Databases: Past, Present, and Future
 (June 13, 2003)

What is a database and why do we have them?

In the simplest definition of the word, a database is a storage container for data. A database system is a storage container for data as well as a mechanism for accessing that data.
There are several kinds of databases we can think of - you can think of a spreadsheet as a database, a mail server, a word document, a relational server based database, desktop database such as Microsoft Access. Even a book is a database. The World Wide Web (WWW) you can think of as an intricate network of databases.

Databases have existed since the beginning of civilization and in fact define civilization, but were initially not electronic in nature. When man needed to store knowledge or keep track of information, they wrote them down, cataloged them using paper indices. So the book was the very first kind of database. These were not electronic databases, but nevertheless served the same purpose. They were used to track ledger accounts, scientific knowledge, and historical fact. Today when we think of databases, we think of electronic databases - not these hard copy items that defined civilization so many thousands of years ago.

Databases serve a couple of purposes.

- They compensate for the fact that we have terrible memories - forget something - just look it up
- They compensate for the fact that we can analyze only small pieces of information at a time -- because they group data for us, aggregate it much faster than any human could.
- They allow us to share facts and eventually knowledge - for example with Credit databases -- such as TRW all creditors can lookup and arrive at a credit risk for you - they know right away if you are likely to default on a loan

The Birth of the Relational Database

When people speak of true databases -- they often refer to more structured electronic databases such as relational databases, object databases, OLAP databases, and spatial databases. These kinds of databases can trace their roots back to the 1970 when E.F. Codd from IBM introduced the idea of relational model of databases in his paper

entitled "A Relational Model of Data for Large Shared Banks". Before that time most business databases were based on the network model or simple flat file structure.
The relational model was based on mathematical set theory. It served a couple of purposes

- It abstracted the representation of data from its physical storage and strived to manipulate data using this abstract model.
- It tried to minimize on redundancy of data by breaking data into distinct non-duplicating sets that could then be related an infinite number of ways to produce an infinite number of representations
- It increased consistency of data - e.g. if you change a name of a customer -- it would change in all reports you did about that customer - because that piece was maintained in only one location - but generated numerous views or representations of the data.

Later a language called SQL (also developed by IBM) was created to generate reports and update data in this new relational model.
IBM's System-R database was born from this work, but was largely ignored by IBM, and then very shortly Oracle came out with their commercial database based on E.F Codd's relational theory and research from System-R and research in Berkely Ingres. The Berkeley Ingres research project was also started at this time and strived to extend the relational model to deal with more complex data. Many object and object-relational databases can trace their roots back to Ingres. Oracle beat IBM to market with its relational database and made billions.
Other relational databases began to sprout from these early pioneers - Informix, Sybase, and Berkeley Ingres project later gave birth to the Berkeley Postgres which strived to add more Object-Oriented features to the relational model (both projects were headed by Michael Stonebraker) which then transformed into PostgreSQL. System-R gave birth to DB2.

Catalog of Various kinds of databases

The relational model and SQL gave birth to other models which we will briefly summarize in the next couple of sections. The main ones that come to mind are Object databases, Object-Relational, Spatial Database, OLAP databases and very recently XML databases.

Object Databases

The object database model followed shortly after the relational database model. This model was developed to compensate for the deficiencies of the relational database model -- namely to construct and query complex datastructures without having to denature them into a relational 2-dimensional structure. Object databases use a language very similar to SQL called OSQL (object sql). This bares a striking similarity to SQL e.g.
Below is an example of an OSQL statement that lists all orders for the Smith Company

```
SELECT co FROM co in customerorders WHERE name( customer(co)) = 'Smith
Company'
```

The object database proved to be too complicated for most purposes and it was also (at least initially) much slower than relational databases. As a result - there are very few pure object databases in this world. Most databases provide a compromise between the relational model and the object model. Examples of object-relational databases are Oracle and Postgres. These allow you to define complex datatypes and store them in tables but still tout the standard relational primitive types.

Spatial Databases

Spatial databases were developed to correlate data in space. They provided answers to questions such as how much money have we spent within a 20 mile radius from this specific location? How far has waste product extended from the spill location? How many miles away is the closest hospital to this house?
Most spatial databases don't stand on their own, but instead are just an extension to relational databases. They use a dialect of SQL called Spatial Features Structured Query Language (SFSQL) - which simply adds spatial functions to SQL - such as distance, touches, centroid, inside, area, extent.
In fact most spatial databases store spatial data in relational databases, but in specialized fields used to hold spatial data. Examples of spatial databases are Oracle Spatial (which sits on top of Oracle) , ESRI Arc SDE (which can sit on top of a Microsoft SQL Server or Oracle database), PostGIS (sits on top of PostgreSQL), DB2 Spatial Extender which adds spatial functionality to IBM DB2 databases, and even MySQL is providing limited functionality for Spatial data in its upcoming 4.1 version.

Below is an example of what a spatial query looks like. This is using the Postgis dialect of SFSQL, but the syntax is pretty similar in other spatial databases.

```
SELECT building.gid, building.name, policestation.name ,
min(distance(building.the_geom, policestation.the_geom)) As distance
FROM building, policestation, (SELECT b.gid, min(distance(b.the_geom, p.the_geom))
```