Overview of the Class and Introduction to DB schemas and queries

Lois Delcambre

CS 386/586 Introduction to Databases

Instructor: Lois Delcambre lmd@cs.pdx.edu 503 725-2405

TA: TBA

Office Hours: Immediately before and after class (in/near the classroom)

Others (face-to-face or phone) by appt.

This is a first class in relational database management systems (using SQL and relational algebra). If you've already had a class in relational database systems, you should talk to me; you probably don't need to take this class.

Prerequisites: CS 161 Intro to CS 1 (programming) and CS 250 Discrete Structures 1 (discrete math).

If you are a post-bac student, you could enroll as an undergraduate (in 386) or graduate (in 586). If you plan to work on an MS CS degree, you should consider taking 586.

Why is This Course in the Curriculum?

- It teaches valued job skills
- It integrates CS concepts
  - languages, data structures, concurrency
- The (digital) world runs on data
- It provides an example of the practical power (query optimizers) of an underlying theory (relational algebra)
- It is one of the few (only?) topics that focuses on information that is stored persistently, e.g., on disk (not in main memory).

Class web page

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

Contains complete class schedule including reading assignments, HW 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 – hopefully a day ahead. (One slide set per week)

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

Overview of the Syllabus

- Assignments - TBA
- First Exam OPEN BOOK: In class. Work by yourself. Ask questions only of the instructor or exam monitor.
- Second Exam OPEN BOOK: In class. Work by yourself. Ask questions only of the instructor or exam monitor.
- Third Exam OPEN BOOK: During finals week. Work by yourself. Ask questions only of the instructor or exam monitor.

Course Text

Database Systems: The Complete Book, 2nd Edition, *by Hector Garcia-Molina, Jeffery D. Ullman, and Jennifer Widom, Pearson/Prentice Hall, 2012.*

You may find it useful to have a SQL reference, e.g., ***SQL:1999*** by Melton – or other web-based resources.

Academic Integrity

You are responsible for knowing the PSU Academic Integrity Policy.

I feel that it is very important to enforce the academic integrity policy; please make sure you know what the rules are and follow them – for assignments and for tests.

Communication Mechanisms

- Class web page: Complete class schedule with links to assignments, lectures, and (often) sample answers for assignments. RELOAD the page every time you visit it. http://www.cs.pdx.edu/~lmd/cs386
- Announcements, Questions, and Answers: You must register to use the CS386/CS586 site on Piazza. To register, go here: https://piazza.com/pdx/fall2012/cs161
  to visit piazza go here: https://piazza.com/pdx/fall2012/cs161/home
  Be sure to supply your name; use an e-mail address that you check regularly. You can decide how often you want to receive notifications.
- Office hours:
  - Immediately before and after class
  - In person and telephone meetings, by request.
  - TA office hours: TBA

Piazza – what can a student do?

- Post a new question.
- "Edit" a question.
- Provide a "student answer".
- "Edit" a student answer.
- You can see the history of how it was changed; use the slider.
- Tag a question. For example, put #assignment_1 at the end of a question, if the question is about Assignment 1.
- Post a follow-up comment to a question.
  Note: post a new question if it is about something new.

Piazza – what can an instructor do?

- Post questions (just as students can) but questions will mostly come from students.
- Answer a question (in the instructor answer part).
- Mark a student answer as "good answer".
- Important: instructors can post "Notes"; I will use these Notes to make announcements about the class – including changes to deadlines (if they ever occur), corrections, etc. This is the only place where announcements will be made. Search for "instructor-note" or click on the #instructor-note tag to see all announcements.

Class Courtesy

Please …

- Be prompt
- Turn your cell phones off
- No headphones or earbuds
- One person talking at a time (except during in-class exercises)
- Use Piazza to ask appropriate questions/provide appropriate answers. (Instructors will monitor.)

Motivation for relational databases and queries

Using avsailors database

Consider the following files

sailors

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	40
85	Art	3	25.5
95	Bob	3	63.5

reserves

sid	bid	day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/5/98
74	103	9/8/98

boats

bid	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

How would you answer these questions? In a spreadsheet … or in a program …

- List the name of the red boats.
- List the id and name of the sailors who are over age 36.
- List the id and name of the sailors who are over age 36 or who have a rating that is less than 8.
- List the id and name of the sailors who reserved a boat on 9/8/98.
- List the name of the boats that were reserved on 10/10/98.
- For every reservation, list the reservation date, the sailor id, the sailor name, the boat id, the boat name, and the boat color.

Can you write programs to answer these questions generically?

- Original: List the names of the red boats.
- List the names of the <color> boats.
- List the <attributes> of the <color> boats.

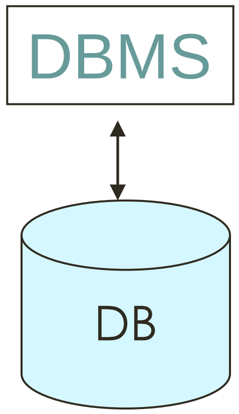
- Original: List the id and name of the sailors who are over age 36.
- List the <attributes> of <table name> where <attribute name> <comparator> <constant>.

- Notice: the last form of query shown just above could answer the first question above about about red boats.

What's a DB?

- database (DB) - a collection of persistent data

  Persistent: Lifetime not bound to a process
- database management system (DBMS) - a software system that supports the definition, population, and query of a database.



One assumption that we make for DB tables

For each table, every row has values for the same attributes – in the proper position. (Null values are okay.)

boats		
bid	**bname**	**color**
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

The third row doesn't match the other rows; it has an extra value that's not part of the schema (structure) for this table.

boats			
bid	**name**	**color**	
101	Interlake	blue	
102	Interlake	red	
103	Clipper	green	33'
104	Marine	red	

Tables can have keys and foreign keys

sailors

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35
64	Hor		
71	Zorb		
74	Hor		
85	Art		
95	Bob		

reserves

sailor	boat	day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98

boats

bid	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

A key for a table: one or more attributes whose values uniquely identify the rows in the table (for all future data). sid is unique for all sailors. The combination of (sailor, boat) is unique for all reserves

A foreign key in a table: one or more attributes whose values must match the values of a key in some table. reserves.sailor is a foreign key that references sailors.sid

Foreign key can't be violated (referential integrity)

- For a table that references another table (like bid in reserves), it must point to a valid row (e.g., to a bid that is in the boats table).
- The second row has an invalid bid (107) because there is not row in the boats table with 107.

  (Only part of the reserves table is shown here.)

boats

bid	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

reserves

sid	bid	day
22	101	10/10/98
22	107	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
…		

SQL Demo

- Class web page: http://cs.pdx.edu/~lmd/cs386
- Click on “link to DB information” (near the top)
- Click on “PostgreSQL query page”
- Click on “Database Class” in left panel
- Log in … with userid: introdb_readonly
  and password: introdb
- Click on “introdb_sailors”
- Click on “public”
- You’ll see the three tables in the Sailors DB

SQL demo (cont.)

- Click “Browse” on each of the three tables; you can see the data in the database
- Click “SQL” in the upper right corner; you should see a separate, small window.
- Try out some queries like this:
- SELECT name FROM boats WHERE color = ‘red’
- SELECT sid, sname FROM sailors WHERE age > 36
- SELECT sid, sname FROM sailors WHERE age > 36 OR rating < 8
- SELECT sailors.sid, sailors.sname
  FROM sailors JOIN reserves USING (sid)
  WHERE day = '10/10/98’

High-level introduction to databases

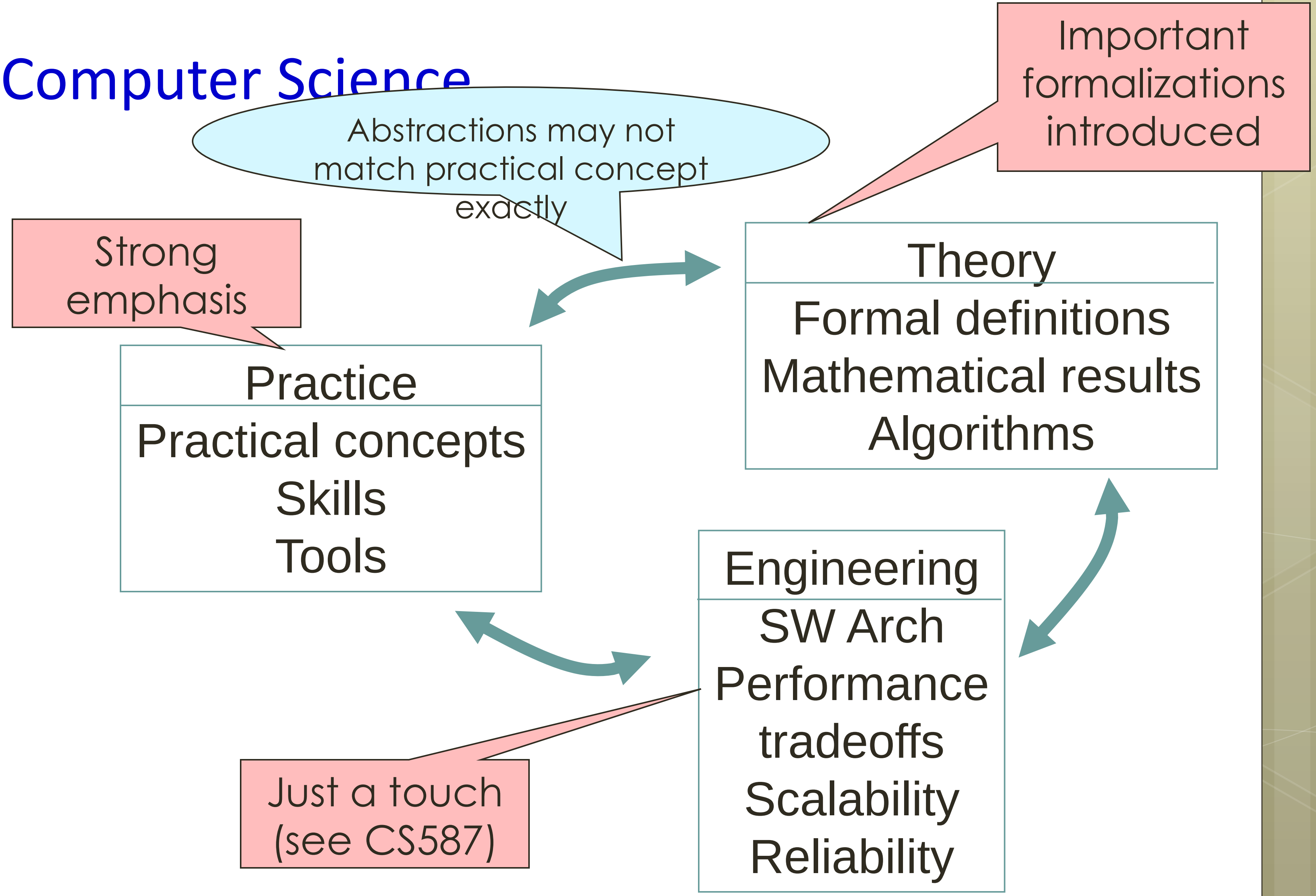
What is computer science?

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



Practice and Theory in Database

Practice

- Tables, columns, rows, keys
- SQL
- Application structure
- Logical & physical database design
- Transactions
- Security

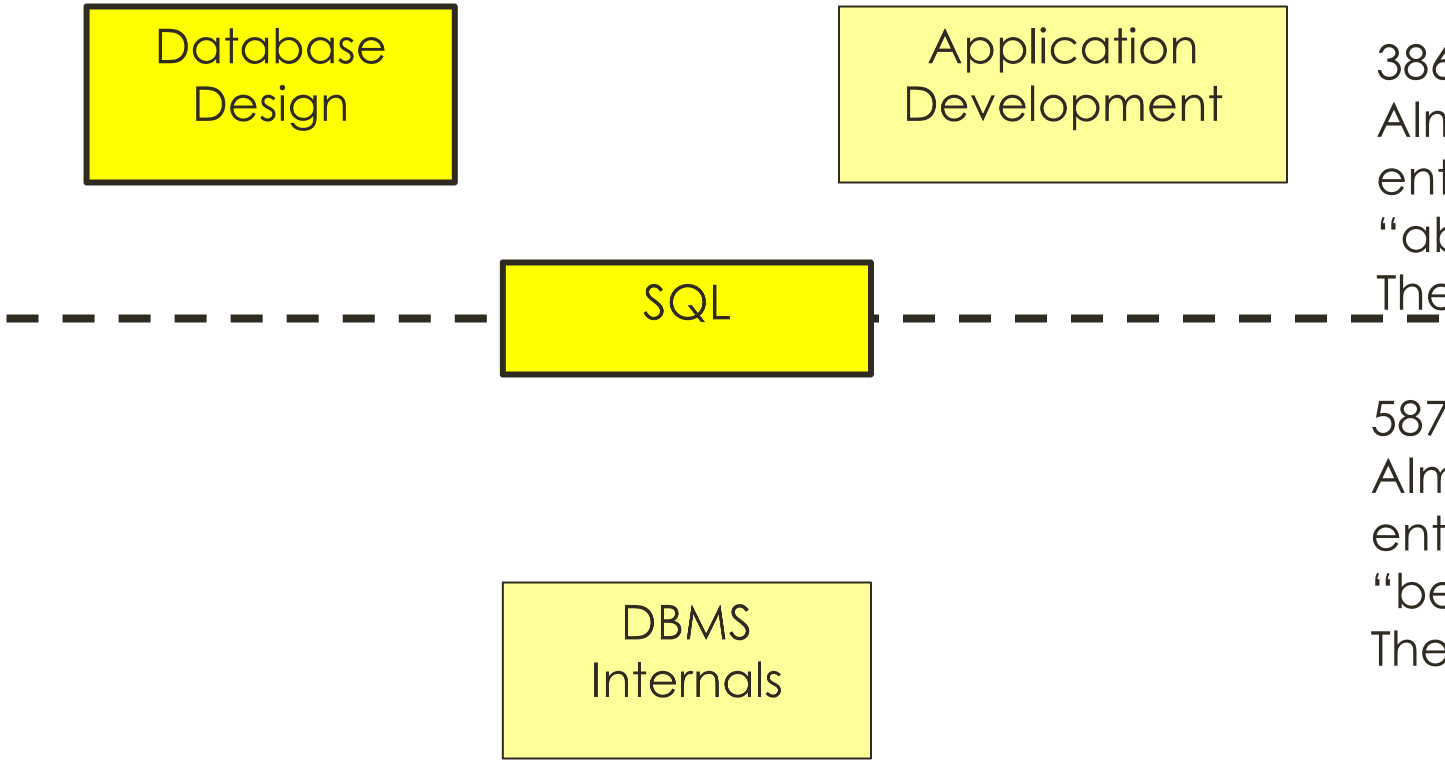
Theory

- Relational model: relations, attributes, tuples
- Relational algebra, equivalences
- Functional dependencies, normalization
- Schedules, serializability

Engineering (Design tradeoffs) in DB

- Basic system structure
- Storage and Indexing
- Query evaluation (operators, optimization)

Where Does This Course Focus?



386/586: Almost entirely "above The line"

587 (510) Almost entirely "below The line"

Terminology

Table Structure: the schema for the table

The *name* of the table

The name of the *columns* (*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

Table Rows

Account

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

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

Table Instance

An *instance* of the table...

the current contents or data in the table.

Account

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

Another Table Instance

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

Account

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

new

Intension vs. Extension

The *intension* of the table

Account

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

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

Database (One or More Tables)

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

Deposit	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	11/02/00	10,000.00

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

Table Keys

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

Deposit	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	11/02/00	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.

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

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

key

foreign key

key

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	110/2/00	10,000.00

foreign key

key

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.

Database Domains for Columns

Account

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

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

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

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

Specification of a Relational Schema

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

There can be more than one key for a table.

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 ";
```

Notice that all SQL statements end with a semicolon (but PostgreSQL doesn't allow a semicolon at the end of a query).

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.