

Topic 1: SQL

L01: SQL introduction

Wolfgang Gatterbauer

CS3200 Database design (fa22)

<https://northeastern-datalab.github.io/cs3200/fa22s3/>

9/7/2022

What is a DBMS (Database management system)?

A Database Management System (DBMS) is a piece of software designed to store and manage databases

- A large, integrated collection of data
- Models a real-world enterprise
 - Entities (e.g., Students, Courses)
 - Relationships (e.g., Alice is enrolled in cs3200)

A Motivating, Running Example

- Consider building a Course Management System (CMS):

- Students
- Courses
- Professors

} *Entities*

- Who takes what
- Who teaches what

} *Relationships*

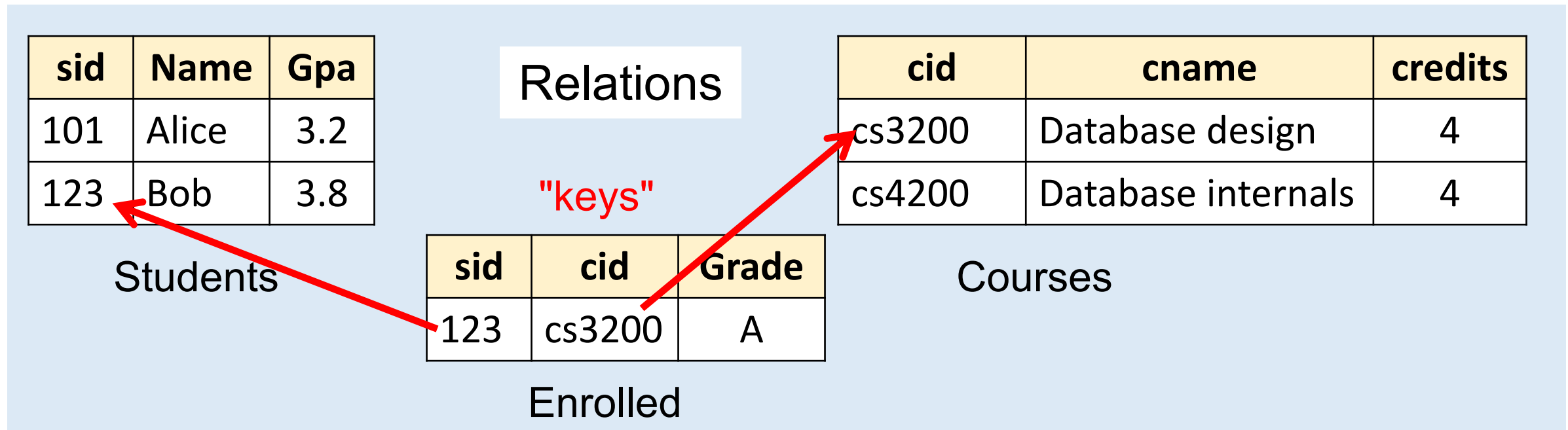
Data models

- A **data model** is a collection of concepts for describing data
 - The relational model of data is the most widely used model today
 - Main Concept: a relation (which is basically the same as a table)
- A **schema** is a description of a particular collection of data, using the given data model
 - E.g. every relation in a relational data model has a schema describing the number of columns and their types

Modeling the CMS

- Logical Schema

- Students(sid: string, name: string, gpa: float)
- Courses(cid: string, cname: string, credits: int)
- Enrolled(sid: string, cid: string, grade: string)

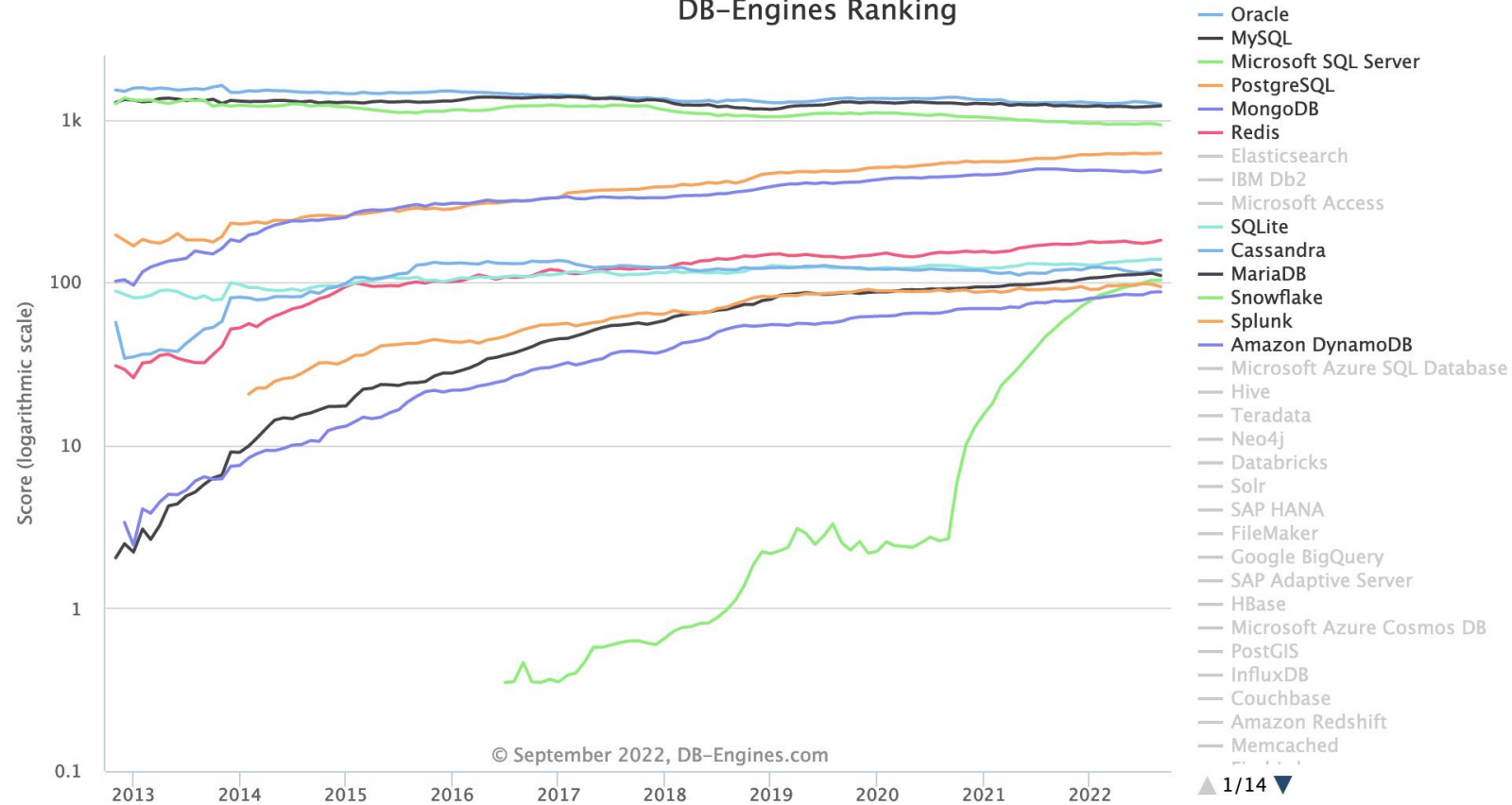


DBMS we are using:
PostgreSQL

Why PostgreSQL instead of MariaDB (or MySQL)



DB-Engines Ranking



Why PostgreSQL instead of MariaDB (or MySQL)



Although PostgreSQL has been around for a while, the relative **decline of MySQL** has made it a serious contender for the title of most used open source database. Since it works very similarly to MySQL, developers who prefer open source software are converting in droves.

Advantages

- By far, PostgreSQL's most mentioned advantage is the efficiency of its central algorithm, which means it outperforms many databases that are advertised as more advanced. This is especially useful if you are working with large datasets, for which I/O processes can otherwise become a bottleneck.
- It is also one of the most flexible open source databases around; you can write functions in a wide range of server-side languages: Python, Perl, Java, Ruby, C, and R.
- As one of the most commonly used open source databases, PostgreSQL's community support is some of the best around.

I also prefer PostgreSQL over MySQL because it has a more principled interpretation of SQL (and a powerful EXPLAIN command)

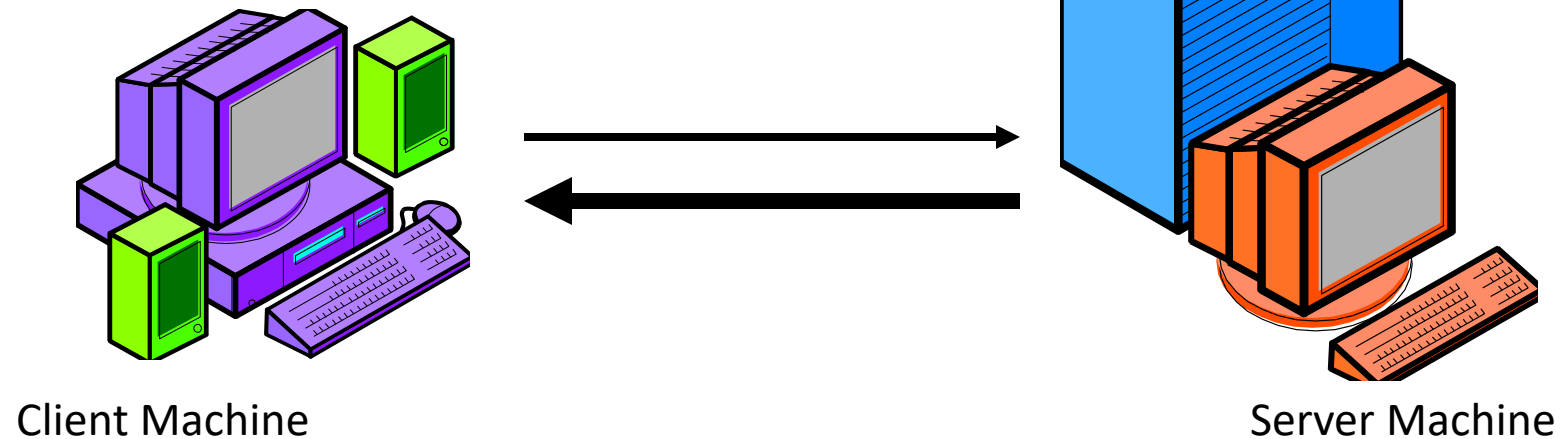
The screenshot shows a web browser displaying a Guardian article. The URL is <https://www.theguardian.com/info/2018/nov/30/bye-bye-mongo-hello-postgres>. The page features a dark blue header with the text 'Support The Guardian' and 'Available for everyone, funded by readers'. Below this are 'Contribute' and 'Subscribe' buttons. A navigation bar includes 'News', 'Opinion', 'Sport', 'Culture', 'Lifestyle', and 'More'. The article title is 'Bye bye Mongo, Hello Postgres'. The byline lists 'Philip McMahon, Maria-Livia Chiorean, Susie Coleman and Akash Askoolum'. The date is 'Fri 30 Nov 2018 05.36 EST'. There are social media icons for Facebook, Twitter, and Email, and a share count of 438. The main image shows an elephant using its trunk to pick up a large bundle of greenery. A caption below the image reads: '▲ An elephant picking up some greenery. Photograph: Michael Sohn/AP'.

Client/Server Architecture

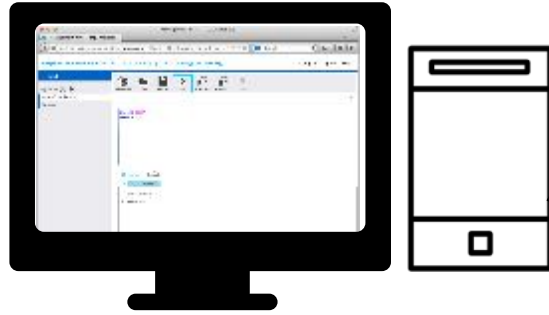
- There is a single server that stores the database (called DBMS or RDBMS):
 - Could be a beefy system in the cloud (e.g. SQL Azure, or any DBMS on EC2)
 - But can be your own desktop...
 - ... or a huge cluster running a parallel DBMS
- Many clients run apps and connect to DBMS
 - E.g., PGAdmin
 - In most applications, some Java, Python, or C++ program
- Clients “talk” to server using some protocol

Client-server Computing

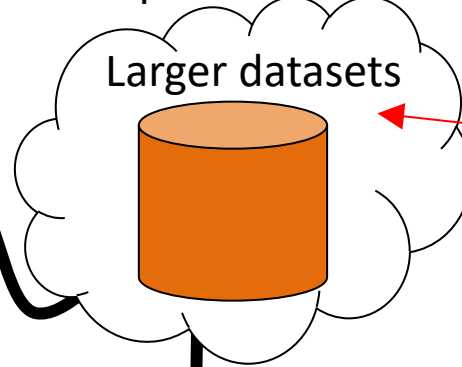
- A system which allows the synchronized division of processing between two or more computers
 - Client programs run on client machine(s)
 - Server programs run on server machine(s)
 - Co-ordinate and work together to do the required processing
 - “Few serving many” model



pgAdmin via browser

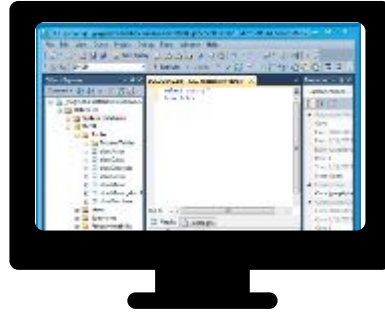


Cloud provider like Amazon, Microsoft

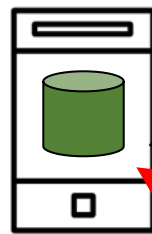


One database for all clients.

pgAdmin via browser

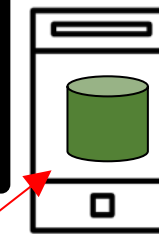
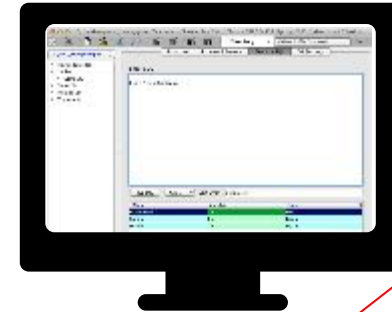


Postgres



IMDB

Firefox & SQLite manager



IMDB

You "own" your own local database(s)



What is the most widely deployed database? SQLite!

SQLite

- open source & cross-platform
- easy to install
- has no server ("embedded")
- ideal for single-user application; has limitations when it comes to concurrency / simultaneous transactions (one writer at a time)
- does not allow partitioning; everything is stored in one single file
- extra functions are written in C/C++
- is used by quasi every web browser in the world ...

PostgreSQL

- open source & cross-platform
- takes a bit longer to install
- uses a server
- ideal for shared repository; allows concurrency (many simultaneous transactions), locking and fine-grained access control
- scales to >GB easily; allows partitioning (distributing) the data across several files / nodes
- supports user-defined functions

SQL overview

SQL Introduction

- SQL is a standard language for querying and manipulating data
- SQL is a very high-level programming language
 - This works because it is optimized well!
- Many standards out there:
 - ANSI SQL, SQL92 (a.k.a. SQL2), SQL99 (a.k.a. SQL3),
 - Vendors support various subsets
 - We focus on the most commonly used constructs in SQL

SQL stands for
Structured Query Language

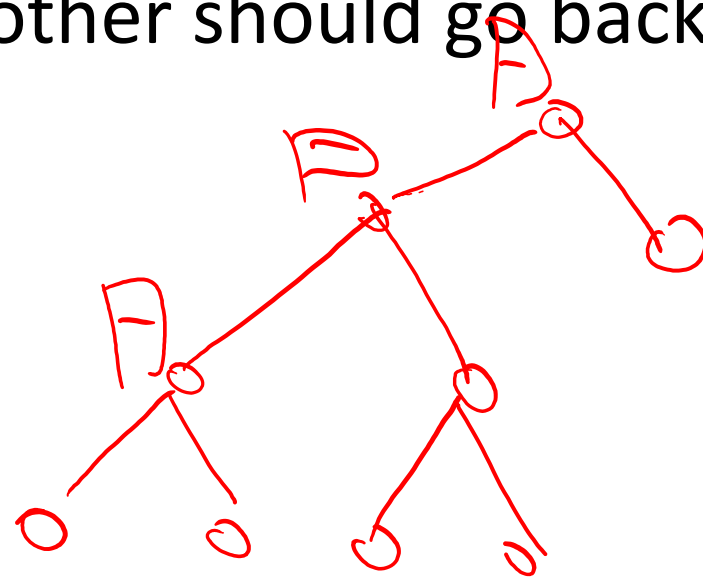
NB: Probably the world's most successful **parallel** programming language (multicore?)

SQL Has Three Major Sub-Languages

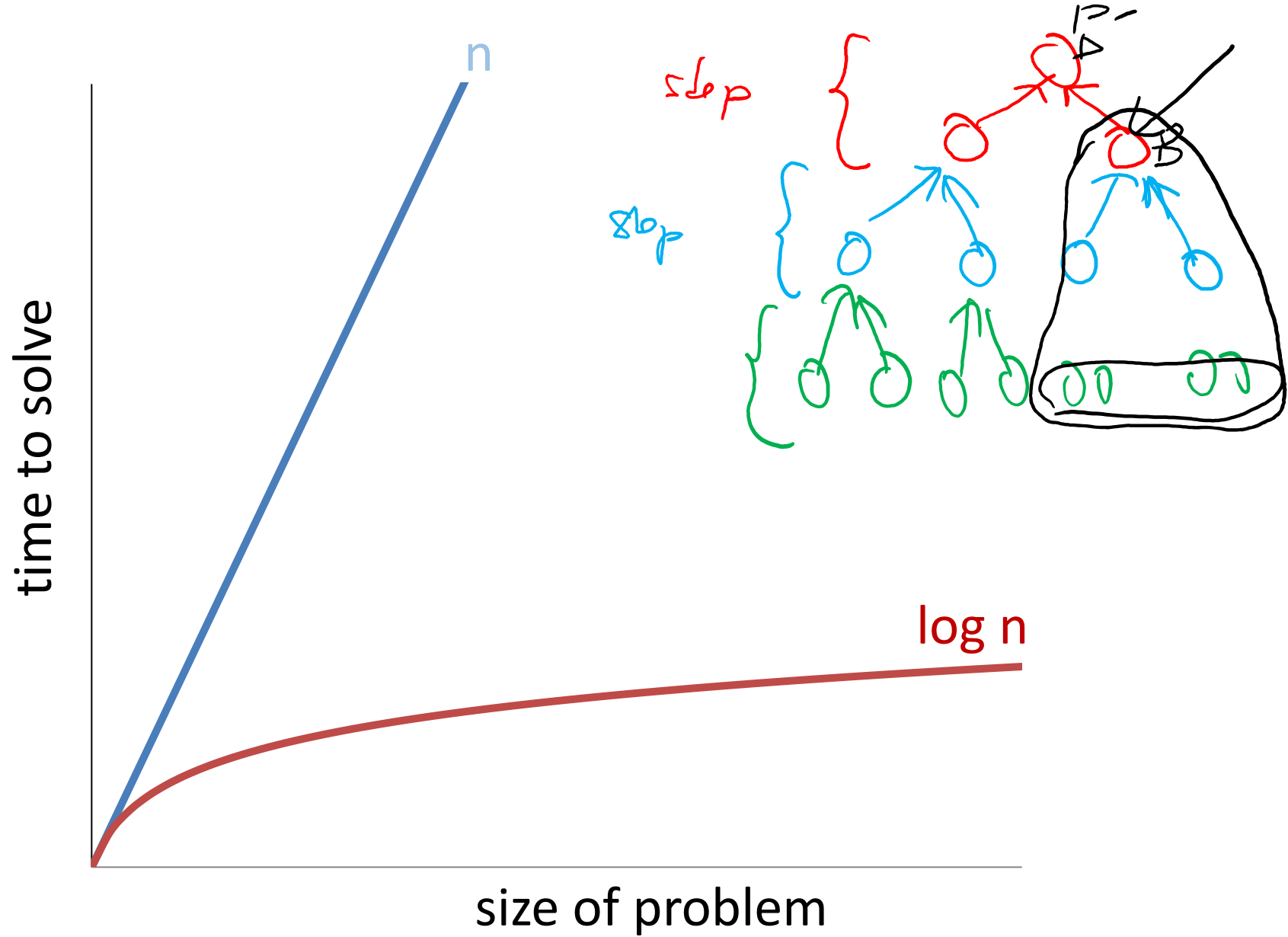
- Data Manipulation Language (DML)
 - Insert/delete/modify tuples in tables
 - Commands that maintain and **query a database** (our main focus!)
- Data Definition Language (DDL)
 - Define a relational schema (create, alter, and drop tables; establish constraints)
 - Create/alter/drop tables and their attributes
- Data Control Language (DCL)
 - Commands that control a database, including administering privileges and committing data

An Algorithm

- Stand up and think of the number 1
- Pair off with someone standing, add your numbers together, and adopt the sum as your new number
- One of you should sit down; the other should go back to step 2



Scalability



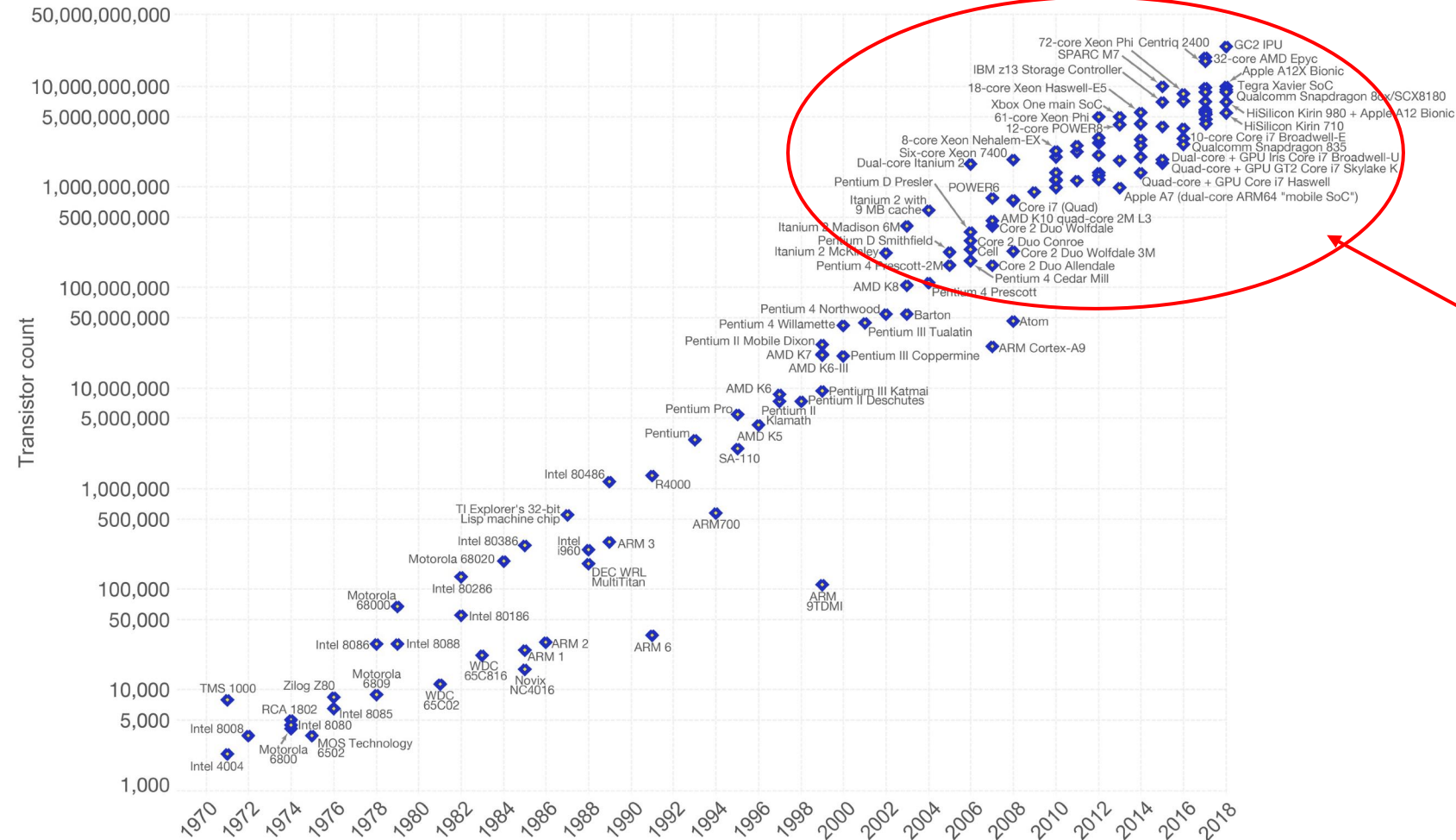
Most spectacular these days: theoretic potential for perfect scaling!

- perfect scaling
 - given sufficient resources, performance does not degrade as the database becomes larger
- key: parallel processing
- cost: number of processors polynomial in the size of the DB
 - remember our in-class counting exercise
- all (most) relational operators highly parallelisable

Moore's law

Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



Multi-cores

Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at [OurWorldinData.org](https://ourworldindata.org). There you find more visualizations and research on this topic.

Licensed under CC-BY-SA by the author Max Roser.

What is SQL?

The Positives

- It's a language (like English, Spanish, German, ...)
- There are only a few key words that you have to learn – it's fairly simple
- It's major purpose is to communicate with a database and ask a database for data
- It's a declarative language (you define what to do)

The Challenges

- Simplicity has it's cost – it gets complex quickly
 - Imagine only having 2 verbs (go, put, wait) to express all you do in a lifetime
 - It's either infeasible or you have to combine a lot basic actions to construct a more complex action
(e.g. skydiving = put parachute into backpack, put the backpack on your back, go airplane, wait until airplane is at 14k feet, go to open door, go outside airplane, ...)
- Declarative programming is perceived as non-intuitive (well, decide for yourself 😊)

Compare semantics between Excel and Database tables

Excel

	A	B	C	D
1	PName	Price	Category	Manufacturer
2	Gizmo	19.99	Gadgets	GizmoWorks
3	PowerGizmo	29.99	Gadgets	GizmoWorks
4	SingleTouch	149.99	Photography	Canon
5	MultiTouch	203.99	Household	Hitachi

table heading

row

column

Database¹

Table name

TABLE	Product	Search	Show All	
rowid	PName	Price	Category	Manufacturer
1	Gizmo	19.99	Gadgets	GizmoWorks
2	PowerGizmo	29.99	Gadgets	GizmoWorks
3	SingleTouch	149.99	Photography	Canon
4	MultiTouch	203.99	Household	Hitachi

attribute name

tuple/ entity/
record/ row

attribute/ field/
column

¹A Database (DB) is simply a system that holds multiple tables (like Excel has multiple sheets)

Tables in SQL

Attribute names

Table name

Key

<u>PName</u>	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Tuple / row
(Entity)

Attribute

Data Types in SQL

- Atomic types
 - Character strings: CHAR(20), VARCHAR(50)
 - Numbers: INT, BIGINT, SMALLINT, FLOAT
 - Others: MONEY, DATETIME, ...
- Record (aka tuple)
 - Every attribute must have an atomic type
- Table (aka relation)
 - A set of tuples (hence tables are flat!)

Table Schemas

- The schema of a table is the table name, its attributes, and their types:

```
Product(Pname: string, Price: float,  
        Category: string, Manufacturer: string)
```

- A key is an attribute whose values are unique; we underline a key

Basic SQL

SQL Query

- Basic form (there are many many more bells and whistles)

```
SELECT <attributes>  
FROM   <one or more relations>  
WHERE  <conditions>
```

Call this a SFW query.

You can run our queries

If you like to run the query for now, visit:

<https://bit.ly/3hmLFt0>

(for <http://sqlfiddle.com/#!17/a9eb13/1>)

Our friend here shows that you can follow along in Postgres. Just install the database from the text file "302 - ..." available in our sql folder on Canvas



302

The screenshot shows the SQL Fiddle interface. The left pane contains SQL code for dropping tables and a query. The right pane shows the query results. Below the panes are buttons for 'Build Schema', 'Edit Fullscreen', 'Browser', and 'Run SQL'. At the bottom, a table displays the results of the query, and a green bar shows 'Record Count: 2; Execution Time: 5ms' with a 'View Execution Plan' link.

```
4 -----  
5  
6  
7  
8 -----  
9 -- Drop tables if they already exist  
10 -----  
11  
12 DROP TABLE IF EXISTS Product;  
13 DROP TABLE IF EXISTS Company;  
14  
15
```

```
1 SELECT *  
2 FROM Product  
3 WHERE category='Gadgets'  
4
```

pname	price	category	manufacturer
Gizmo	19.99	Gadgets	GizmoWorks
PowerGizmo	29.99	Gadgets	GizmoWorks

✓ Record Count: 2; Execution Time: 5ms [View Execution Plan](#) [link](#)

Did this query solve the problem? If so, consider donating \$5 to help make sure SQL Fiddle will be here next time you need help with a database problem. Thanks!

Simple SQL Query

If you like to run the query for now, visit:

<https://bit.ly/3hmLFt0>

(for <http://sqlfiddle.com/#17/a9eb13/1>)

Our friend here shows that you can follow along in Postgres. Just install the database from the text file "302 - ..." available in our sql folder on Canvas



302

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT *  
FROM Product  
WHERE category= 'Gadgets '
```



Simple SQL Query

If you like to run the query for now, visit:

<https://bit.ly/3hmLFt0>

(for <http://sqlfiddle.com/#17/a9eb13/1>)

Our friend here shows that you can follow along in Postgres. Just install the database from the text file "302 - ..." available in our sql folder on Canvas

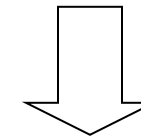


302

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT *  
FROM Product  
WHERE category= 'Gadgets '
```



Selection

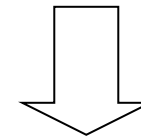
PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks

Simple SQL Query

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

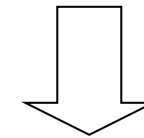
```
1 SELECT pName, price, manufacturer  
2 FROM Product  
3 WHERE price > 100
```



Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT pName, price, manufacturer
FROM Product
WHERE price > 100
```



**Selection
& Projection**

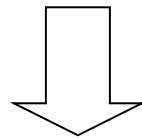
PName	Price	Manufacturer
SingleTouch	\$149.99	Canon
MultiTouch	\$203.99	Hitachi

Selection vs. Projection

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT pName, price  
FROM Product  
WHERE price > 100
```



Selection vs. Projection

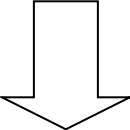
Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

One **projects** onto some attributes (columns)
-> happens in the **SELECT** clause

```
SELECT pName, price
FROM Product
WHERE price > 100
```

One **selects** certain entires=tuples (rows)
-> happens in the **WHERE** clause
-> acts like a **filter**



PName	Price
SingleTouch	\$149.99
MultiTouch	\$203.99

SQL: A Few Details on Syntax

- SQL commands are case insensitive:
 - SELECT = Select = select
 - Product = product, Category = category
- But values are not:
 - Different: 'Gadgets', 'gadgets'
 - (Notice: in general, but default settings will vary from DBMS to DBMS. E.g. MySQL is case insensitive. Just to be safe, always assume values to be case sensitive!)

```
WHERE LOWER(Category) = 'gadgets'
```

- Use single quotes for constants:
 - 'abc' - yes
 - "abc" - no (except MySQL and SQLite)

Eliminating Duplicates



Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
PowerGizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Our colorful hands represent "team exercises"

```
SELECT category  
FROM Product
```



Eliminating Duplicates



Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
PowerGizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

- ✓
- ✓
- ✓
- ✓

Set vs. Bag semantics

```
SELECT category
FROM Product
```



Category
Gadgets
Gadgets
Photography
Household



Category
Gadgets
Photography
Household

Eliminating Duplicates



Product			
PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
PowerGizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Set vs. Bag semantics

"Relational completeness"

```
SELECT category  
FROM Product
```



Category
Gadgets
Gadgets
Photography
Household

{'G', 'G', 'P'}

```
SELECT DISTINCT category  
FROM Product
```



Category
Gadgets
Photography
Household

Sorting ("Ordering")

Ordering the Results

PName	Price	Category	Manufacturer
Gizmo	19.99	Gadgets	GizmoWorks
PowerGizmo	29.99	Gadgets	GizmoWorks
SingleTouch	149.99	Photography	Canon
MultiTouch	203.99	Household	Hitachi

```
SELECT pName, price, manufacturer
FROM Product
WHERE category='Gadgets'
AND price > 10
```

pName	price	manufacturer
k	5	...
B	10	...
C	10	...

Assume you like to sort (order) the results by price,
and break ties in price by alphabetical order in pName



Ordering the Results

Product

PName	Price	Category	Manufacturer
Gizmo	19.99	Gadgets	GizmoWorks
PowerGizmo	29.99	Gadgets	GizmoWorks
SingleTouch	149.99	Photography	Canon
MultiTouch	203.99	Household	Hitachi

```
SELECT pName, price, manufacturer
FROM Product
WHERE category='Gadgets'
AND price > 10
ORDER BY price, pName
```

pName	price	manufacturer
k	5	...
B	10	...
C	10	...

Assume you like to sort (order) the results by price,
and break ties in price by alphabetical order in pName

Ordering is ascending by default. To get descending:



	5	
↑ C	10	
B	10	↓

Ordering the Results

Product

PName	Price	Category	Manufacturer
Gizmo	19.99	Gadgets	GizmoWorks
PowerGizmo	29.99	Gadgets	GizmoWorks
SingleTouch	149.99	Photography	Canon
MultiTouch	203.99	Household	Hitachi

```
SELECT pName, price, manufacturer
FROM Product
WHERE category='Gadgets'
AND price > 10
ORDER BY price, pName
```

pName	price	manufacturer
k	5	...
B	10	...
C	10	...

Assume you like to sort (order) the results by price, and break ties in price by alphabetical order in pName

Ordering is ascending by default. To get descending:

```
ORDER BY price DESC, pname ASC
```

Product

PName	Price	Category	Manufacturer
Gizmo	19.99	Gadgets	GizmoWorks
PowerGizmo	29.99	Gadgets	GizmoWorks
SingleTouch	149.99	Photography	Canon
MultiTouch	203.99	Household	Hitachi



```
SELECT DISTINCT category  
FROM Product  
ORDER BY category
```



```
SELECT category  
FROM Product  
ORDER BY pName
```



```
SELECT DISTINCT category  
FROM Product  
ORDER BY pName
```

