

Lectures 2: Introduction to SQL

Lecture 2: SQL Part I

Outline

1. SQL introduction & schema definitions
 - ACTIVITY: Table creation
2. Basic single-table queries
 - ACTIVITY: Single-table queries!
3. Multi-table queries
 - ACTIVITY: Multi-table queries!

1. SQL Introduction & Definitions

What you will learn about in this section

1. What is SQL?
2. Basic schema definitions
3. Keys & constraints intro

SQL Motivation

- Dark times 5 years ago.
 - Are databases dead?
- Now, as before: everyone sells SQL
 - Pig, Hive, Impala
 - The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models.
- “Not-Yet-SQL?”



Basic SQL

SQL Introduction

- SQL is a standard language for querying and manipulating data

- SQL is a **very high-level** programming language

SQL stands for
Structured Query Language

- Many standards out there:
 - ANSI SQL, SQL92 (a.k.a. SQL2), SQL99 (a.k.a. SQL3),
 - Vendors support various subsets

SQL is a...

- Data Definition Language (DDL)
 - Define relational *schemata*
 - Create/alter/drop tables and their attributes
- Data Manipulation Language (DML)
 - Insert/delete/select/delete tuples in tables
 - Query one or more tables – discussed next!

Tables in SQL

Product

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

A relation or table is a multiset of tuples having the attributes specified by the schema

Let's break this definition down

Tables in SQL

Product

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

A multiset is an unordered list (or: a set with multiple duplicate instances allowed)

List: [1, 1, 2, 3]

Set: {1, 2, 3}

Multiset: {1, 1, 2, 3}

i.e. no *next()*, etc. methods!

Tables in SQL

Product

PName	Price	
Gizmo	\$19.99	
Powergizmo	\$29.99	
SingleTouch	\$149.99	
MultiTouch	\$203.99	

An attribute (or column) is a typed data entry present in each tuple in the relation

Attributes must have an **atomic** type in standard SQL, i.e. not a list, set, etc.

Tables in SQL

Product

PName	Price	Manufacturer
Gizmo	\$19.99	GizmoWorks
Powergizmo	\$29.99	GizmoWorks
SingleTouch	\$149.99	Canon

Also referred to sometimes as a **record**

A tuple or row is a single entry in the table having the attributes specified by the schema

Tables in SQL

Product

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

The number of tuples is the cardinality of the relation

The number of attributes is the Degree of the relation

Tables in SQL

Product

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

A relation or table is a multiset of tuples having the attributes specified by the schema

Let's break this definition down

Data Types in SQL

- Atomic types:
 - Characters: CHAR(20), VARCHAR(50)
 - Numbers: INT, BIGINT, SMALLINT, FLOAT
 - Others: MONEY, DATETIME, ...
- Every attribute must have an atomic type
 - 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

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

Key constraints

A key is a minimal subset of attributes that acts as a unique identifier for tuples in a relation

- A key is an implicit constraint on which tuples can be in the relation
 - i.e. if two tuples agree on the values of the key, then they must be the same tuple!

Students(sid:string, name:string, gpa: float)

1. Which would you select as a key?
2. Is a key always guaranteed to exist?
3. Can we have more than one key?

NULL and NOT NULL

- To say “don’t know the value” we use **NULL**
 - NULL has (sometimes painful) semantics, more detail later

Students(sid:string, name:string, gpa: float)

sid	name	gpa
123	Bob	3.9
143	Jim	NULL

Say, Jim just enrolled in his first class.

In SQL, we may constrain a column to be NOT NULL, e.g., “name” in this table

General Constraints

- We can actually specify arbitrary assertions
 - E.g. *“There cannot be 25 people in the DB class”*
- In practice, we don’t specify many such constraints. Why?
 - Performance!

Summary of Schema Information

- Schema and Constraints are how databases understand the semantics (meaning) of data
- They are also useful for optimization
- SQL supports general constraints:
 - Keys and foreign keys are most important

The End

Question ?