SQL - Structured Query Language

- Version of Relational Calculus (describe what results are desired)
- Developed by IBM Research in mid 70's

* Create Tables *

```
CREATE TABLE suppliers
    (sno VARCHAR(5) NOT NULL PRIMARY KEY,
     name VARCHAR(16),
     city VARCHAR(16),
     code INT)
```

* Delete Tables *

```
DROP TABLE suppliers;
```

* SELECT Statement *

```
SELECT * FROM suppliers;
```

You can add filters, such as

```
SELECT * FROM suppliers WHERE city = "Rome";
```
**INSERT Statements**

```sql
INSERT INTO supplier (sno, city, name, co) VALUES ('123', 'Charleston', 'Walmart', 2)
```

**Update Statements**

```sql
UPDATE supplier SET city = 'Mount Pleasant' where sno = '123'
```

**Delete Statements**

```sql
DELETE FROM supplier WHERE sno = '123'
```

---

Websites that demonstrate SQL:
- [sqlzoo.net](http://sqlzoo.net)
- [www.w3schools.com/sql/sql_trysql.asp](http://www.w3schools.com/sql/sql_trysql.asp)
Map - Reduce

We've seen the type of data that works well in relational databases. What about large unstructured data, such as the web? Google had to answer the question. Their answer was map-reduce (They've since moved to other technology for some applications).

Map-reduce:

* Programming model for processing and generating large datasets.
* Programs written in this style are automatically parallelized.
* Runs on large cluster of commodity machines and is highly scalable.

Made to process large amounts of data, such as crawled documents, inverted indices, web request logs, representations of graph structure of web documents, etc.
* Computations are conceptually straightforward (often); however, input data is usually large.

**Execution overview**

1. User Program
   - (1) fork
   - fork (1)
2. Worker
   - (2) assign map
3. Master
   - (3) read
4. Worker
   - (4) read
5. Worker
   - (5) read
6. Worker
   - (6) write
   - Output file 0
   - Output file 1

Input Files

Map phase
Intermediate files
Reduce Phase
Output files
Example Programs

* Grep

* Sort

* Indexing System
  * Data is more than 20 terabytes in 2004
  * Produces data structures needed during search (Google)

* Counting # of occurrences of words
  * Key
  * Output "word", 1 ∈ mapper