L01-L06: SQL

CS3200 Database design (fa18 s2)

https://northeastern-datalab.github.io/cs3200/

Version 9/6/2018

L01: SQL introduction

CS3200 Database design (fa18 s2)

https://northeastern-datalab.github.io/cs3200/

Version 9/6/2018

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

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

<u>SQL</u> stands for <u>S</u>tructured <u>Q</u>uery <u>L</u>anguage

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

Microprocessor Transistor Counts 1971-2011 & Moore's Law



Moore's law

Date of introduction

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 ⁽ⁱ⁾)

Compare semantics between Excel and Database tables

Excel

	А	В	С	D	_	
1	PName	Price	Category	Manufacturer	table heading	
2	Gizmo	19.99	Gadgets	GizmoWorks	_	
3	PowerGizmo	29.99	Gadgets	GizmoWorks		
4	SIngleTouch	149.99	Photography	Canon	_	
5	MultiTouch	203.99	Household	Hitachi	row	
	column					

Table name

Database ¹	TABLE Pr	oduct	Searc <u>h</u>	Show All		
	rowid	PName	Price	Category	Manufacturer	attribute
	1	Gizmo	19.99	Gadgets	GizmoWorks	name
	2	PowerGizmo	29.99	Gadgets	GizmoWorks	
	3	SingleTouch	149.99	Photography	Canon	
	4	MultiTouch	203.99	Household	Hitachi	tuple/ entity
	İ				attribute/ field/ column	record/ row

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

Tables in SQL

Attri	bute names	Table nam	e	
	Product	Key		
L	PName	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(<u>Pname: string, Price: float,</u> Category: string, Manufacturer: string)

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

Basic SQL



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

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

Call this a <u>SFW</u> query.

Simple SQL Query

Our friend here shows that you can follow along in SQLite. Just install the database from the text file "300 - ..." available in our sql folder



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

Simple SQL Query

Our friend here shows that you can follow along in SQLite. Just install the database from the text file "300 - ..." available in our sql folder



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

Practice with our small example databases



If you are using Windows:

- 1. Download the appropriate text files from our repository
- 2. Open them with "Wordpad" (not "Notepad" which messes up the text!)
- 3. Paste the SQL commands into <u>http://sqffiddle.com</u> or <u>https://www.db-fiddle.com/</u> and execute

Name Date modified duick access > 301 - Small IMDB for SOLity 258 KB Lenovo (Bax) /2014 2:19 AM Microsoft SOL Ser 302 - Simple products - SQLite 11/2/20 Text Document 2 KB Desktop 303 - Simple products - SQLserver 2 %8 Downloads 🦿 🗢 | 303 - Simple products - SQLser 303 - Simple products - SQLserver -× File Edit Format View Help View A Find **Courser New** ample SQLserver commands for 45881/70455 Modern Data Management Replace Wolfgang Gatterbauer-- v151102 Paste Select all Drop tables if they already exis Clipboard Editing -----if exists (select 1 1 1 1 5 1 'product' and type = 'U' sys.tables here name EGIN drop table Product ENDI exists (select * from ys.tables and type = 'U') BEGIN drop table Company END where name = 'Company -- Example SQLaerver commands for 45881/70455 Modern Data Management Create the tablesreate table Company (-- Wolfgang Gatterbauer CName char(20) PRIMARY KEY, StockPrice 1 Country char -- v151102 PName char(20), (20)); create table Product Price decimal anufacturer char(20), PRIMARY (9, 2). Category char(2c) REFERENCES Company(CName)); KEY (PName), FOREIGN KEY (Manufacture) Popu te the tables -- Drop tables if they already exist insert into Company values ("GizmoWorks", -------25, 'USA'); insert into Company values (Canon', 65, 'Japan'); insert into Company values ('Hitachi', 15, 'Japan'); insert into Product if exists (values ('Gizmo', 19.99, 'Gadgets', 'GizmoWork'); insert into Product select * values ('PowerGizmo', 29.99, 'Gadgets', 'Gizmowerks'); insert into from sys.tables Product values ('SingleTouch', 149.99, 'Photography', 'Canon'); insert where name = 'product' into Product values ('MultiTouch', 203.99, 'Household', 'Hitachi'); and type = 'U') ----- OPTIONAL: Show all foreign key BEGIN relationships referencing our table company drop table Product ----- SELECT * FROM sys.foreign_keysNHERE END referenced object id = object id('company') if exists select * from sys.tables where name = 'Company' and type = 'U') 100% (=) ٤

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

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

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



SELECT pName, price FROM Product WHERE price > 100

PName	Price
SingleTouch	\$149.99
MultiTouch	\$203.99

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