SQL: The Esperanto Language of Data

SELECT * FROM Future;

Zoey Guan

March 20, 2025

AGENDA

1.The Data Crisis Before SQL

2.The Relational Revolution (1969-1974)

3.SQL's Birth & Commercialization (1979-1986)

4.Why SQL Became the Dominant Language

5.The Evolution and Future of SQL

6.Conclusion

The Crisis Before the Revolution

1. **Hierarchical Databases**
   Rigid tree structures required predefined paths.

2. **CODASYL Networks**
   Complex pointer hell. A PhD was needed to query data.
   Required manual pointer navigation

3. Programmers spent 70% of time on data navigation, not analysis (Bachman, 1973).

Edgar Codd & Relational Model (1969)

1

Mathematical Foundation

Relational Algebra = "Data as Math".

Hierarchical model → Relational tables.

2

Tables & Relationships

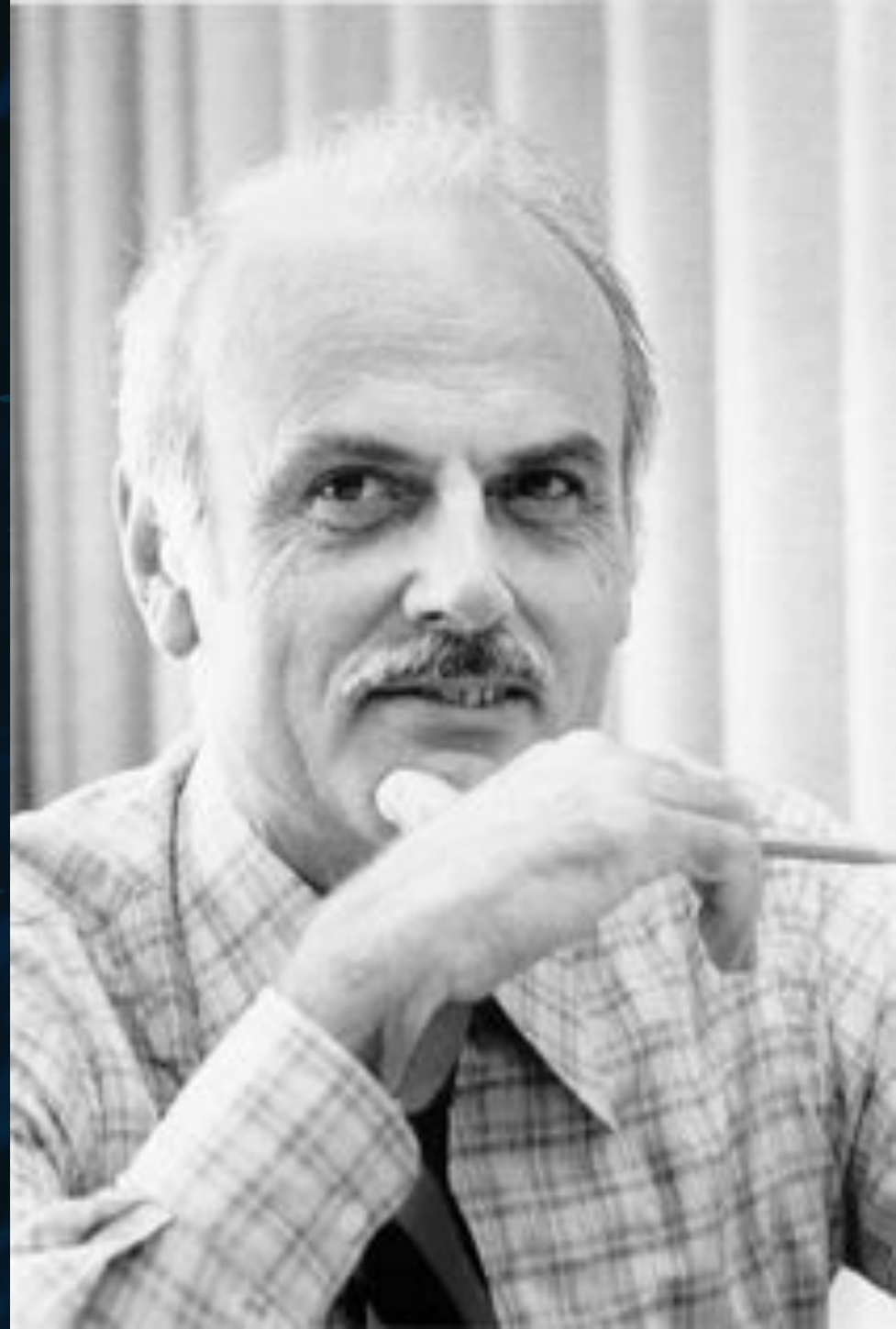
Replaced rigid pointers.

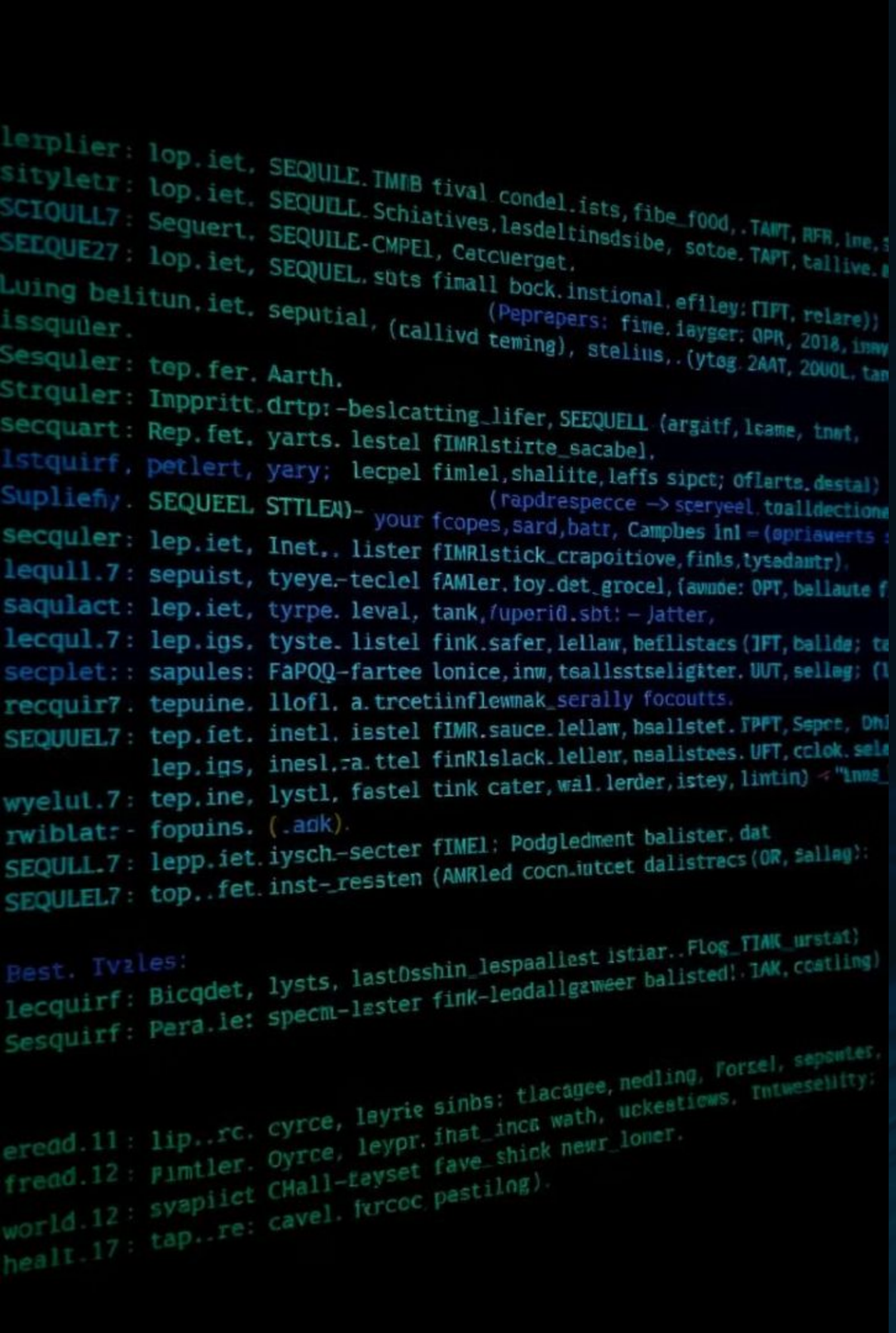
3

IBM's Rejection

IBM initially rejected his ideas.

Too radical for hardware-focused era.





SEQUEL & System R (1974)

IBM’s System R experiment introduced **SEQUEL** (Structured English Query Language).

SEQUEL vs. earlier programming: Declarative queries vs. procedural navigation.

```
SELECT * FROM Employees WHERE Salary > 5000;
```

Legal issue forced renaming to SQL.

“When in doubt, just remove vowels”

‘SEQUEL’ → ‘SQL’

Oracle's Commercialization of SQL (1979)

1. Larry Ellison launched Oracle V2 before IBM could commercialize SQL.
2. SQL became the go-to database language for businesses.
3. Adopted by banks, governments, ERP systems.

SQL Becomes a Global Standard (1986-1992)

1986

ANSI Standard

SQL was standardized in 1986.

1992

ISO Standard

SQL was truly global standardized in 1992.

SQL is the Rosetta Stone

> SQL is the Rosetta Stone — deciphering the myriad languages, from Excel cells to the endless cloud.

Why SQL Still Dominates Today ? - The Key Strengths of SQL

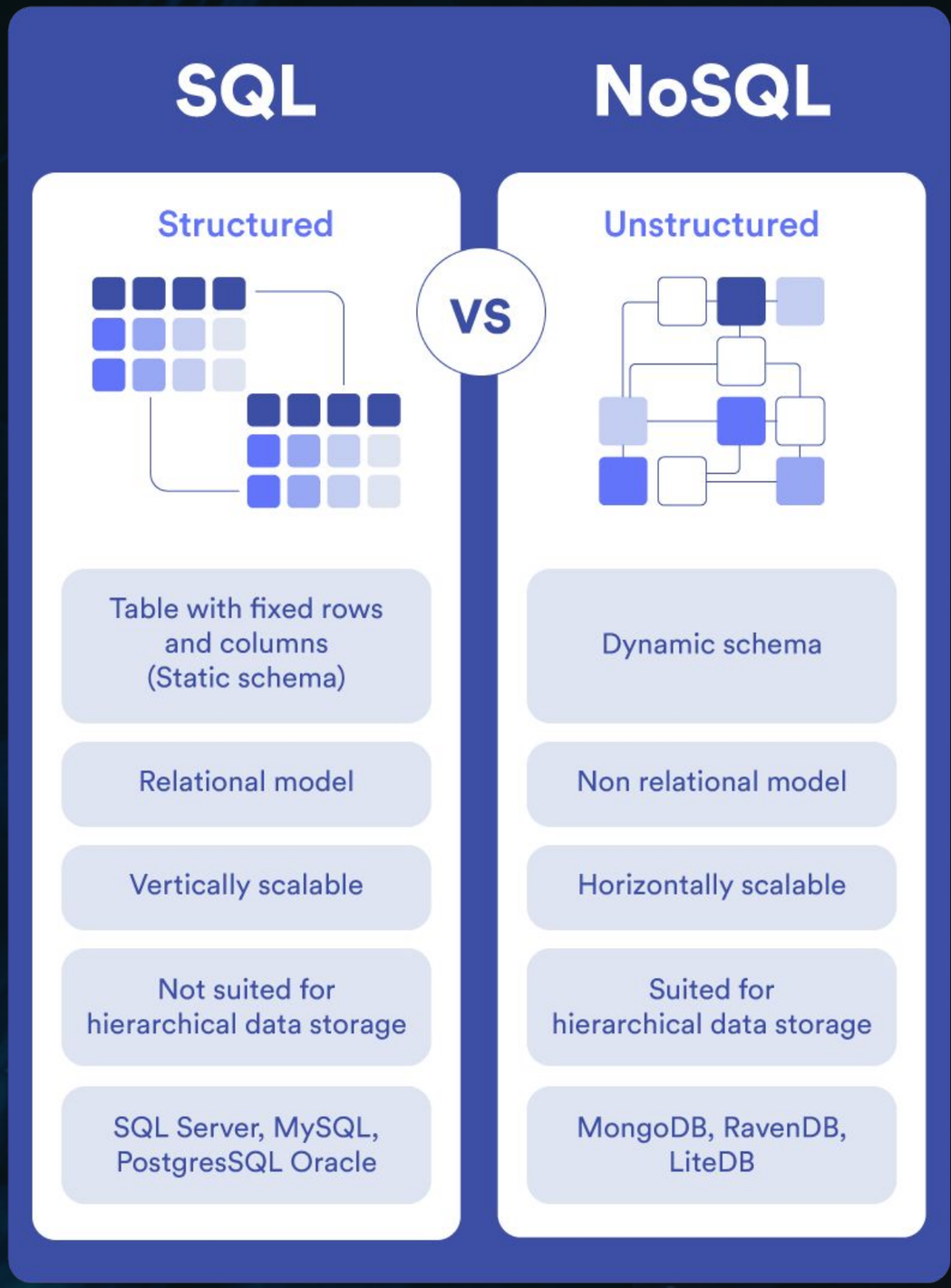
Standardization
- Global SQL standard (ANSI 1986, ISO 1992) ensures cross-platform consistency.

Structured Data Model
- Tables & schema-based organization → Enforces data integrity.

Ease of Use
- Intuitive syntax, close to natural language.

```
SELECT name FROM users;
```

Efficiency & Reliability
- Optimized for structured data & supports ACID transactions.



The ACID Guarantee

- **“ACID Transactions: The Foundation of Data Reliability”**

1. Atomicity
   "All or Nothing."

2. Consistency
   Always maintains data integrity.

3. Isolation
   Prevents conflicts in simultaneous transactions.

4. Durability
   Data is permanently stored.

The Future of SQL

-SQL in the Era of Big Data

1

Cloud-Native SQL

BigQuery, AWS Aurora, Snowflake handle petabytes of data.

2

AI Symbiosis

GPT-4 writes SQL with 92% accuracy.

Prediction: SQL will outlive Python.

The Future of SQL
-The SQL + NoSQL Hybrid Model

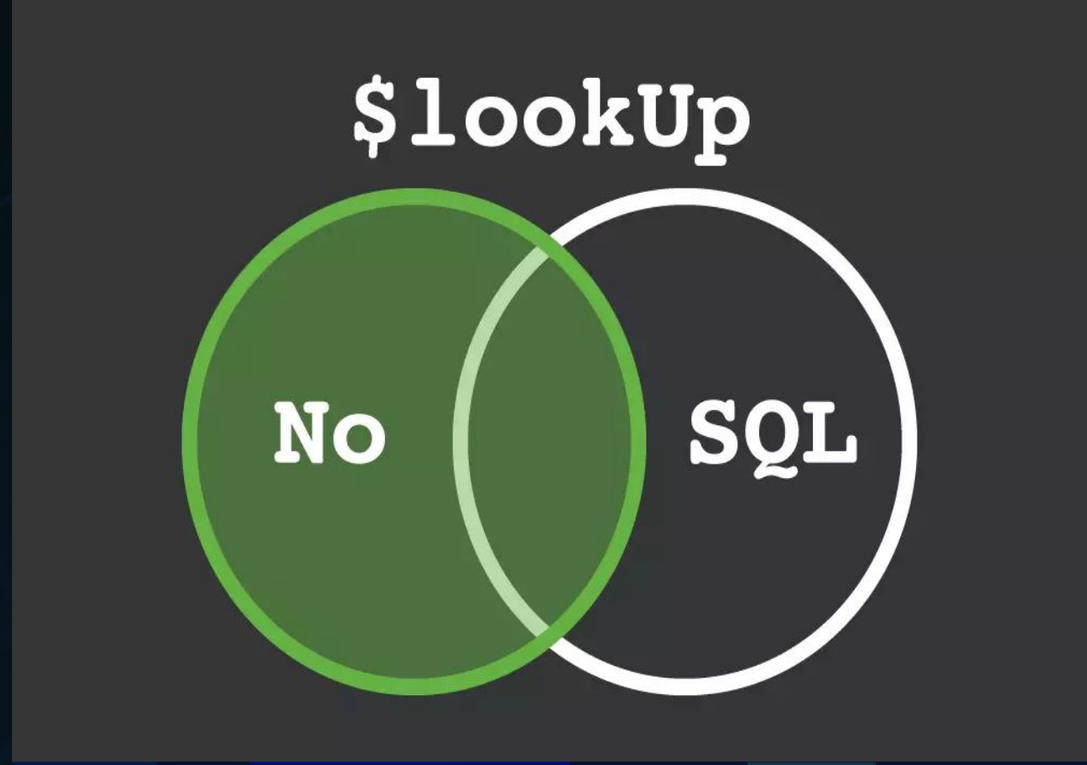
- SQL & NoSQL are converging.
- Example: MongoDB now supports SQL queries via SQL Query API.
- Enables flexibility of NoSQL + structure of SQL.

BEFORE MongoDB use the JSON:

```
db.employees.find({ "salary": { "$gt": 5000 } })
```

NOW MongoDB allows the SQL:

```
SELECT name, salary FROM employees WHERE salary > 5000;
```



Conclusion

DELETE FROM obsolete_tech WHERE vision < 1970;

THANK YOU!