SQL

Gary KL Tam

Department of Computer Science
Swansea University

What is SQL?

- SQL stands for Structured Query Language.

- Middleware
  - It is a database programming language that acts a means of communication between two other programs such as a website and a database.

- Fast
  - SQL can perform many of its functions in a fraction of a second.
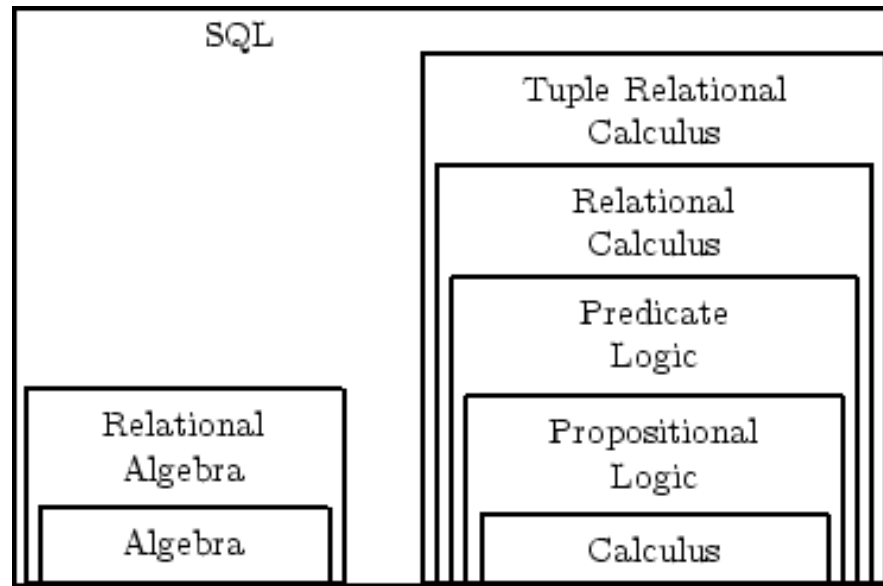
A little history of SQL

- Edgar F. Codd (IBM), "A Relational Model of Data for Large Shared Data Banks," 1970.
  - defined the relational database model
- First proposal: SEQUEL (1974, IBM Research).
  - SEQUEL was already taken, so it became SQL - Structured Query Language
- First Implementation:
  - 1978, IBM: First testing
  - 1981, IBM: SQL/DS
  - 1983, IBM: DB2
- Other vendors followed:
  - Oracle 1979
  - Sybase, Ingres etc

http://jobsearchtech.about.com/od/skills/p/SQL_Skills.htm

SQL

SQL standard

- Standardization was crucial for its diffusion.
- ANSI (1986) / ISO (1987) standard

- Note: two uses of the word "standard"
  - a technical specification (as in ANSI, ISO, etc.)
  - "what everybody uses"

- Most relational DBMSs support basic SQL and add on proprietary extensions.

- Arguably, the SQL standard, although well-defined, is not a true standard anymore - failure by DBMS vendors to do compliance testing.

Diagram showing nested boxes. Outer box labeled "SQL" contains two parts. Left side: "Relational Algebra" containing "Algebra". Right side: nested boxes "Tuple Relational Calculus" > "Relational Calculus" > "Predicate Logic" > "Propositional Logic" > "Calculus".

- **Relational algebra** provides the required base for computing queries of SQL
- **Tuple relational calculus (TRC)**, to a large degree, underlines the appearance of SQL
- Relational algebra is a **procedural** way for stating queries—concerned with "*how*"
- Relational calculus employs **declarative** expressions—concerned with "*what*"
- A relational query language is **relationally complete** if it can express the queries of the relational algebra
- TRC is **equivalent** to relational algebra in its **expressive power**

SQL vendors

- <u>Oracle</u>: The #1 commercial relational database (48.8% market share in 2011)

- <u>MySQL</u>: The world's most popular open source relational database

  - Open source = Free

    LAMP

- Others include

  - Sybase, and Microsoft (SQL server)

- Well-known websites
  - Autotrader
  - Ebay
  - Amazon

- The types of people that may use SQL include
  - Applications Developers,
  - Programmers, and
  - Database Administrators (look after large DBs)

- 1. **Portable**

  - …Although every database vendor implements their own extensions, with every new version the vendors are moving their databases to be in line with the ANSI/ISO SQL standard.

  - …If you learn SQL, you'll be **prepared to move from one database to another**.

By Jeremiah Peschka

http://www.brentozar.com/archive/2011/06/nine-reasons-developers-should-learn-sql/

- 2. **Never change**
  - …Vendors have implemented their own extensions that eventually make it into the SQL standard, but the **core of SQL doesn't change**…

- 3. **An easy place for performance gain**
  - …to make **a few simple changes** (add an index, change a query slightly) and see **tremendous performance improvements**…. Having spent a considerable portion of my career as an application developer staring at a profiler, I can attest to this….

By Jeremiah Peschka

http://www.brentozar.com/archive/2011/06/nine-reasons-developers-should-learn-sql/

- 4. **Make you a better developer**
  - …There are different ways of thinking about …problems… solutions. Learning SQL will teach you to think in **sets** rather than iteratively.

- 5. **Improve communication in team**
  - …Learning SQL will give you a leg up when you're **communicating** your goals to the DBA team….

    By Jeremiah Peschka

  - Gary: remember the admission director job at a University?

http://www.brentozar.com/archive/2011/06/nine-reasons-developers-should-learn-sql/

- 6. **Job security**
  - … If you can work with the **database team** and the **development team**, you're now a **valuable asset** that both teams depend on for success…

- 7. **Not really that hard**
  - …SQL is a **different way of thinking…** There are only a handful of commands, operators, and data types.….

By Jeremiah Peschka

http://www.brentozar.com/archive/2011/06/nine-reasons-developers-should-learn-sql/

- 8. **Know when it is not appropriate to do something in the database**
  - Once you understand SQL, you'll have a much better grasp on the **limitations of an RDBMS**. You'll know which portions of an application can safely live in a database and which will need to be moved further up the stack to a <u>different layer</u>.

- 9. **Simply troubleshooting**
  - …understanding SQL makes it easy **to locate the problem** in one of many different layers of your application.

Additional comments

- Any decent IT or computer science school should teach **relational theory and SQL**. As it has been stated, there is no reason, other than laziness, why a developer should'nt know already basic-intermediate SQL. IMHO, if you don't go through similar process in your studies or career, well, you **must not call yourself a developer, programmer, software engineer** or any other title, you're just **a code monkey**.

- Ironically, **laziness** is one reason to learn **SQL**. It solves so many problems **beautifully and simply**.

- Learning SQL will teach you to think in **sets** rather than iteratively. ...". I've found that **developers have no idea about set theory**. Their knee-jerk reaction to looping is to **create a cursor** rather than looking at alternative, and more likely better, solutions. Where I'm currently working there are dozens of cursors in the stored procedures created by programmers. By comparison in 10 years I've created one cursor.

- They had no clue at that time of the importance that SQL plays when dealing with RDBMS like SQL Server or Oracle. At that time the average clients database was no bigger than a few hundred MB ... and now and clients have databases measuring **hundreds of gigabytes**....

http://www.brentozar.com/archive/2011/06/nine-reasons-developers-should-learn-sql/