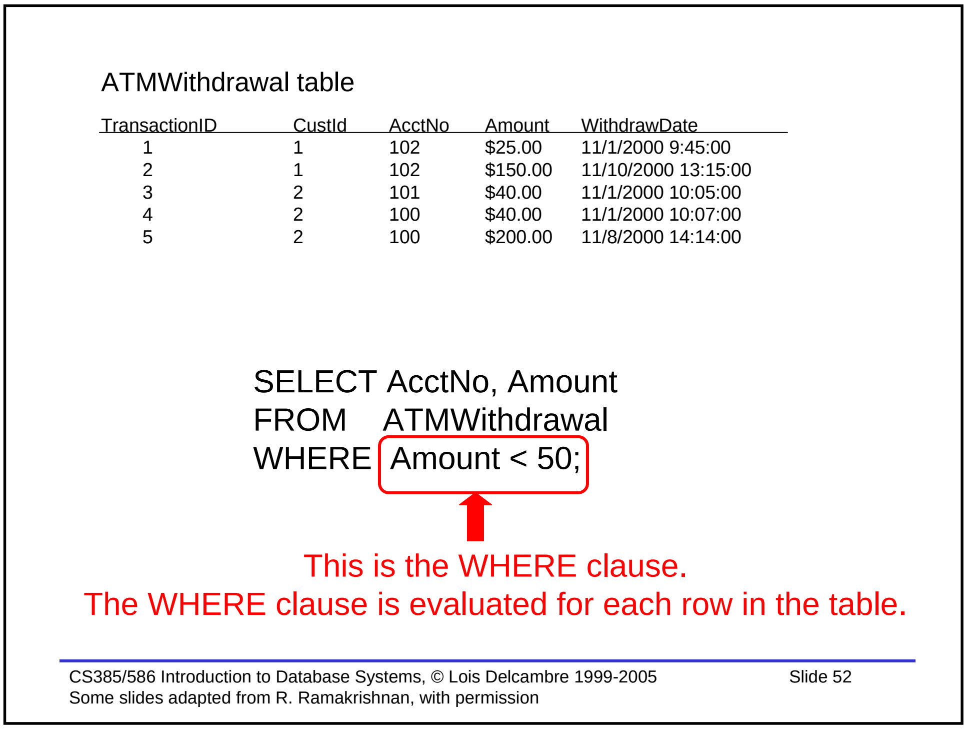
CUSTOMER	ID	Name	Phone	Address



ATMWithdrawal table

TransactionID	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/2000 9:45:00
2	1	102	$150.00	11/10/2000 13:15:00
3	2	101	$40.00	11/1/2000 10:05:00
4	2	100	$40.00	11/1/2000 10:07:00
5	2	100	$200.00	11/8/2000 14:14:00

```
SELECT AcctNo, Amount
FROM    ATMWithdrawal
WHERE  Amount < 50;
```

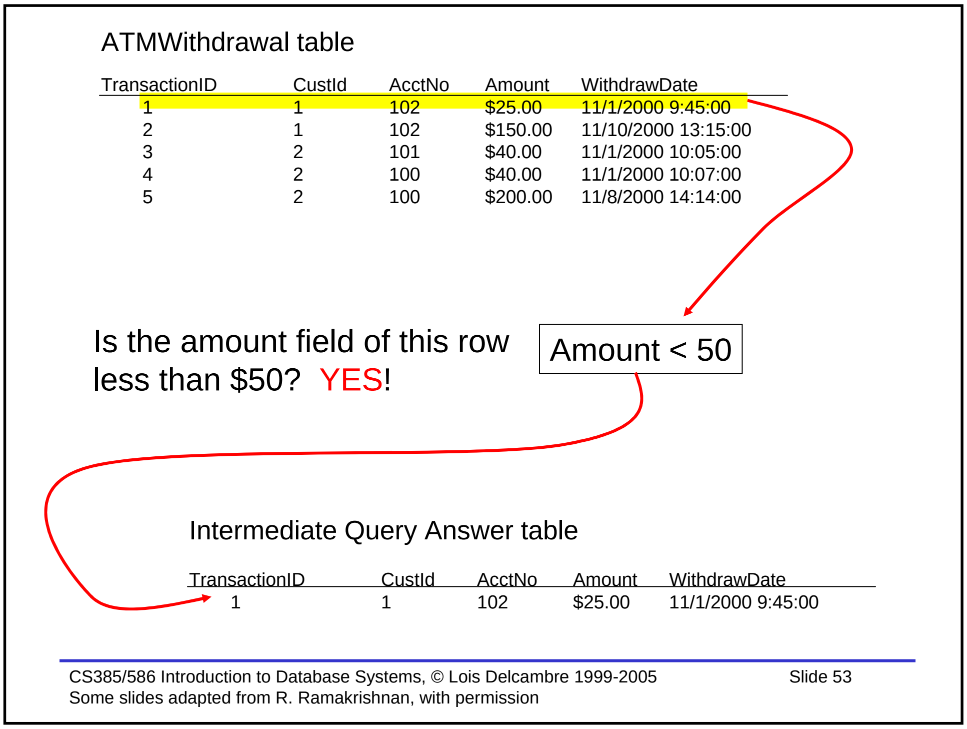


ATMWithdrawal table

TransactionID	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/2000 9:45:00
2	1	102	$150.00	11/10/2000 13:15:00
3	2	101	$40.00	11/1/2000 10:05:00
4	2	100	$40.00	11/1/2000 10:07:00
5	2	100	$200.00	11/8/2000 14:14:00

```
SELECT AcctNo, Amount
FROM    ATMWithdrawal
WHERE  Amount < 50;
```

This is the WHERE clause.
The WHERE clause is evaluated for each row in the table.



ATMWithdrawal table

TransactionID	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/2000 9:45:00
2	1	102	$150.00	11/10/2000 13:15:00
3	2	101	$40.00	11/1/2000 10:05:00
4	2	100	$40.00	11/1/2000 10:07:00
5	2	100	$200.00	11/8/2000 14:14:00

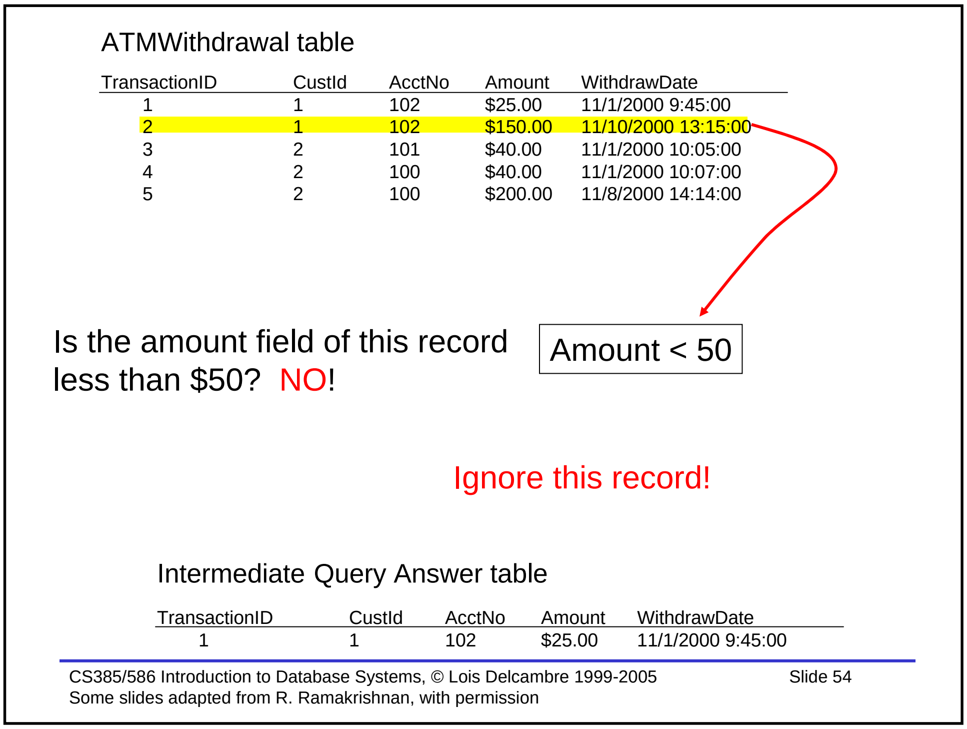
Is the amount field of this row less than $50? YES!

Amount < 50

Intermediate Query Answer table

TransactionID	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/2000 9:45:00

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 53
Some slides adapted from R. Ramakrishnan, with permission



ATMWithdrawal table

TransactionID	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/2000 9:45:00
2	1	102	$150.00	11/10/2000 13:15:00
3	2	101	$40.00	11/1/2000 10:05:00
4	2	100	$40.00	11/1/2000 10:07:00
5	2	100	$200.00	11/8/2000 14:14:00

Is the amount field of this record less than $50? NO!

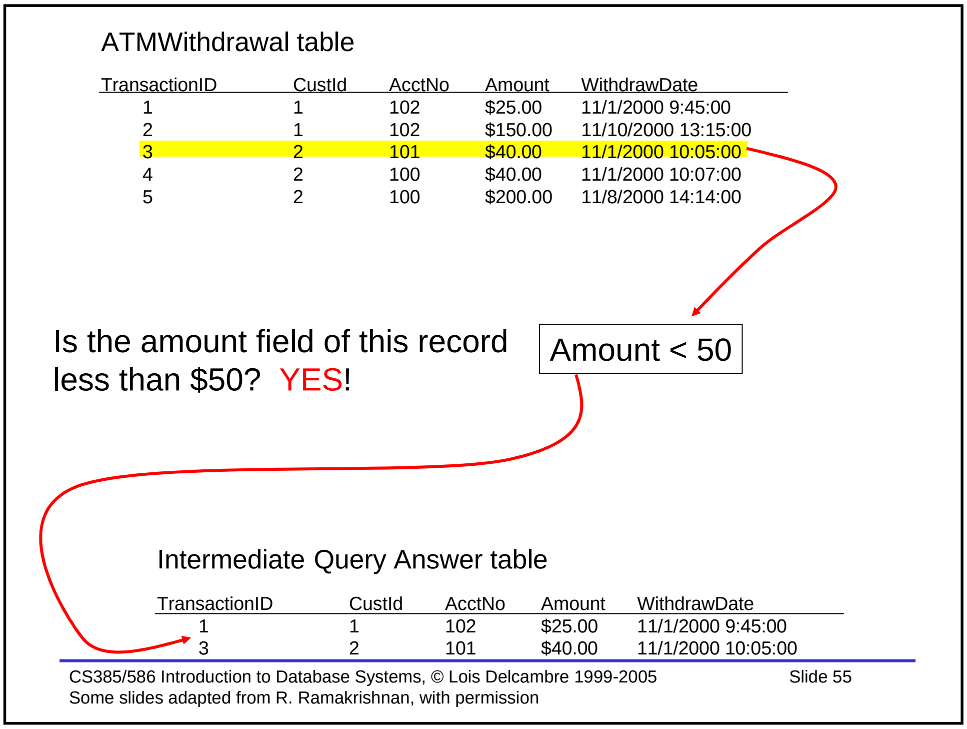
Amount < 50

Ignore this record!

Intermediate Query Answer table

TransactionID	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/2000 9:45:00

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 54
Some slides adapted from R. Ramakrishnan, with permission



ATMWithdrawal table

TransactionID	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/2000 9:45:00
2	1	102	$150.00	11/10/2000 13:15:00
3	2	101	$40.00	11/1/2000 10:05:00
4	2	100	$40.00	11/1/2000 10:07:00
5	2	100	$200.00	11/8/2000 14:14:00

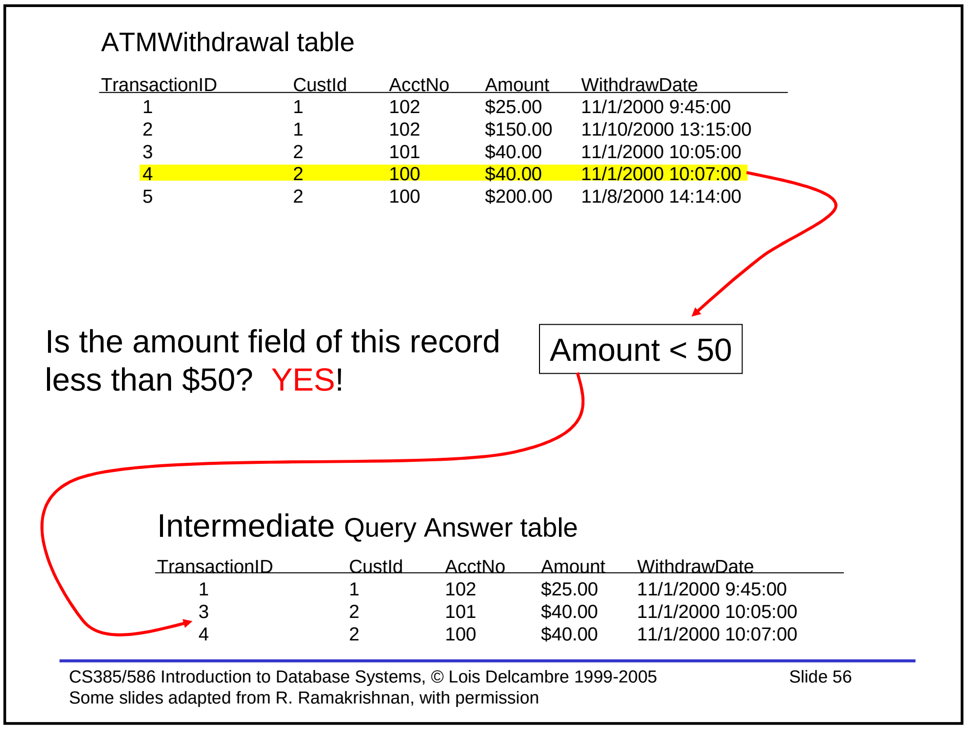
Amount < 50

Is the amount field of this record less than $50? YES!

Intermediate Query Answer table

TransactionID	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/2000 9:45:00
3	2	101	$40.00	11/1/2000 10:05:00

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 55
Some slides adapted from R. Ramakrishnan, with permission



ATMWithdrawal table

TransactionID	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/2000 9:45:00
2	1	102	$150.00	11/10/2000 13:15:00
3	2	101	$40.00	11/1/2000 10:05:00
4	2	100	$40.00	11/1/2000 10:07:00
5	2	100	$200.00	11/8/2000 14:14:00

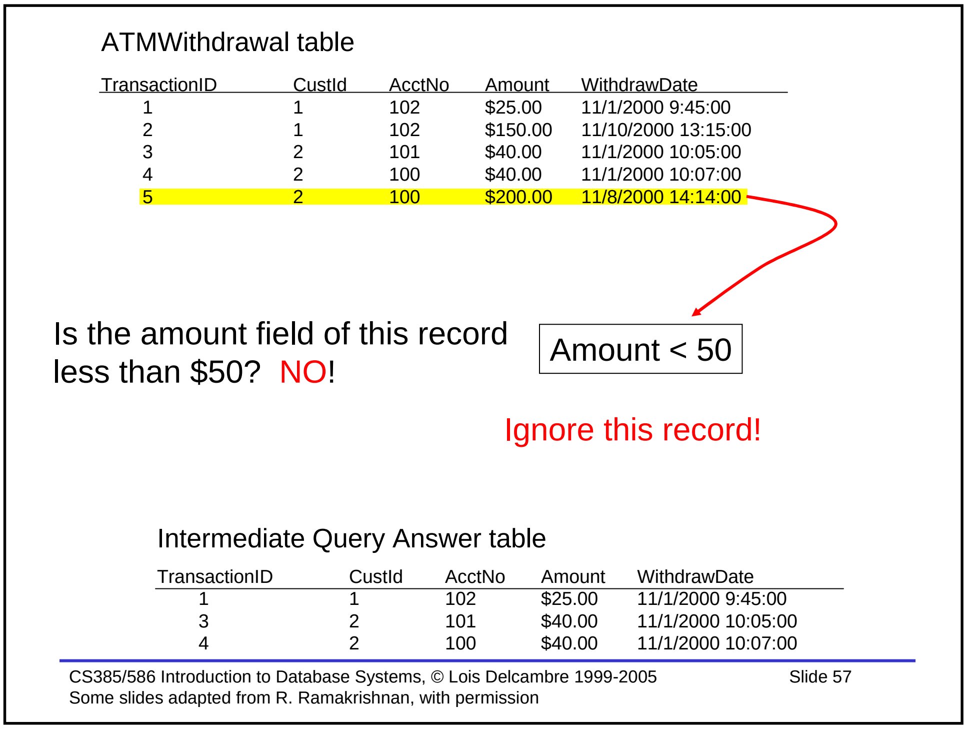
Amount < 50

Is the amount field of this record less than $50? YES!

Intermediate Query Answer table

TransactionID	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/2000 9:45:00
3	2	101	$40.00	11/1/2000 10:05:00
4	2	100	$40.00	11/1/2000 10:07:00

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 56
Some slides adapted from R. Ramakrishnan, with permission



ATMWithdrawal table

TransactionID	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/2000 9:45:00
2	1	102	$150.00	11/10/2000 13:15:00
3	2	101	$40.00	11/1/2000 10:05:00
4	2	100	$40.00	11/1/2000 10:07:00
5	2	100	$200.00	11/8/2000 14:14:00

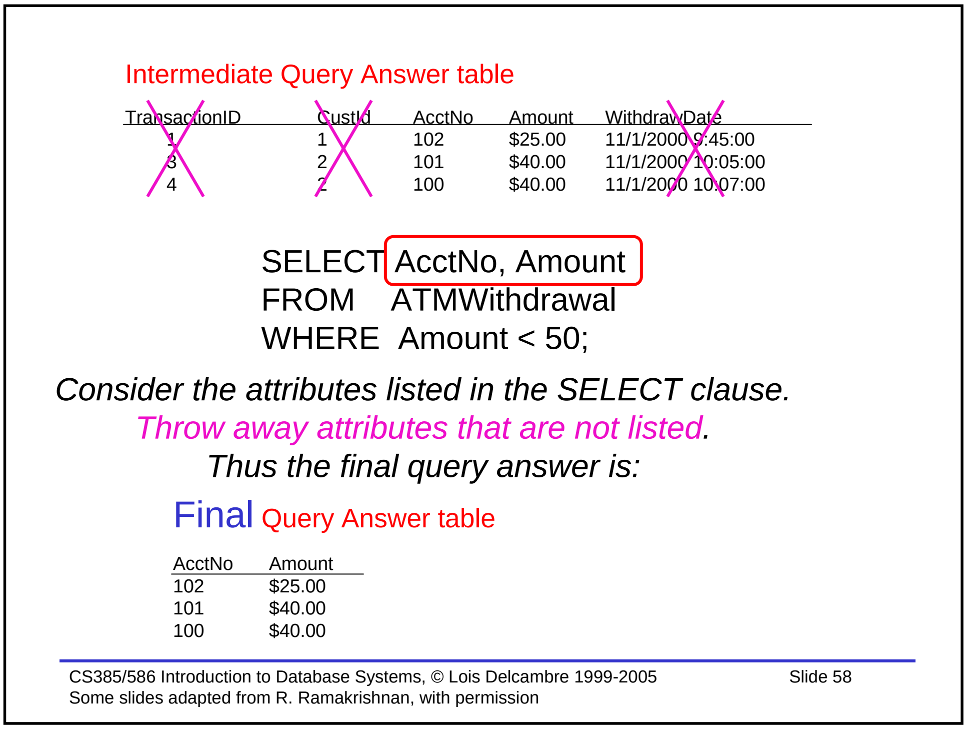
Is the amount field of this record less than $50? NO!

Amount < 50

Ignore this record!

Intermediate Query Answer table

TransactionID	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/2000 9:45:00
3	2	101	$40.00	11/1/2000 10:05:00
4	2	100	$40.00	11/1/2000 10:07:00



Intermediate Query Answer table

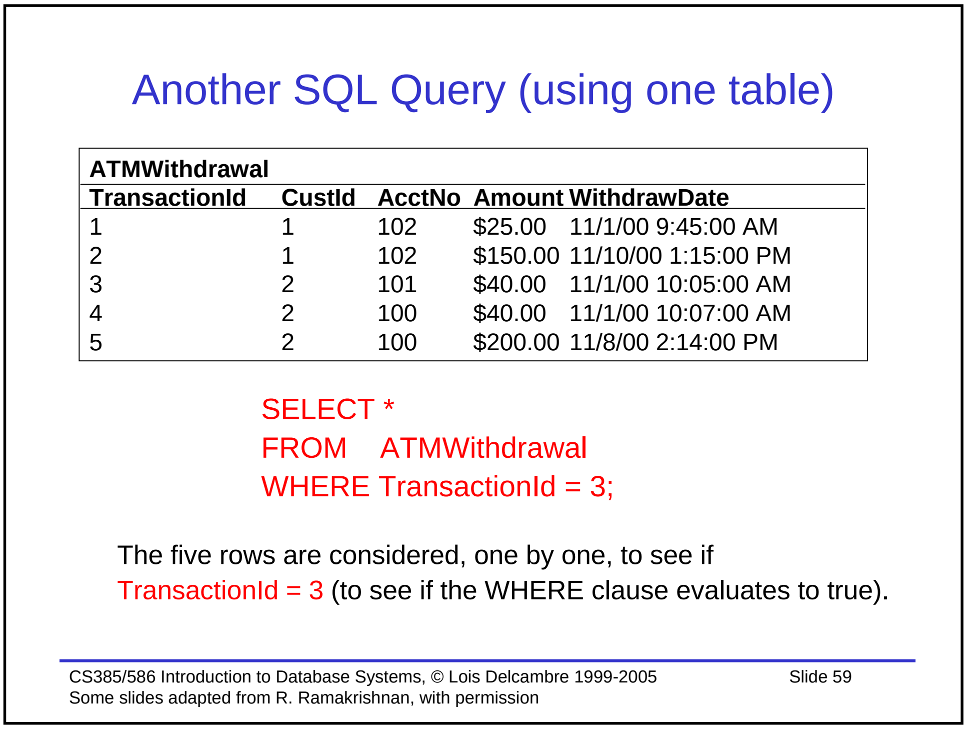
~~TransactionID~~	~~CustId~~	AcctNo	Amount	~~WithdrawDate~~
~~1~~	~~1~~	102	$25.00	~~11/1/2000 9:45:00~~
~~3~~	~~2~~	101	$40.00	~~11/1/2000 10:05:00~~
~~4~~	~~2~~	100	$40.00	~~11/1/2000 10:07:00~~

```
SELECT AcctNo, Amount
FROM   ATMWithdrawal
WHERE  Amount < 50;
```

Consider the attributes listed in the SELECT clause.
Throw away attributes that are not listed.
Thus the final query answer is:

Final Query Answer table

AcctNo	Amount
102	$25.00
101	$40.00
100	$40.00



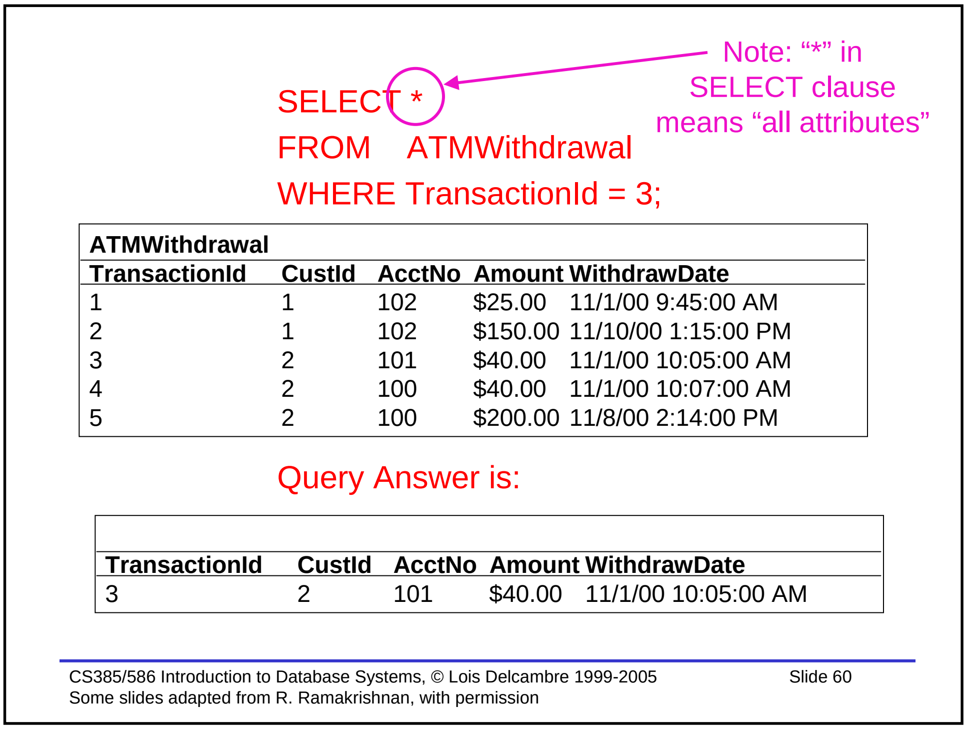
Another SQL Query (using one table)

ATMWithdrawal

TransactionId	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/00 9:45:00 AM
2	1	102	$150.00	11/10/00 1:15:00 PM
3	2	101	$40.00	11/1/00 10:05:00 AM
4	2	100	$40.00	11/1/00 10:07:00 AM
5	2	100	$200.00	11/8/00 2:14:00 PM

```
SELECT *
FROM    ATMWithdrawal
WHERE TransactionId = 3;
```

The five rows are considered, one by one, to see if TransactionId = 3 (to see if the WHERE clause evaluates to true).



```
SELECT *
FROM    ATMWithdrawal
WHERE TransactionId = 3;
```

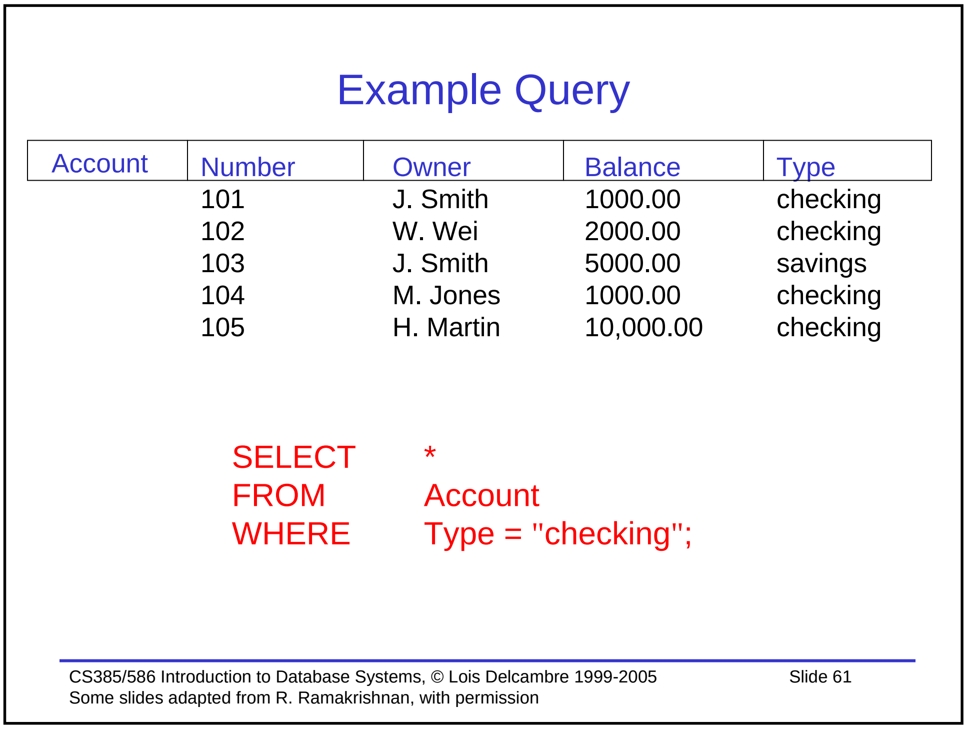
Note: "*" in SELECT clause means "all attributes"

ATMWithdrawal

TransactionId	CustId	AcctNo	Amount	WithdrawDate
1	1	102	$25.00	11/1/00 9:45:00 AM
2	1	102	$150.00	11/10/00 1:15:00 PM
3	2	101	$40.00	11/1/00 10:05:00 AM
4	2	100	$40.00	11/1/00 10:07:00 AM
5	2	100	$200.00	11/8/00 2:14:00 PM

Query Answer is:

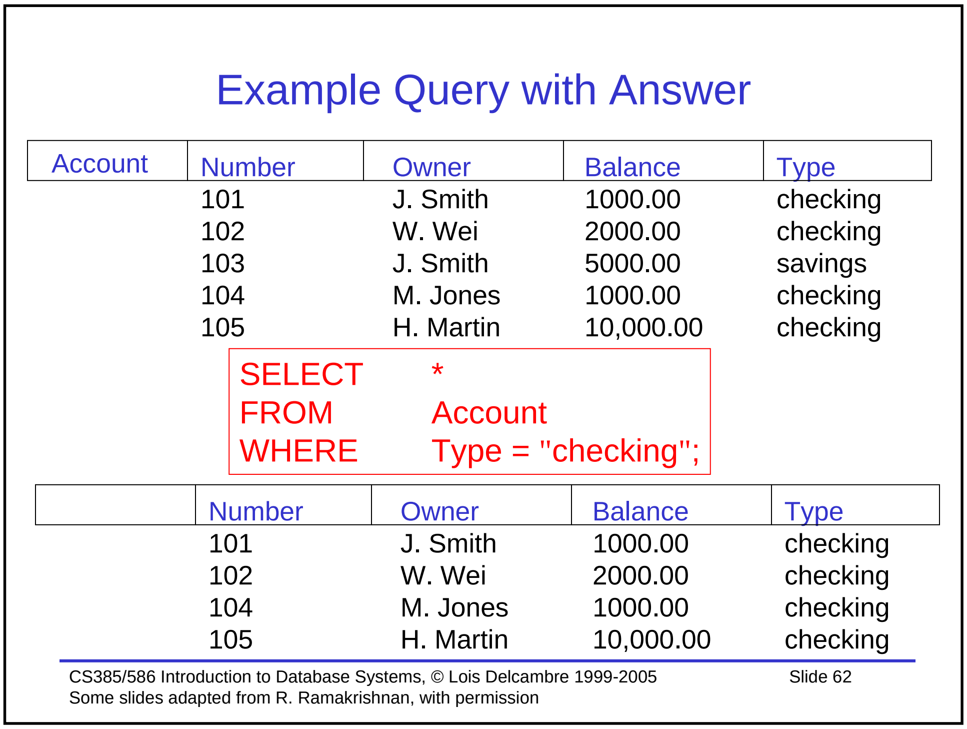
TransactionId	CustId	AcctNo	Amount	WithdrawDate
3	2	101	$40.00	11/1/00 10:05:00 AM



Example Query

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

```
SELECT      *
FROM        Account
WHERE       Type = "checking";
```

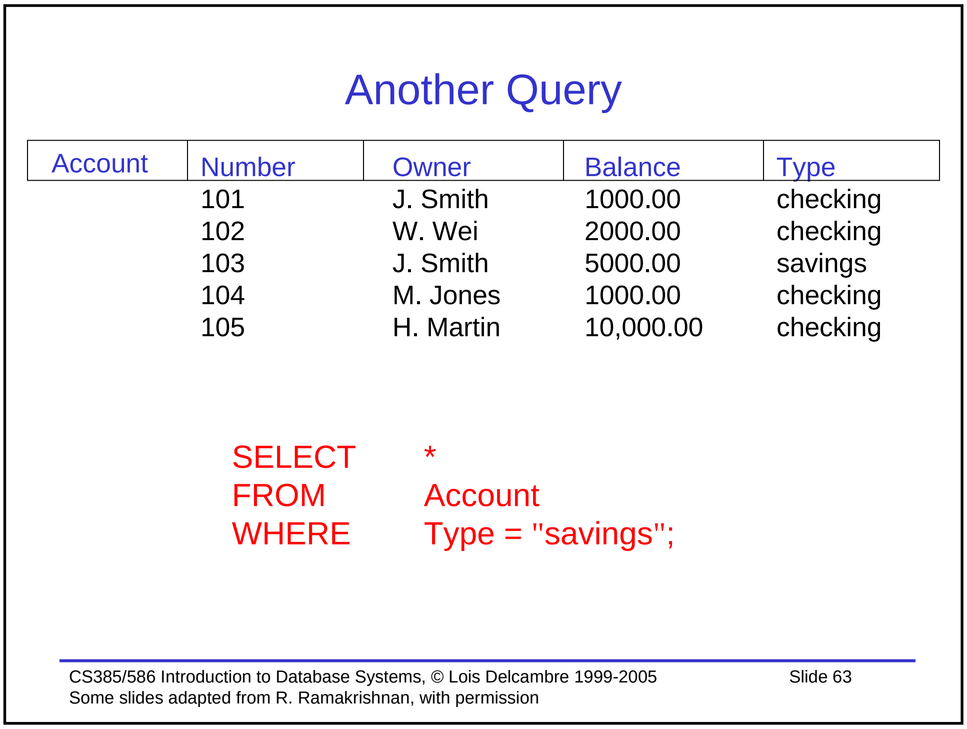


Example Query with Answer

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

```
SELECT      *
FROM        Account
WHERE       Type = "checking";
```

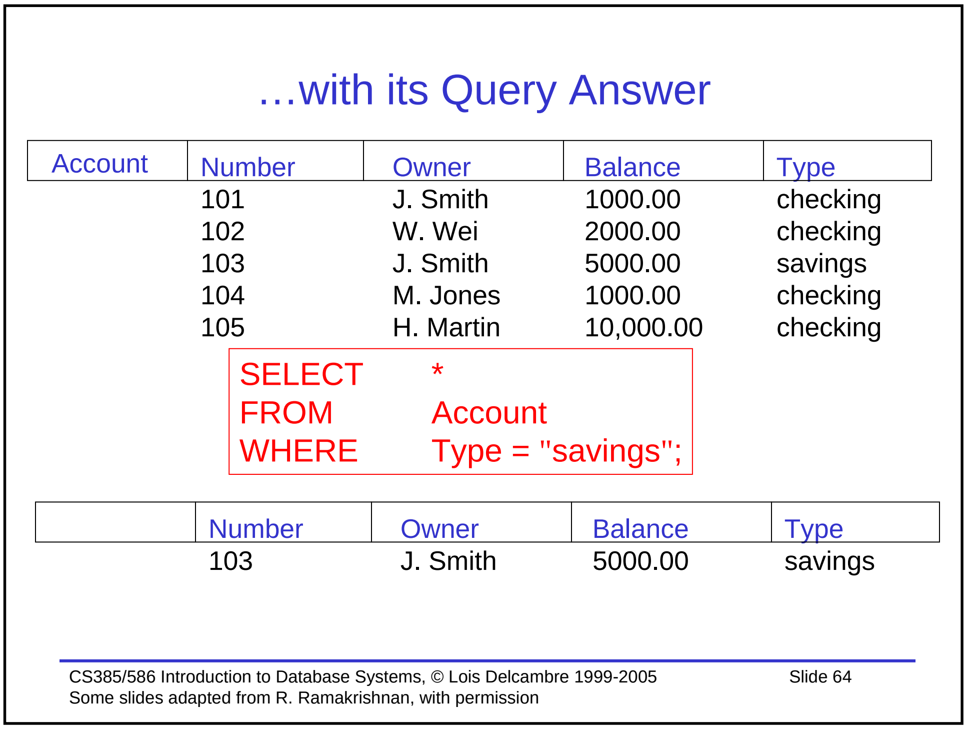
	Number	Owner	Balance	Type
	101	J. Smith	1000.00	checking
	102	W. Wei	2000.00	checking
	104	M. Jones	1000.00	checking
	105	H. Martin	10,000.00	checking



Another Query

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

```
SELECT      *
FROM        Account
WHERE       Type = "savings";
```

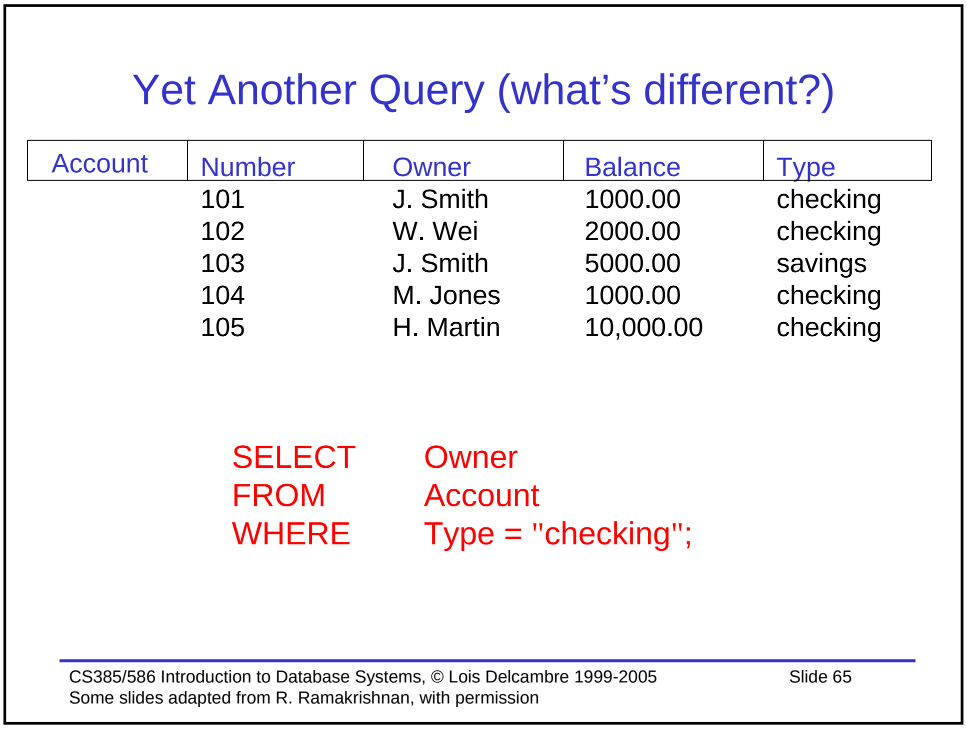


…with its Query Answer

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

```
SELECT      *
FROM        Account
WHERE       Type = "savings";
```

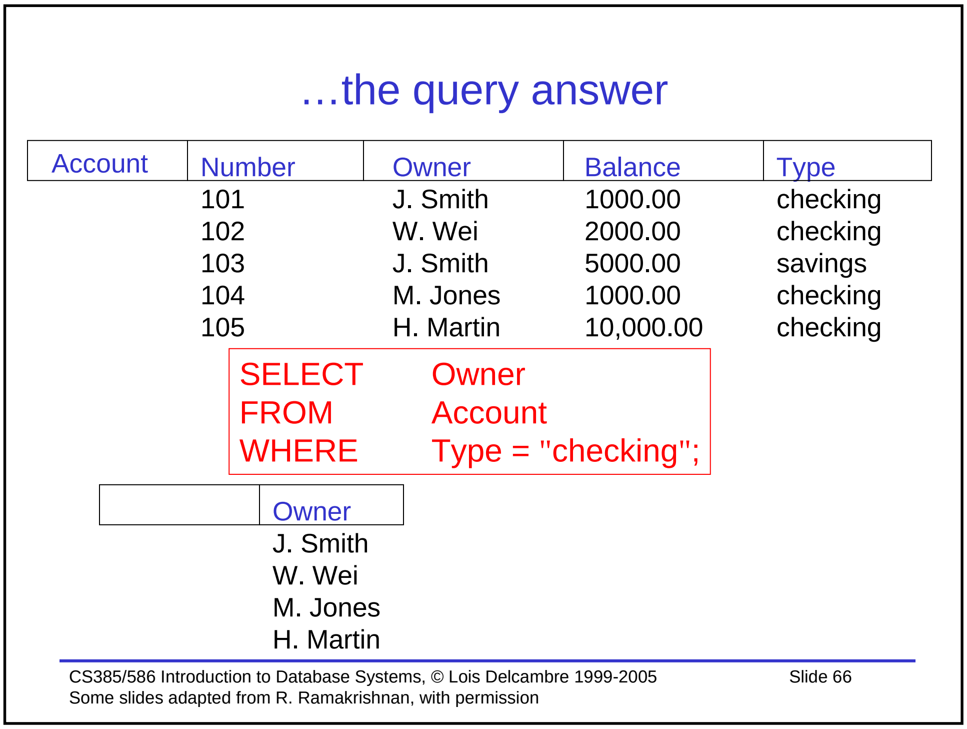
	Number	Owner	Balance	Type
	103	J. Smith	5000.00	savings



Yet Another Query (what’s different?)

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

```
SELECT      Owner
FROM        Account
WHERE       Type = "checking";
```

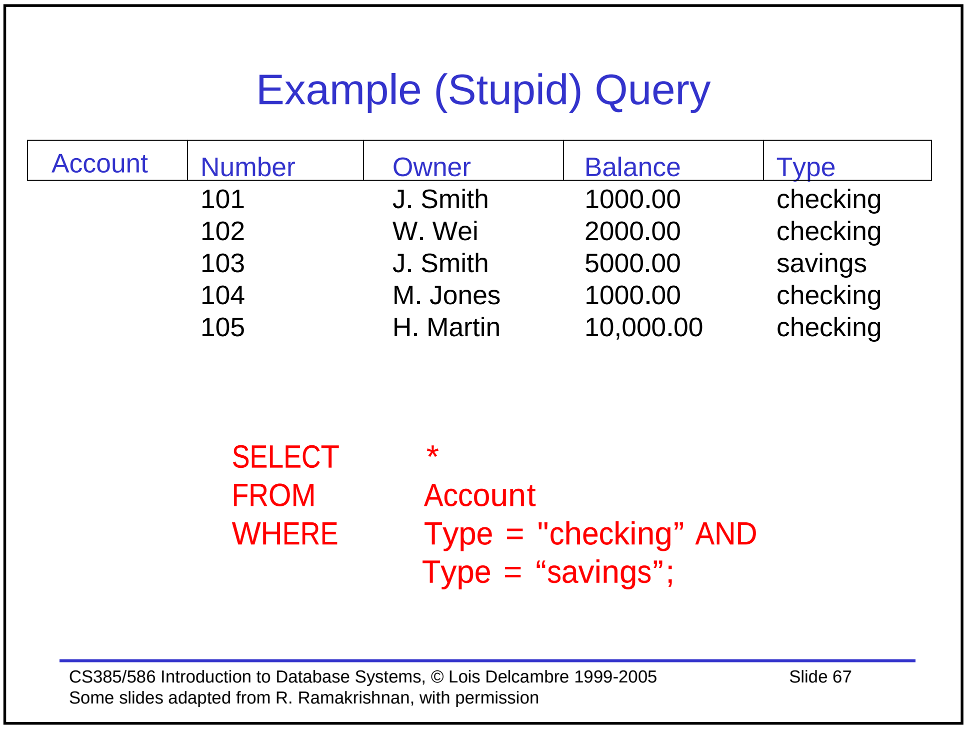


…the query answer

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

```
SELECT      Owner
FROM        Account
WHERE       Type = "checking";
```

	Owner
	J. Smith
	W. Wei
	M. Jones
	H. Martin

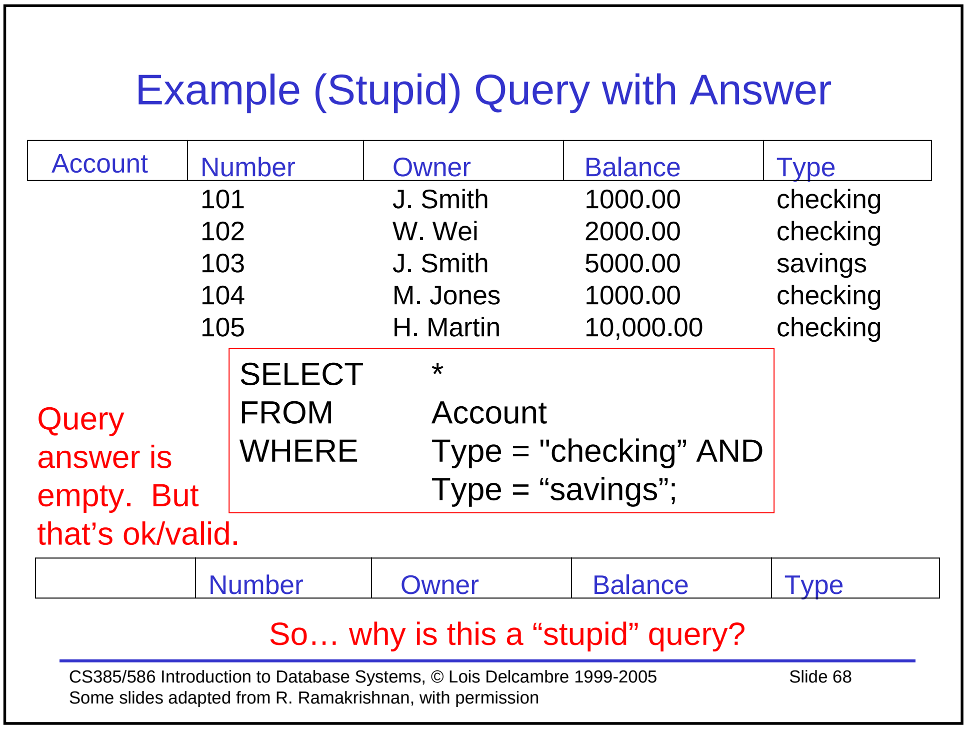


Example (Stupid) Query

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

```
SELECT      *
FROM        Account
WHERE       Type  =  "checking” AND
            Type  =  “savings”;
```

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 67
Some slides adapted from R. Ramakrishnan, with permission



Example (Stupid) Query with Answer

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

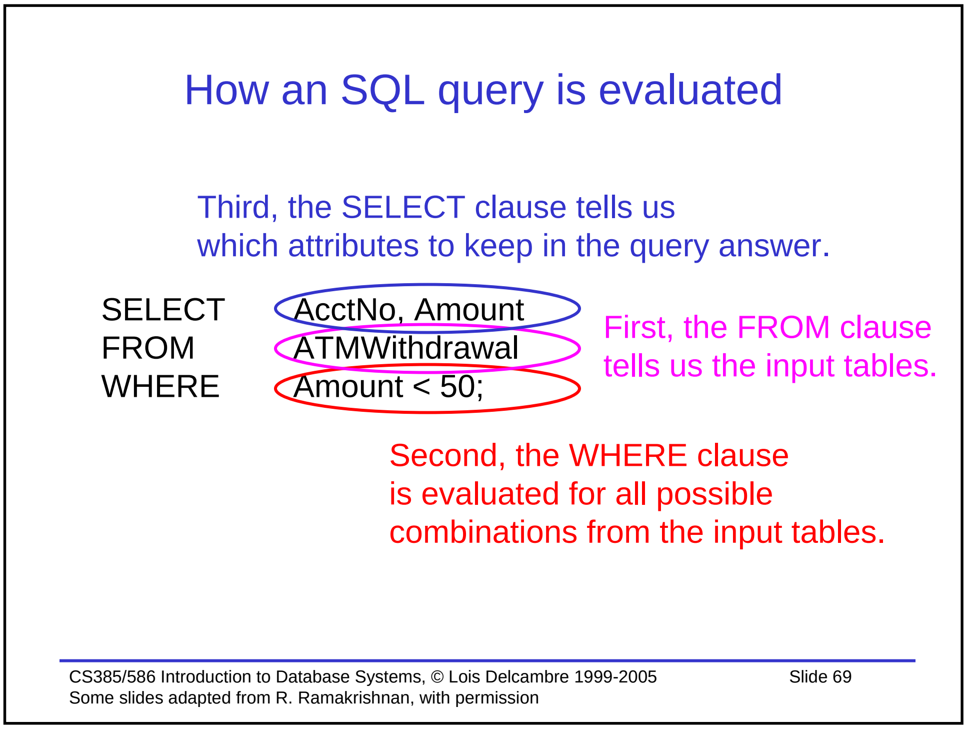
```
SELECT      *
FROM        Account
WHERE       Type = "checking” AND
            Type = “savings”;
```

Query answer is empty. But that’s ok/valid.

	Number	Owner	Balance	Type

So… why is this a “stupid” query?

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 68
Some slides adapted from R. Ramakrishnan, with permission



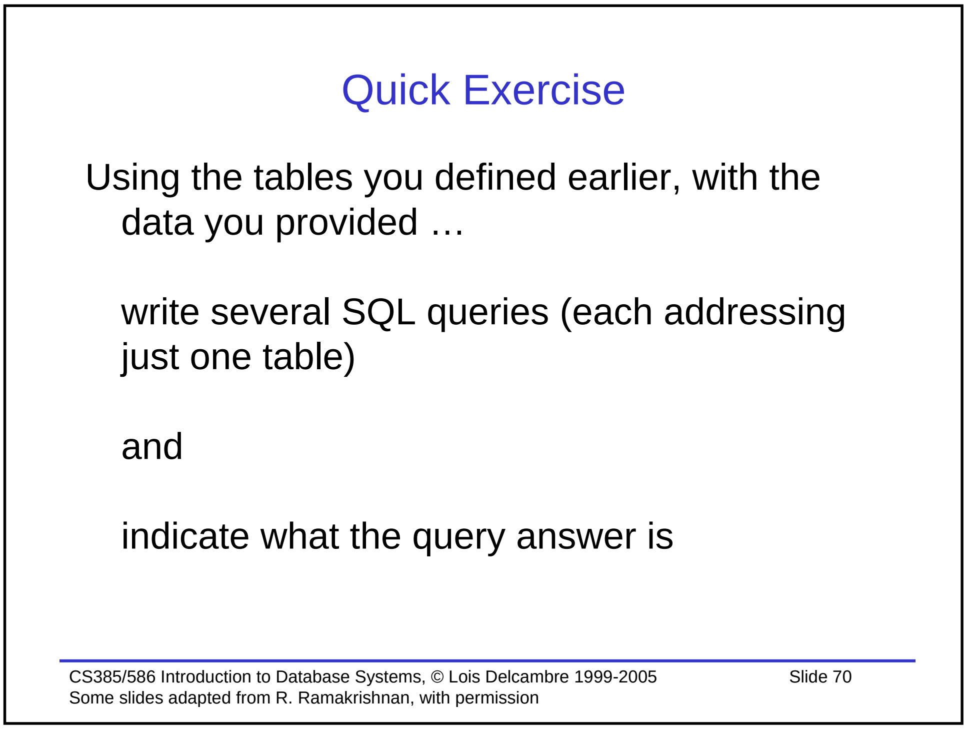
How an SQL query is evaluated

Third, the SELECT clause tells us which attributes to keep in the query answer.

```
SELECT    AcctNo, Amount
FROM      ATMWithdrawal
WHERE     Amount < 50;
```

First, the FROM clause tells us the input tables.

Second, the WHERE clause is evaluated for all possible combinations from the input tables.



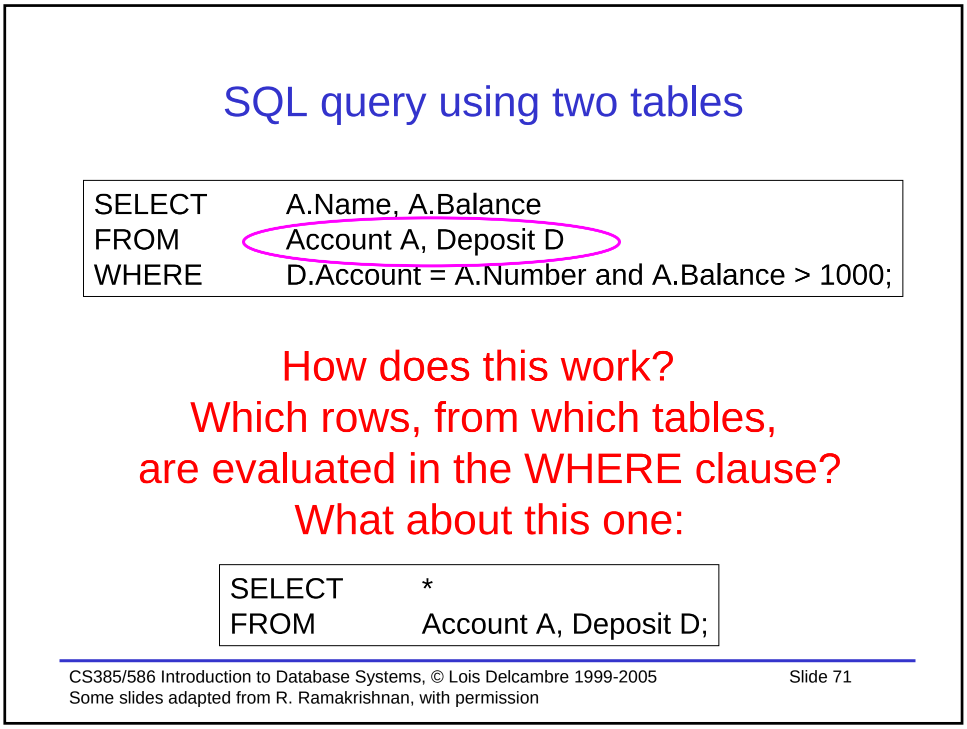
Quick Exercise

Using the tables you defined earlier, with the data you provided …

write several SQL queries (each addressing just one table)

and

indicate what the query answer is

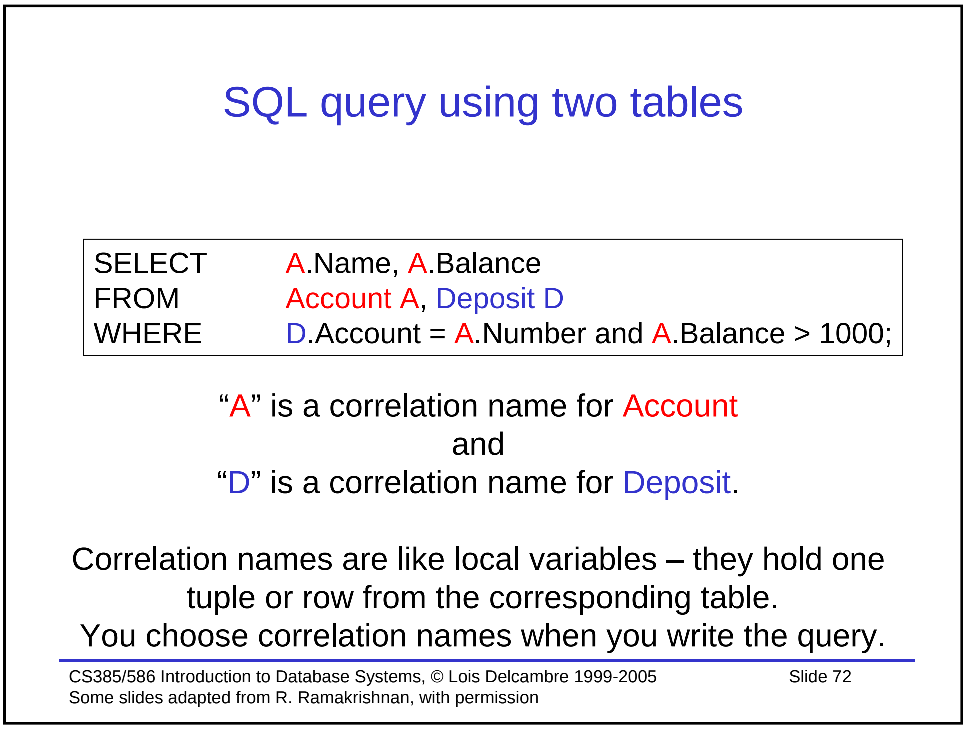


SQL query using two tables

```
SELECT       A.Name, A.Balance
FROM         Account A, Deposit D
WHERE        D.Account = A.Number and A.Balance > 1000;
```

How does this work?
Which rows, from which tables,
are evaluated in the WHERE clause?
What about this one:

```
SELECT       *
FROM         Account A, Deposit D;
```

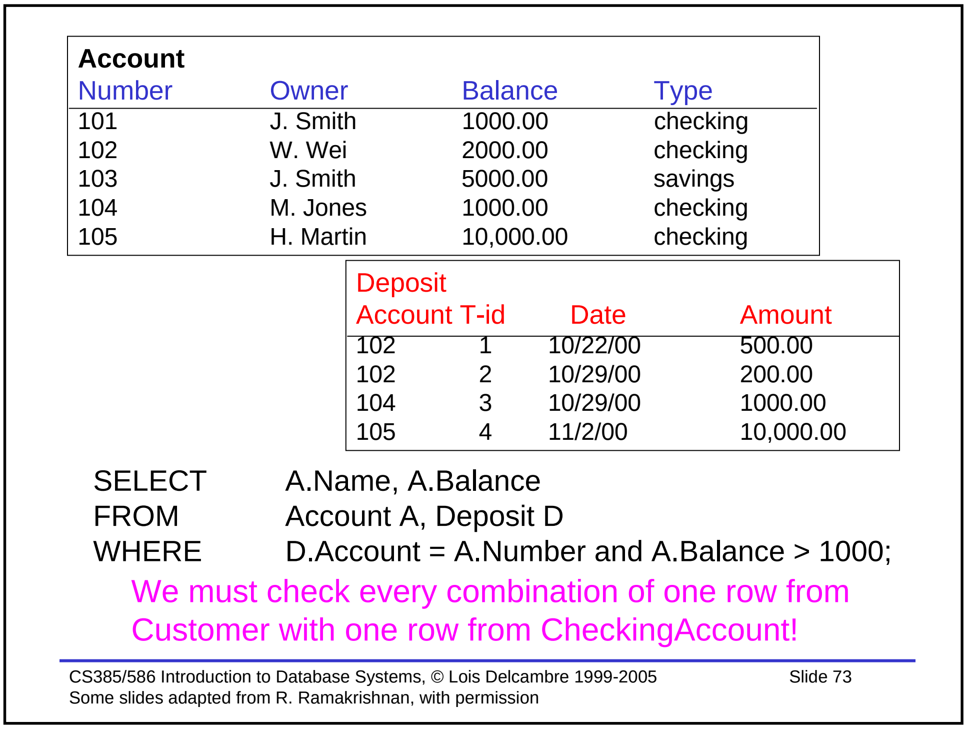


SQL query using two tables

```
SELECT       A.Name, A.Balance
FROM         Account A, Deposit D
WHERE        D.Account = A.Number and A.Balance > 1000;
```

"A" is a correlation name for Account
and
"D" is a correlation name for Deposit.

Correlation names are like local variables – they hold one tuple or row from the corresponding table.
You choose correlation names when you write the query.



Account

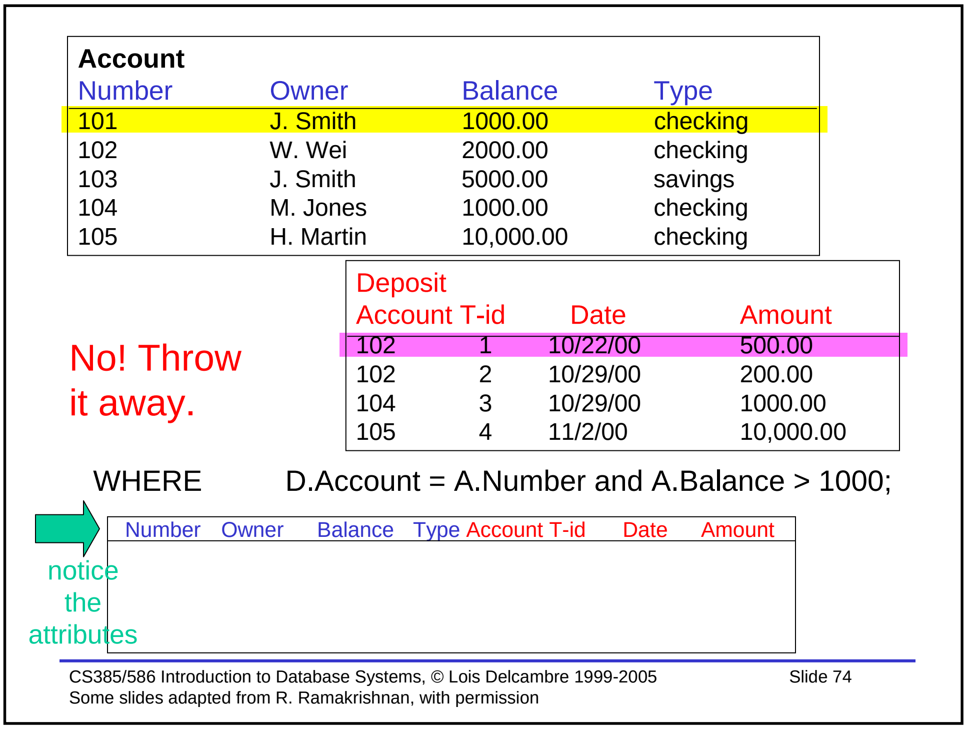
Number	Owner	Balance	Type
101	J. Smith	1000.00	checking
102	W. Wei	2000.00	checking
103	J. Smith	5000.00	savings
104	M. Jones	1000.00	checking
105	H. Martin	10,000.00	checking

Deposit

Account	T-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	11/2/00	10,000.00

```
SELECT      A.Name, A.Balance
FROM        Account A, Deposit D
WHERE       D.Account = A.Number and A.Balance > 1000;
```

We must check every combination of one row from Customer with one row from CheckingAccount!



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

No! Throw it away.

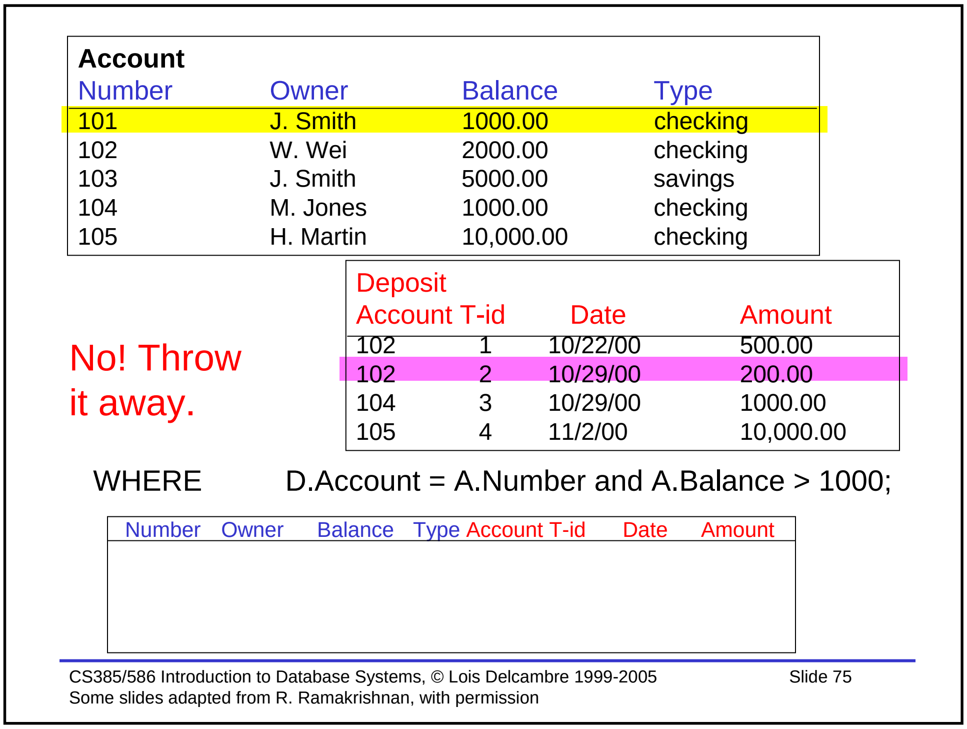
Deposit

Account	T-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	11/2/00	10,000.00

```
WHERE       D.Account = A.Number and A.Balance > 1000;
```

Number	Owner	Balance	Type	Account	T-id	Date	Amount

notice the 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

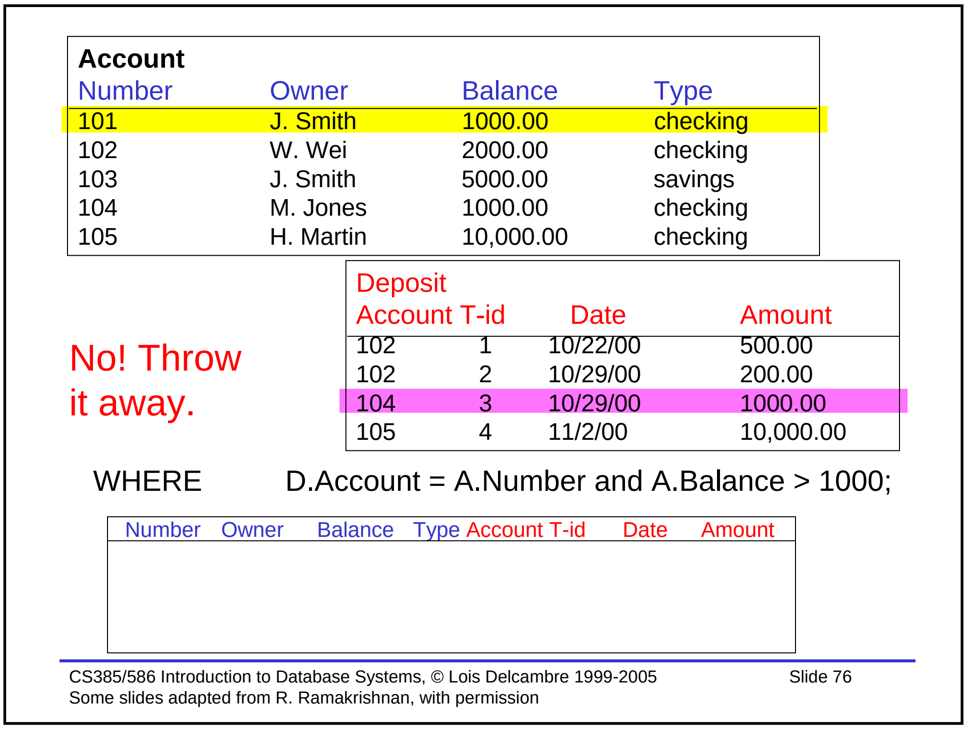
Deposit

Account	T-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	11/2/00	10,000.00

No! Throw it away.

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount



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

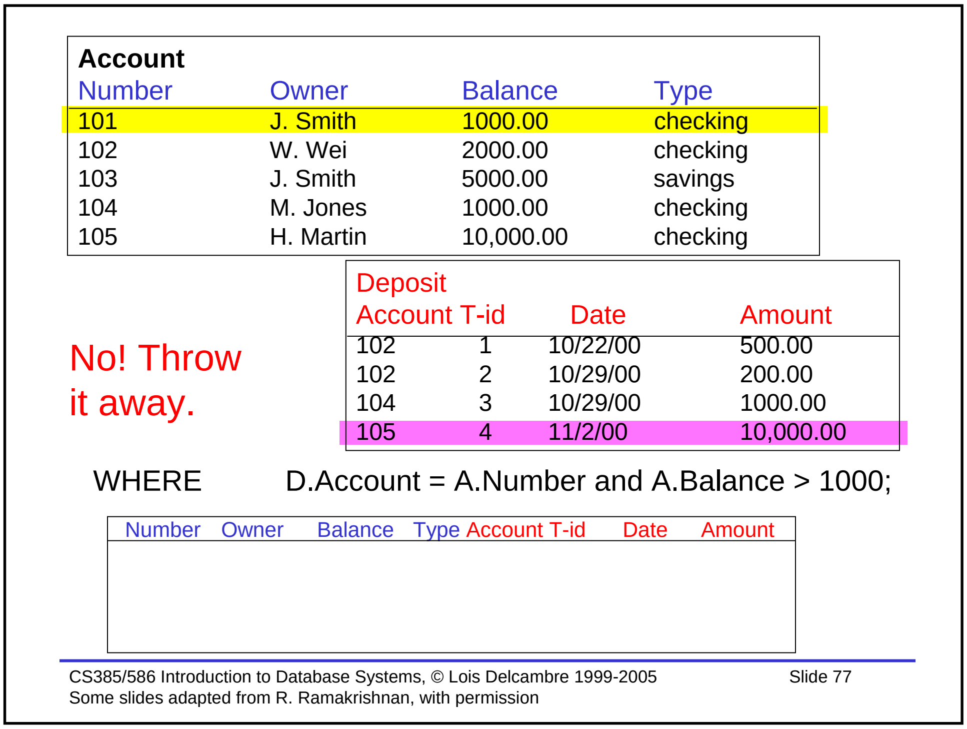
Deposit

Account	T-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	11/2/00	10,000.00

No! Throw it away.

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount



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

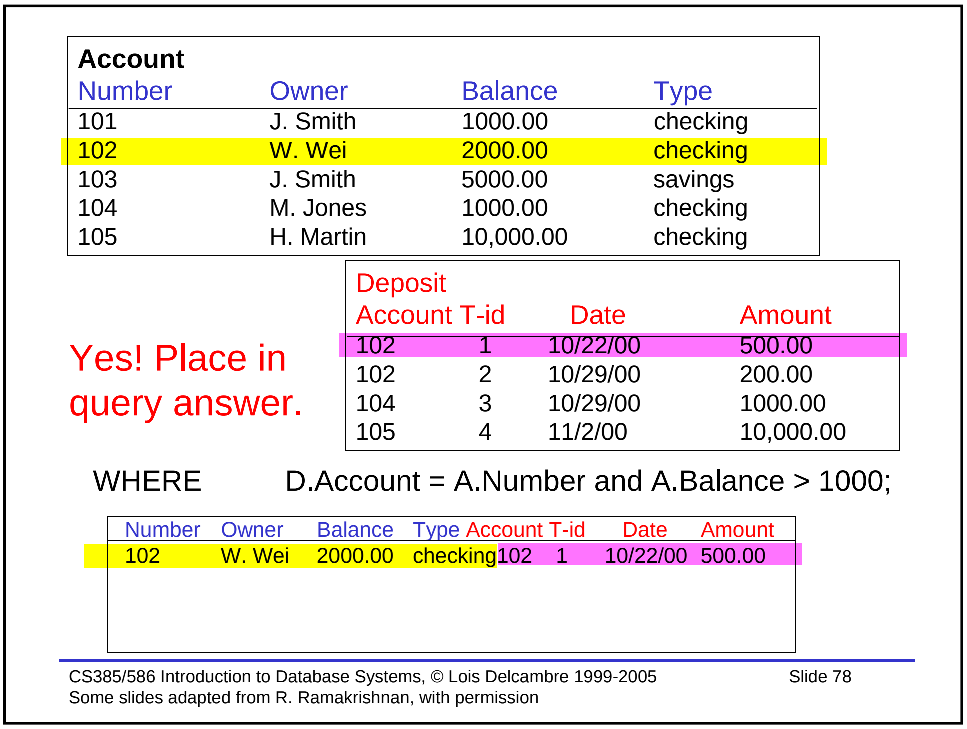
Account	T-id	Date	Amount
102	1	10/22/00	500.00
102	2	10/29/00	200.00
104	3	10/29/00	1000.00
105	4	11/2/00	10,000.00

No! Throw it away.

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 77
Some slides adapted from R. Ramakrishnan, with permission



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

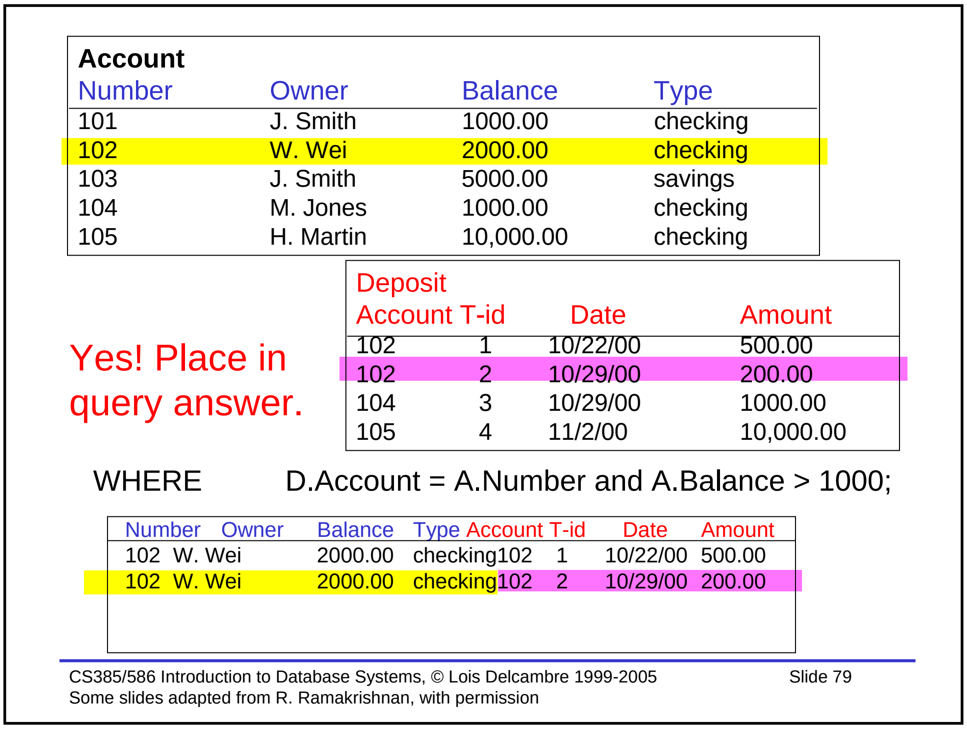
Account	T-id	Date	Amount
102	1	10/22/00	500.00
102	2	10/29/00	200.00
104	3	10/29/00	1000.00
105	4	11/2/00	10,000.00

Yes! Place in query answer.

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount
102	W. Wei	2000.00	checking	102	1	10/22/00	500.00

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 78
Some slides adapted from R. Ramakrishnan, with permission



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

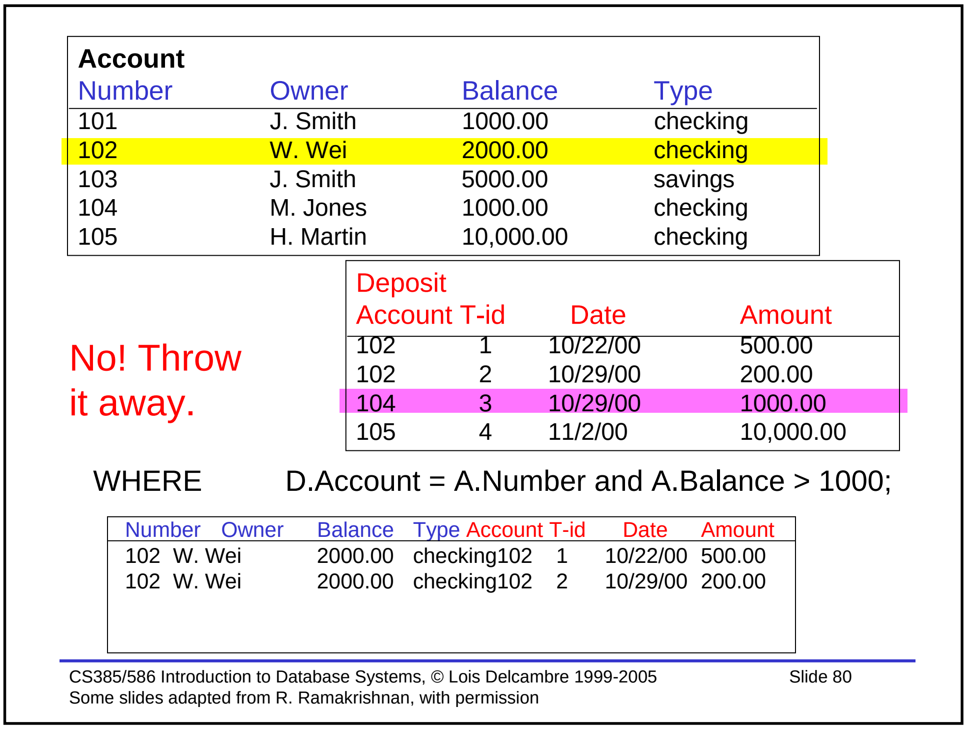
Account	T-id	Date	Amount
102	1	10/22/00	500.00
102	2	10/29/00	200.00
104	3	10/29/00	1000.00
105	4	11/2/00	10,000.00

Yes! Place in query answer.

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount
102	W. Wei	2000.00	checking	102	1	10/22/00	500.00
102	W. Wei	2000.00	checking	102	2	10/29/00	200.00

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 79
Some slides adapted from R. Ramakrishnan, with permission



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

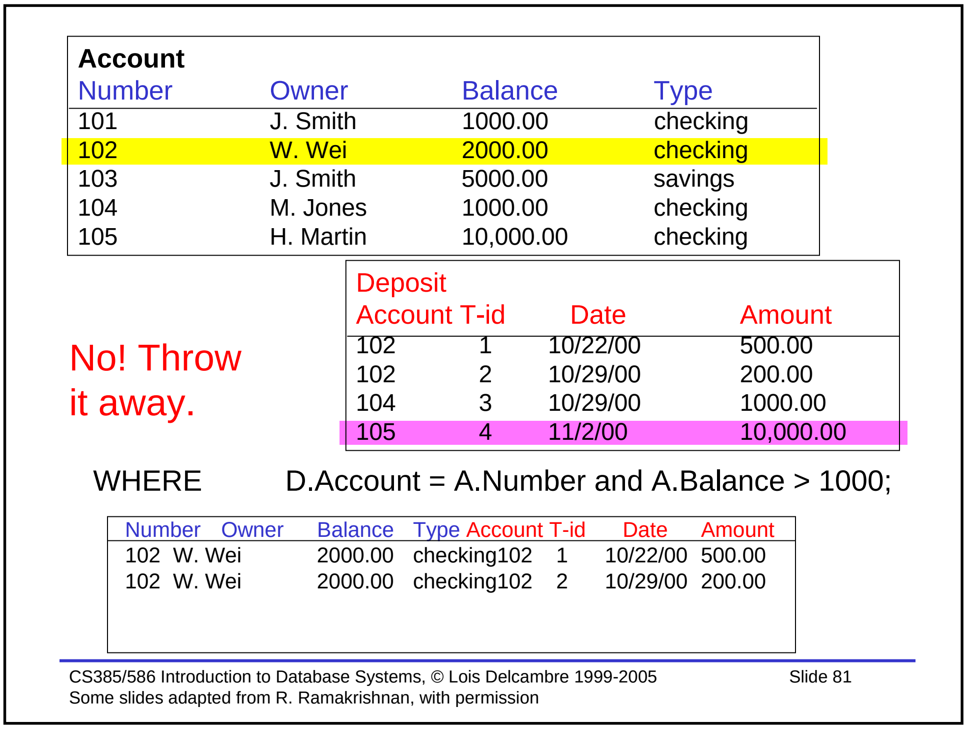
Account	T-id	Date	Amount
102	1	10/22/00	500.00
102	2	10/29/00	200.00
104	3	10/29/00	1000.00
105	4	11/2/00	10,000.00

No! Throw it away.

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount
102	W. Wei	2000.00	checking	102	1	10/22/00	500.00
102	W. Wei	2000.00	checking	102	2	10/29/00	200.00

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 80
Some slides adapted from R. Ramakrishnan, with permission



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

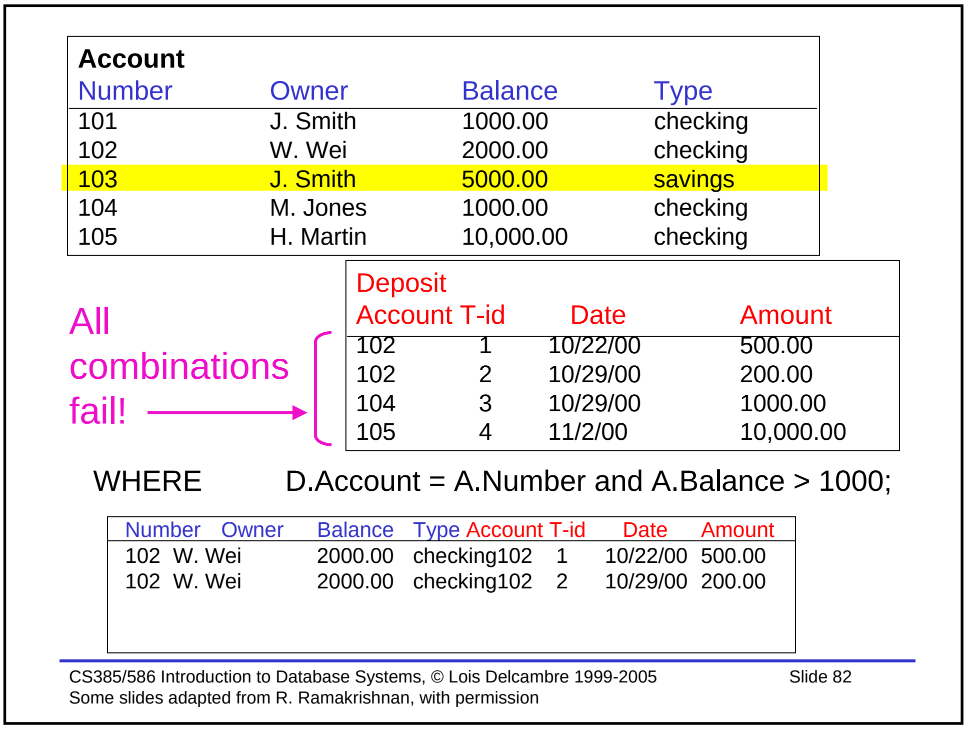
Account	T-id	Date	Amount
102	1	10/22/00	500.00
102	2	10/29/00	200.00
104	3	10/29/00	1000.00
105	4	11/2/00	10,000.00

No! Throw it away.

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount
102	W. Wei	2000.00	checking	102	1	10/22/00	500.00
102	W. Wei	2000.00	checking	102	2	10/29/00	200.00

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 81
Some slides adapted from R. Ramakrishnan, with permission



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

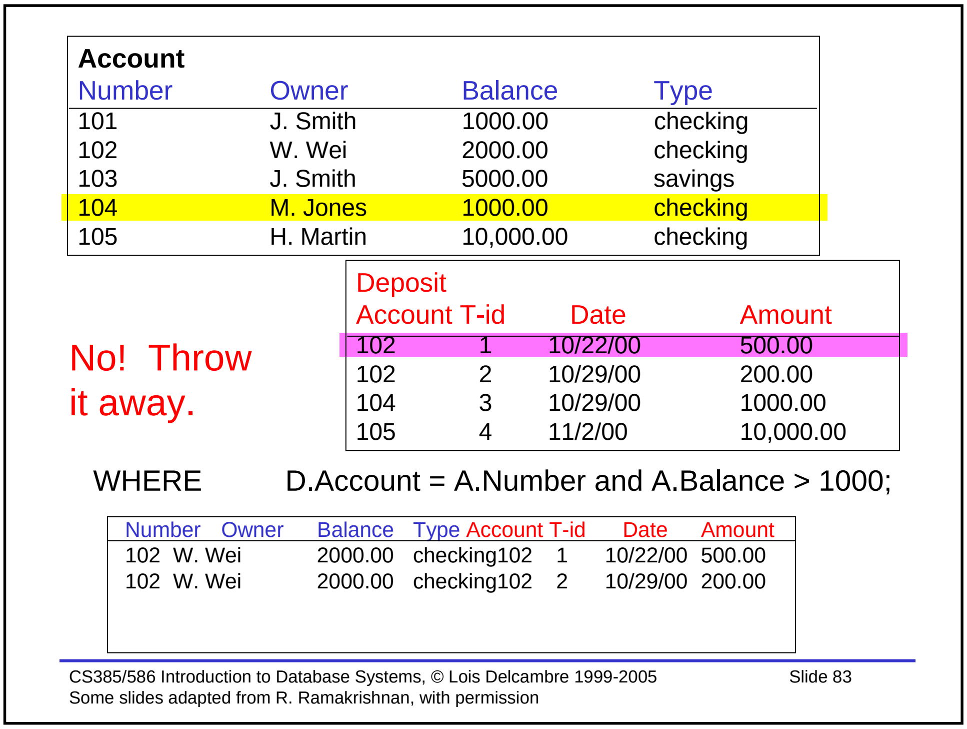
Account	T-id	Date	Amount
102	1	10/22/00	500.00
102	2	10/29/00	200.00
104	3	10/29/00	1000.00
105	4	11/2/00	10,000.00

All combinations fail!

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount
102	W. Wei	2000.00	checking	102	1	10/22/00	500.00
102	W. Wei	2000.00	checking	102	2	10/29/00	200.00

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 82
Some slides adapted from R. Ramakrishnan, with permission



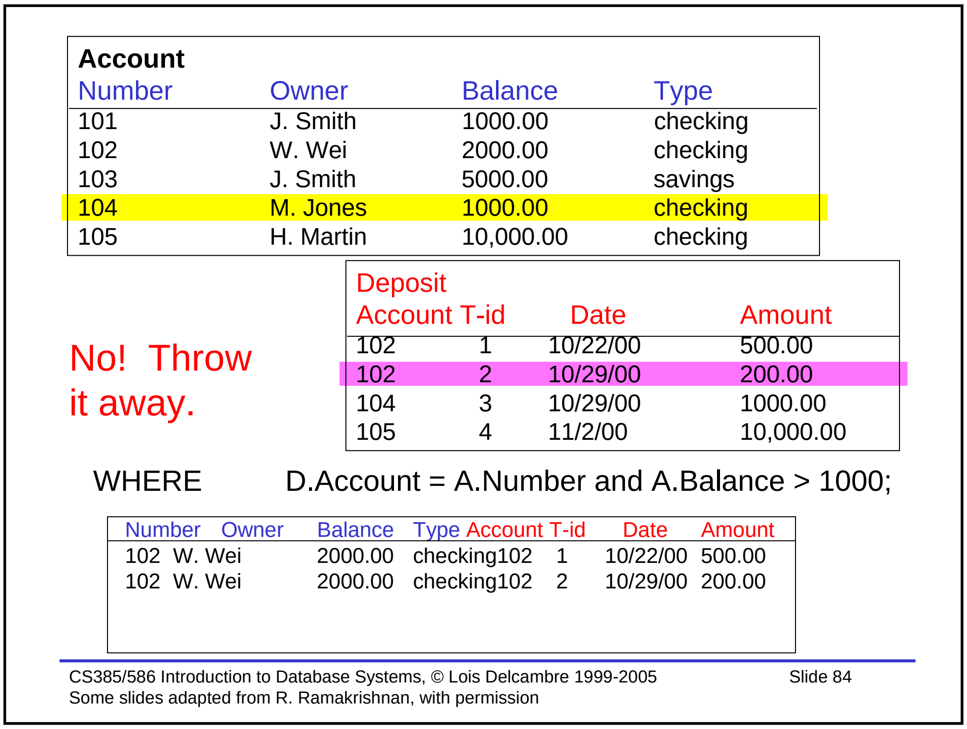
Account			
Number	Owner	Balance	Type
101	J. Smith	1000.00	checking
102	W. Wei	2000.00	checking
103	J. Smith	5000.00	savings
104	M. Jones	1000.00	checking
105	H. Martin	10,000.00	checking

No! Throw it away.

Deposit			
Account	T-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	11/2/00	10,000.00

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount
102	W. Wei	2000.00	checking	102	1	10/22/00	500.00
102	W. Wei	2000.00	checking	102	2	10/29/00	200.00



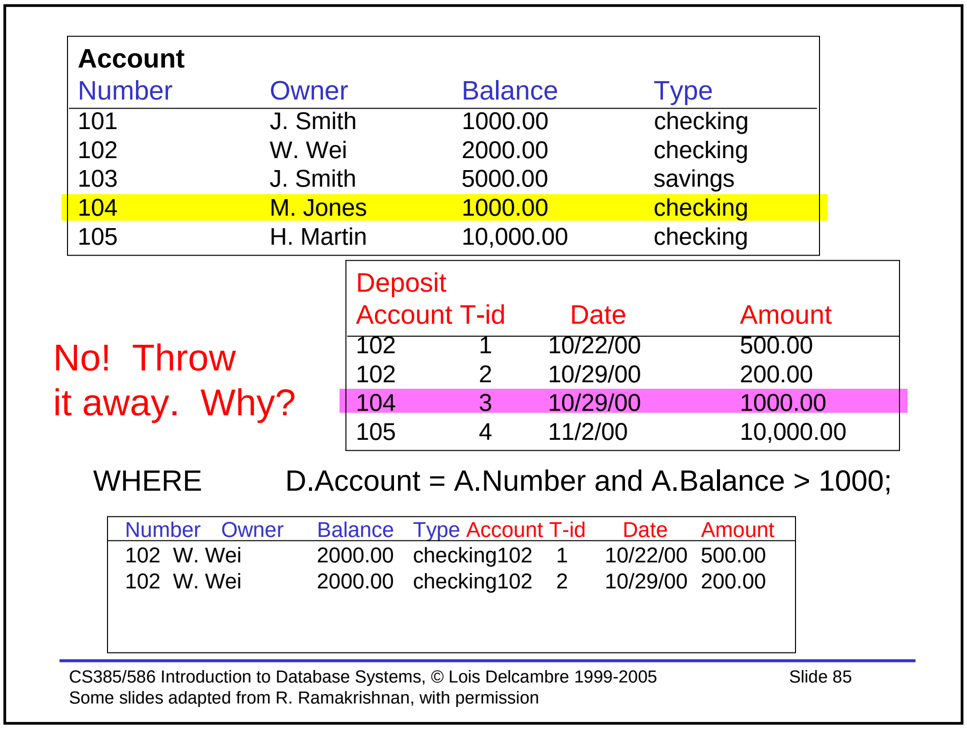
Account			
Number	Owner	Balance	Type
101	J. Smith	1000.00	checking
102	W. Wei	2000.00	checking
103	J. Smith	5000.00	savings
104	M. Jones	1000.00	checking
105	H. Martin	10,000.00	checking

No! Throw it away.

Deposit			
Account	T-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	11/2/00	10,000.00

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount
102	W. Wei	2000.00	checking	102	1	10/22/00	500.00
102	W. Wei	2000.00	checking	102	2	10/29/00	200.00



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

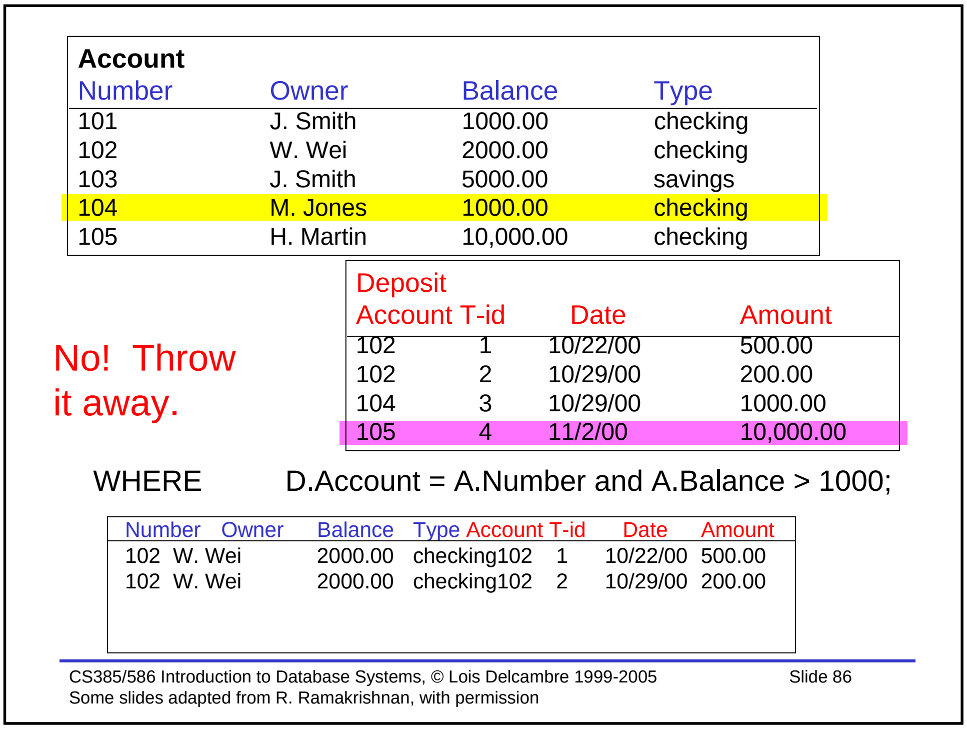
Account	T-id	Date	Amount
102	1	10/22/00	500.00
102	2	10/29/00	200.00
104	3	10/29/00	1000.00
105	4	11/2/00	10,000.00

No! Throw it away. Why?

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount
102	W. Wei	2000.00	checking	102	1	10/22/00	500.00
102	W. Wei	2000.00	checking	102	2	10/29/00	200.00

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 85
Some slides adapted from R. Ramakrishnan, with permission



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

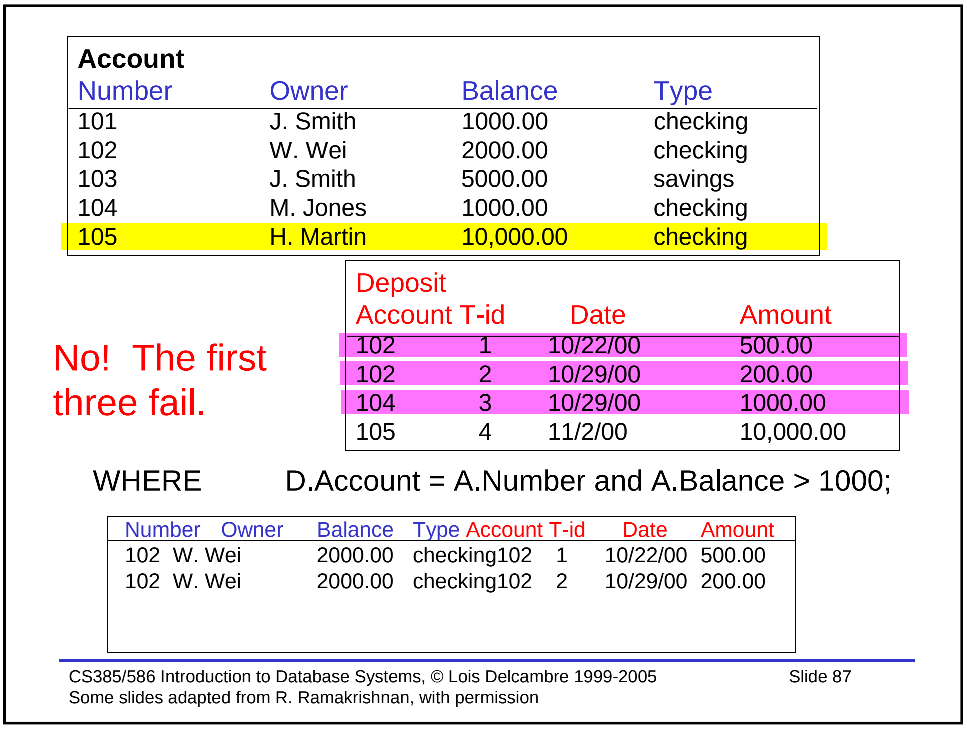
Account	T-id	Date	Amount
102	1	10/22/00	500.00
102	2	10/29/00	200.00
104	3	10/29/00	1000.00
105	4	11/2/00	10,000.00

No! Throw it away.

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount
102	W. Wei	2000.00	checking	102	1	10/22/00	500.00
102	W. Wei	2000.00	checking	102	2	10/29/00	200.00

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 86
Some slides adapted from R. Ramakrishnan, with permission



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

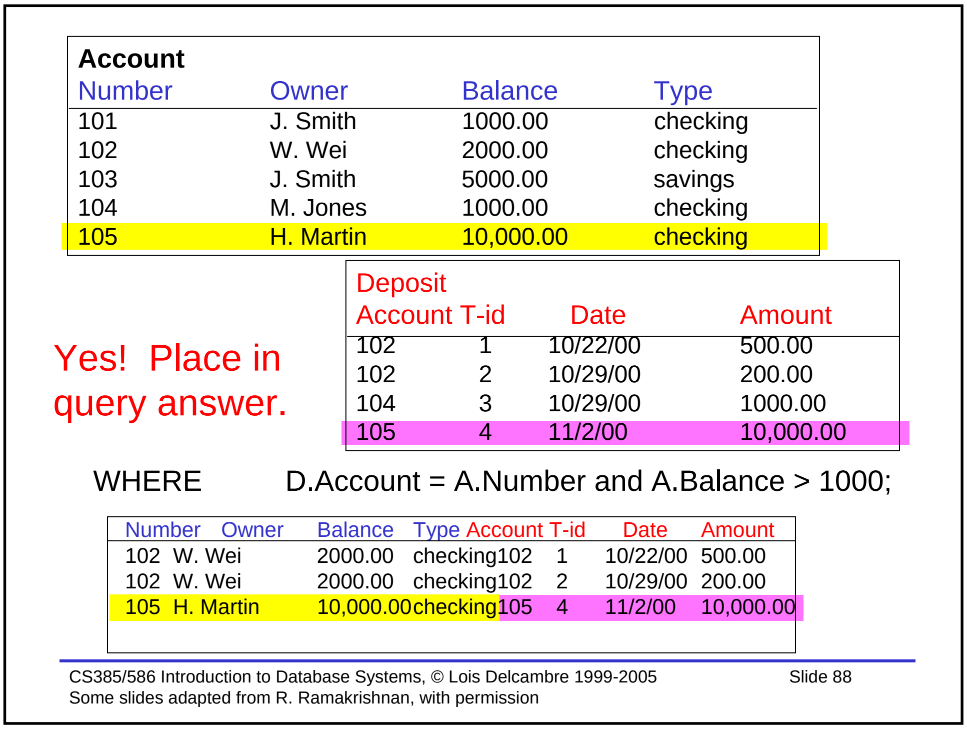
Deposit

Account	T-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	11/2/00	10,000.00

No! The first three fail.

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount
102	W. Wei	2000.00	checking	102	1	10/22/00	500.00
102	W. Wei	2000.00	checking	102	2	10/29/00	200.00



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

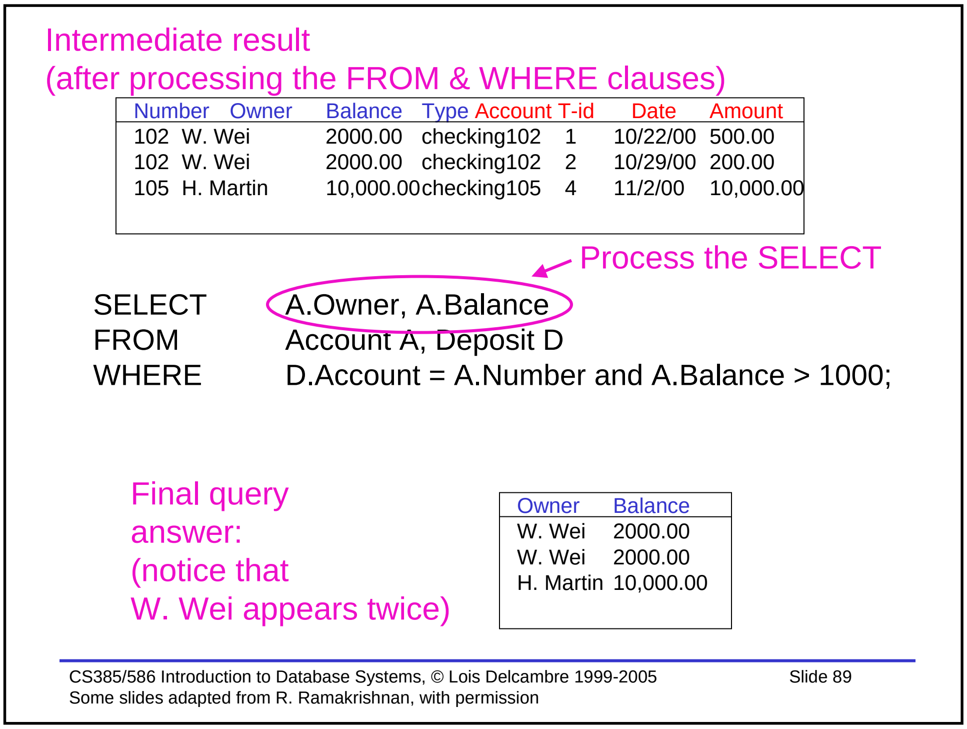
Deposit

Account	T-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	11/2/00	10,000.00

Yes! Place in query answer.

WHERE D.Account = A.Number and A.Balance > 1000;

Number	Owner	Balance	Type	Account	T-id	Date	Amount
102	W. Wei	2000.00	checking	102	1	10/22/00	500.00
102	W. Wei	2000.00	checking	102	2	10/29/00	200.00
105	H. Martin	10,000.00	checking	105	4	11/2/00	10,000.00



Intermediate result
(after processing the FROM & WHERE clauses)

Number	Owner	Balance	Type	Account	T-id	Date	Amount
102	W. Wei	2000.00	checking	102	1	10/22/00	500.00
102	W. Wei	2000.00	checking	102	2	10/29/00	200.00
105	H. Martin	10,000.00	checking	105	4	11/2/00	10,000.00

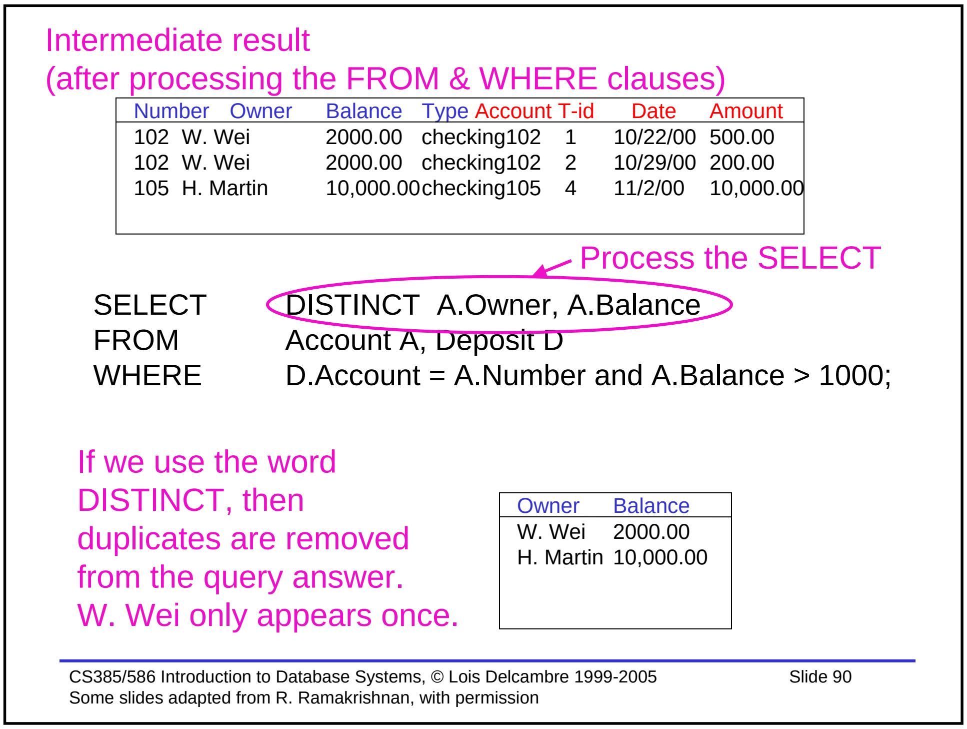
Process the SELECT

```
SELECT      A.Owner, A.Balance
FROM        Account A, Deposit D
WHERE       D.Account = A.Number and A.Balance > 1000;
```

Final query answer:
(notice that
W. Wei appears twice)

Owner	Balance
W. Wei	2000.00
W. Wei	2000.00
H. Martin	10,000.00

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 89
Some slides adapted from R. Ramakrishnan, with permission



Intermediate result
(after processing the FROM & WHERE clauses)

Number	Owner	Balance	Type	Account	T-id	Date	Amount
102	W. Wei	2000.00	checking	102	1	10/22/00	500.00
102	W. Wei	2000.00	checking	102	2	10/29/00	200.00
105	H. Martin	10,000.00	checking	105	4	11/2/00	10,000.00

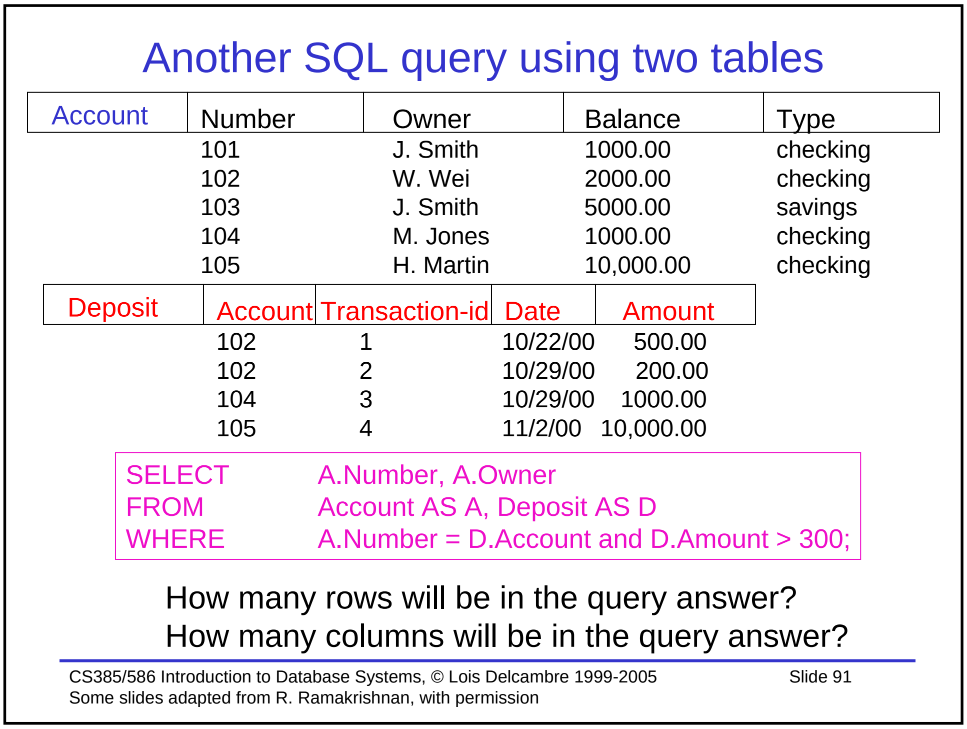
Process the SELECT

```
SELECT      DISTINCT  A.Owner, A.Balance
FROM        Account A, Deposit D
WHERE       D.Account = A.Number and A.Balance > 1000;
```

If we use the word DISTINCT, then duplicates are removed from the query answer.
W. Wei only appears once.

Owner	Balance
W. Wei	2000.00
H. Martin	10,000.00

CS385/586 Introduction to Database Systems, © Lois Delcambre 1999-2005 Slide 90
Some slides adapted from R. Ramakrishnan, with permission



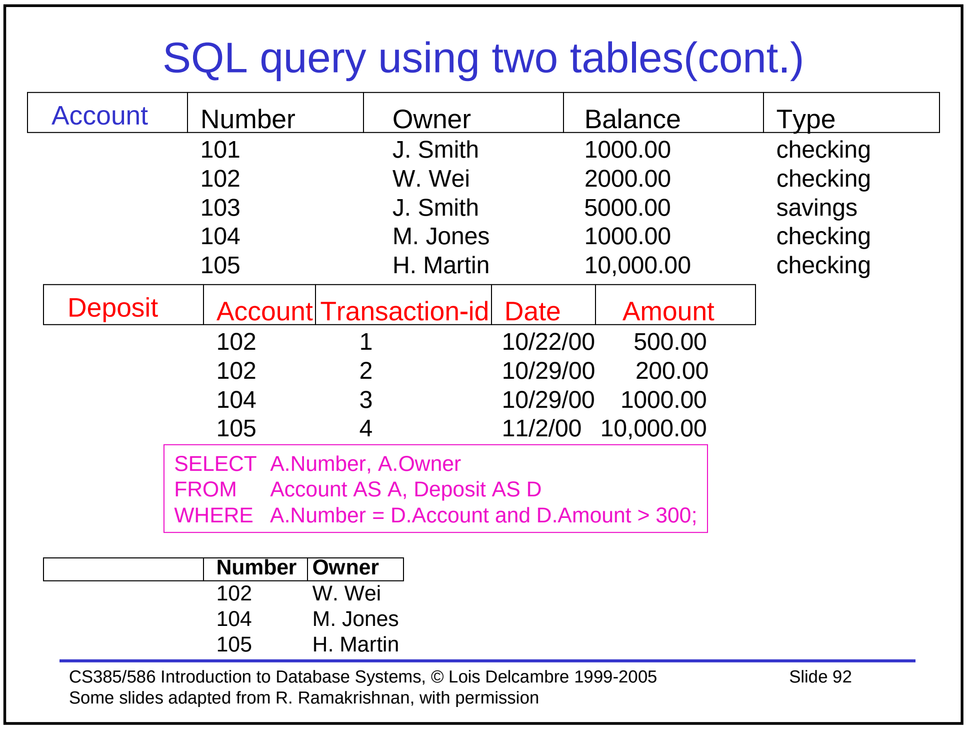
Another SQL query using two tables

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	Account	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	11/2/00	10,000.00

```
SELECT      A.Number, A.Owner
FROM        Account AS A, Deposit AS D
WHERE       A.Number = D.Account and D.Amount > 300;
```

How many rows will be in the query answer?
How many columns will be in the query answer?



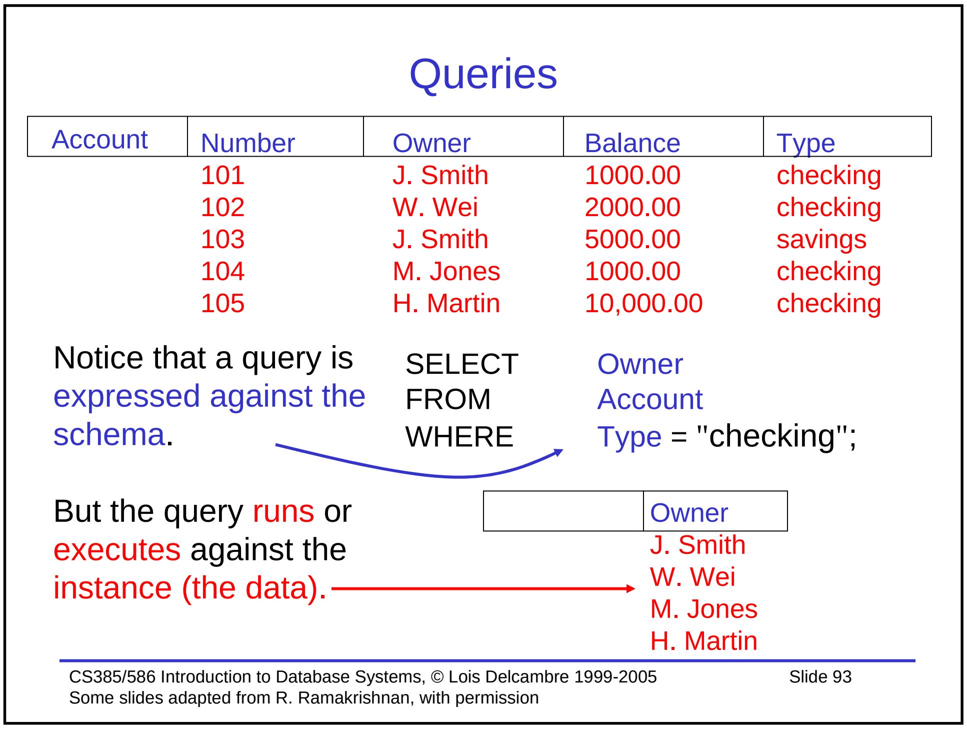
SQL query using two tables(cont.)

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	Account	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	11/2/00	10,000.00

```
SELECT  A.Number, A.Owner
FROM    Account AS A, Deposit AS D
WHERE   A.Number = D.Account and D.Amount > 300;
```

	Number	**Owner**
	102	W. Wei
	104	M. Jones
	105	H. Martin



Queries

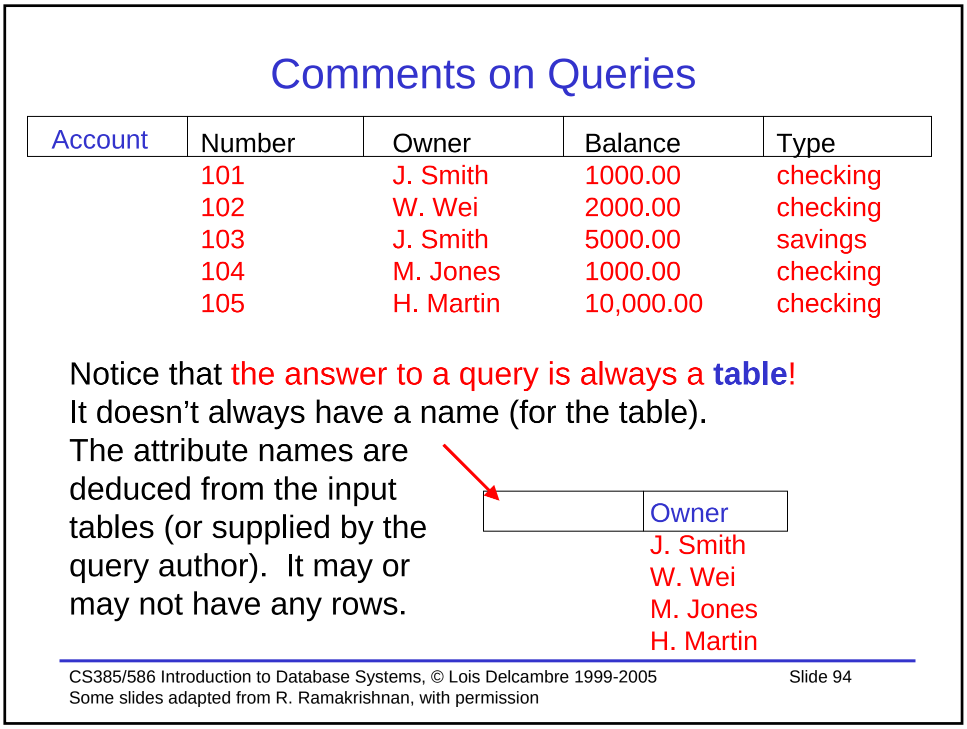
Account	Number	Owner	Balance	Type
	101	J. Smith	1000.00	checking
	102	W. Wei	2000.00	checking
	103	J. Smith	5000.00	savings
	104	M. Jones	1000.00	checking
	105	H. Martin	10,000.00	checking

Notice that a query is expressed against the schema.

```
SELECT   Owner
FROM     Account
WHERE    Type = "checking";
```

But the query runs or executes against the instance (the data).

	Owner
	J. Smith
	W. Wei
	M. Jones
	H. Martin

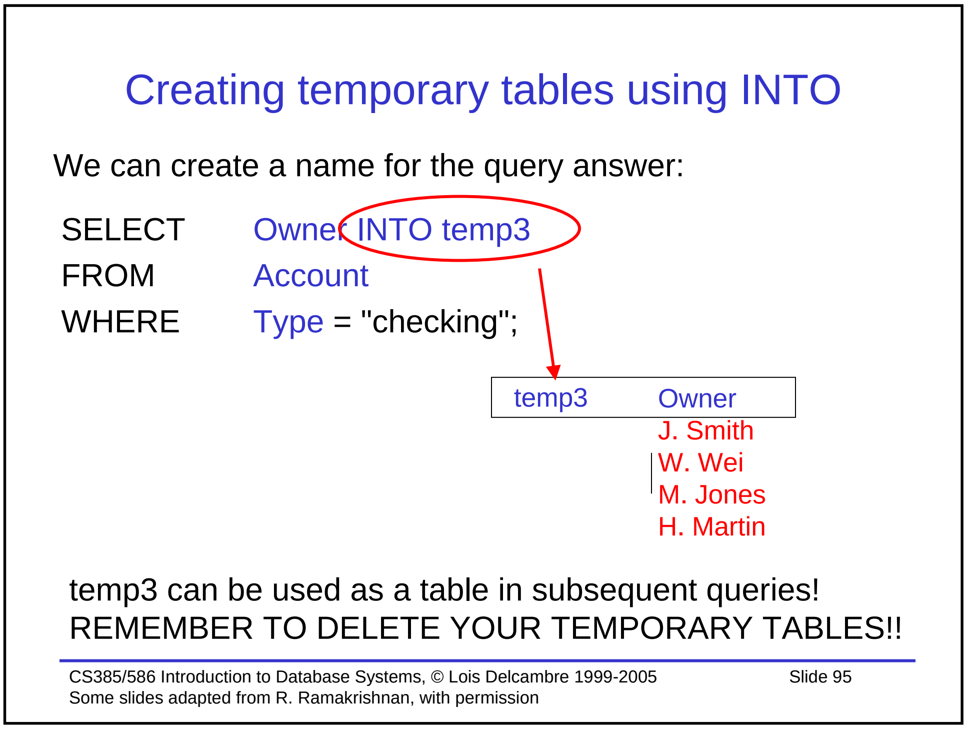


Comments on Queries

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

Notice that the answer to a query is always a **table**!
It doesn't always have a name (for the table).
The attribute names are deduced from the input tables (or supplied by the query author). It may or may not have any rows.

	Owner
	J. Smith
	W. Wei
	M. Jones
	H. Martin



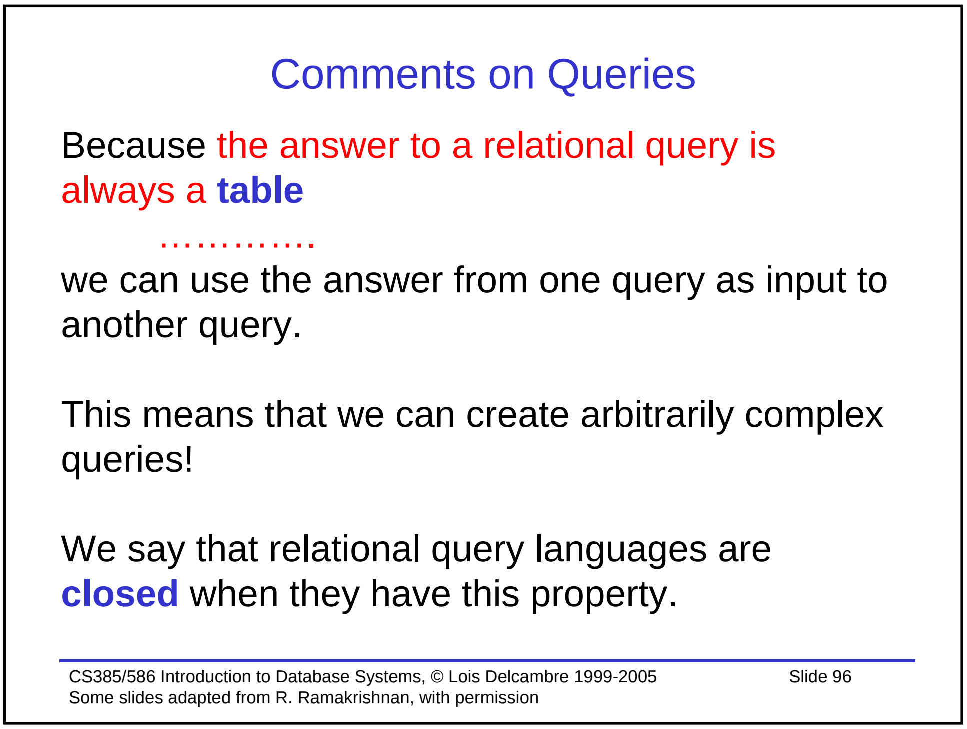
Creating temporary tables using INTO

We can create a name for the query answer:

```
SELECT      Owner INTO temp3
FROM        Account
WHERE       Type = "checking";
```

temp3	Owner
	J. Smith
	W. Wei
	M. Jones
	H. Martin

temp3 can be used as a table in subsequent queries!
REMEMBER TO DELETE YOUR TEMPORARY TABLES!!



Comments on Queries

Because the answer to a relational query is always a **table**

…………..

we can use the answer from one query as input to another query.

This means that we can create arbitrarily complex queries!

We say that relational query languages are **closed** when they have this property.