1 Database History Observations

- Stonebraker argues that a lot of the issues relevant a long time ago are still relevant today [11].
- The “SQL vs. NoSQL” debate of the 2010s is reminiscent of the “Relational vs. CODASYL” debate from the 1970s [8].

2 The 1960s

- **First DBMS - Integrated Data System (IDS) [1, 6]**
  - Developed internally at GE in the early 1960s by Charles Bachman.
  - GE sold their computing division to Honeywell in 1969.
  - Based on the **Network Data Model** that supported **Tuple-at-a-time** query execution [3].

- **CODASYL [12]**
  - Proponents of COBOL and the network data model people got together and proposed a standard for how programs will access a database.
  - The Network Data Model made it difficult to write complex queries due to the sets that maintained relationships. It also made these early systems more susceptible to corruption [11].

- **IBM Information Management System (IMS) [7]**
  - Early database system developed to keep track of purchase orders for Apollo moon mission.
  - Based on the **Hierarchical Data Model** that organized collections of data with parent/child relationships.
  - Programmer-defined physical storage format (e.g., hash table vs. tree) and tuple-at-a-time query execution.

3 The 1970s

- **Relational Model [5]**
  - Ted Codd was a mathematician working at IBM Research who saw developers constantly changing their codebase whenever the database’s schema changed.
  - Codd created the relational model abstraction to avoid this maintenance based on three key ideas:
    1. Store database in simple data structures
    2. Access data through high-level language
    3. Physical storage left up to implementation

- Early implementations of relational DBMSs:
– IBM Research: **System R**
– U.C. Berkeley (Mike Stonebraker): **INGRES**
– Relational Software, Inc. (Larry Ellison): **Oracle**

4 **The 1980s**

• The relational model wins the database marketplace over CODASYL.
  – IBM released their first relational DBMS (**DB2**) in 1983.
  – System R’s “SEQUEL” declarative query language becomes the standard (later renamed to “SQL”).
  – Many new enterprise DBMSs are invented (**Informix, Sybase, TeraData**) but Oracle wins marketplace.
  – Stonebraker leaves INGRES, returns to Berkeley and starts the Postgres project.
• **Object-Oriented Databases** [13]
  – Argued that how people wrote code and how data is stored in a database is different
  – Avoid “relational object impedance mismatch” by tightly coupling objects and database.
  – Few of these original DBMSs from the 1980s still exist today. These systems performed poorly when executing complex queries. There was also no standard API or programming language.

5 **The 1990s**

• No major advancements in database systems or application workloads.
• Microsoft forks **Sybase** and creates **SQL Server** for Windows NT.
• **MySQL** is written as a replacement for **mSQL**.
• Illustra (the commercial version of **Postgres**) gets bought by **Informix**. Graduate students at Berkeley take the original academic **Postgres** code and adds support for SQL.
• **SQLite** started in early 2000.

6 **The 2000s**

• **Data Warehouses**
  – Distributed / Shared-Nothing
  – Relational / SQL
  – Usually closed-source
  – Significant performance benefits from using **Decomposition Storage Model** (i.e., columnar storage).
• **NoSQL Systems** [4]
  – Focus on high-availability and high-scalability
  – Schema-less
  – Non-relational data models
  – No ACID transactions
– Custom APIs instead of SQL
– Usually open-source

7 The 2010s

• **NewSQL Systems** [2, 9]
  – Strive to provide same scalability and performance for OLTP workloads as NoSQL DBMSs without giving up ACID.
  – Relational / SQL
  – Distributed
  – Usually closed-source

• **Hybrid Transactional-Analytical Processing (HTAP)** [10]
  – Execute fast OLTP like a NewSQL system while also executing complex OLAP queries like a data warehouse system.
  – Distributed / Shared-Nothing
  – Relational/SQL
  – Mixed open/closed-source

• **Cloud-based Database Systems**
  – First database-as-a-service (DBaaS) offerings were “containerized” versions of existing DBMSs (e.g., Amazon RDS).
  – There are new DBMSs that are designed from scratch explicitly for running in a cloud environment.

• **Specialized Database Systems**
  – Shared Disk (HDFS, EBS)
  – Embedded
  – Time-Series
  – Multi model
  – Blockchain
References


