

数据库发展史

汇报人：王俊



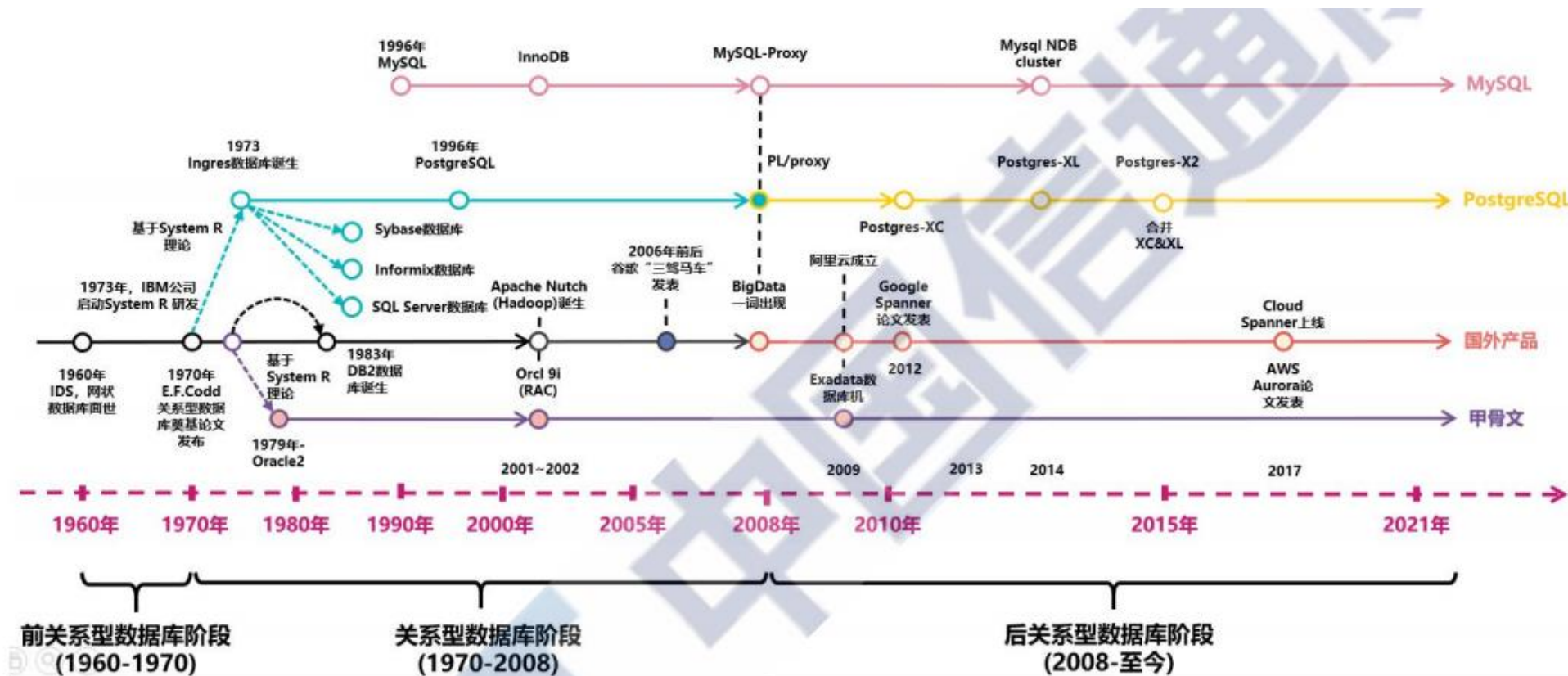
分布式存储与计算实验室

2022.08.29

数据库历史



分布式存储与计算实验室



1960s

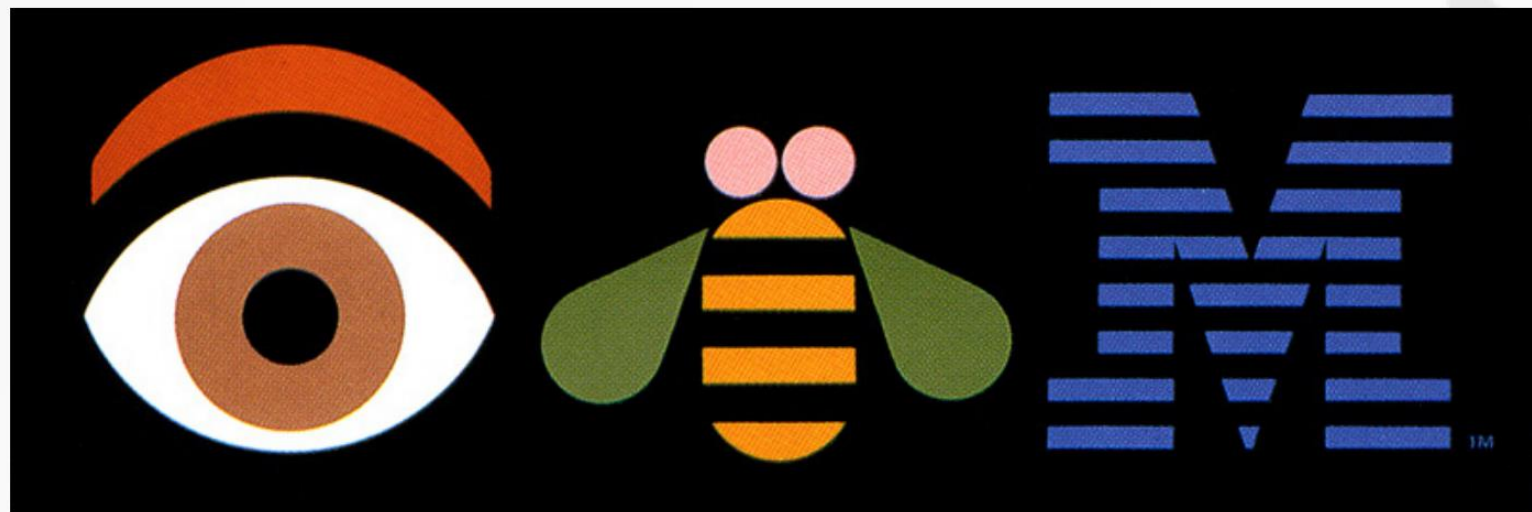


分布式存储与计算实验室

IBM IMS

世界上第一个数据库是为了追踪阿波罗登月任务的采购订单而开发的。

- 层次数据模型
- 存储格式由开发者定义
- Tuple-at-a-time queries



1960s



分布式存储与计算实验室

Schema

SUPPLIER

(sno, sname, scity, sstate)



PART

(pno, pname, psize, qty, price)

Instance

sno	sname	scity	sstate	parts
1001	Dirty Rick	New York	NY	
1002	Squirrels	Boston	MA	

pno	pname	psize	qty	price
999	Batteries	Large	10	\$100

pno	pname	psize	qty	price
999	Batteries	Large	14	\$99

层次数据模型

1970s



分布式存储与计算实验室

CODASYL

COBOL的开发者们在Charles Bachman领导下，一起开发了用COBOL语言访问数据库的标准API。

- 网状数据模型
- Tuple-at-a-time queries.

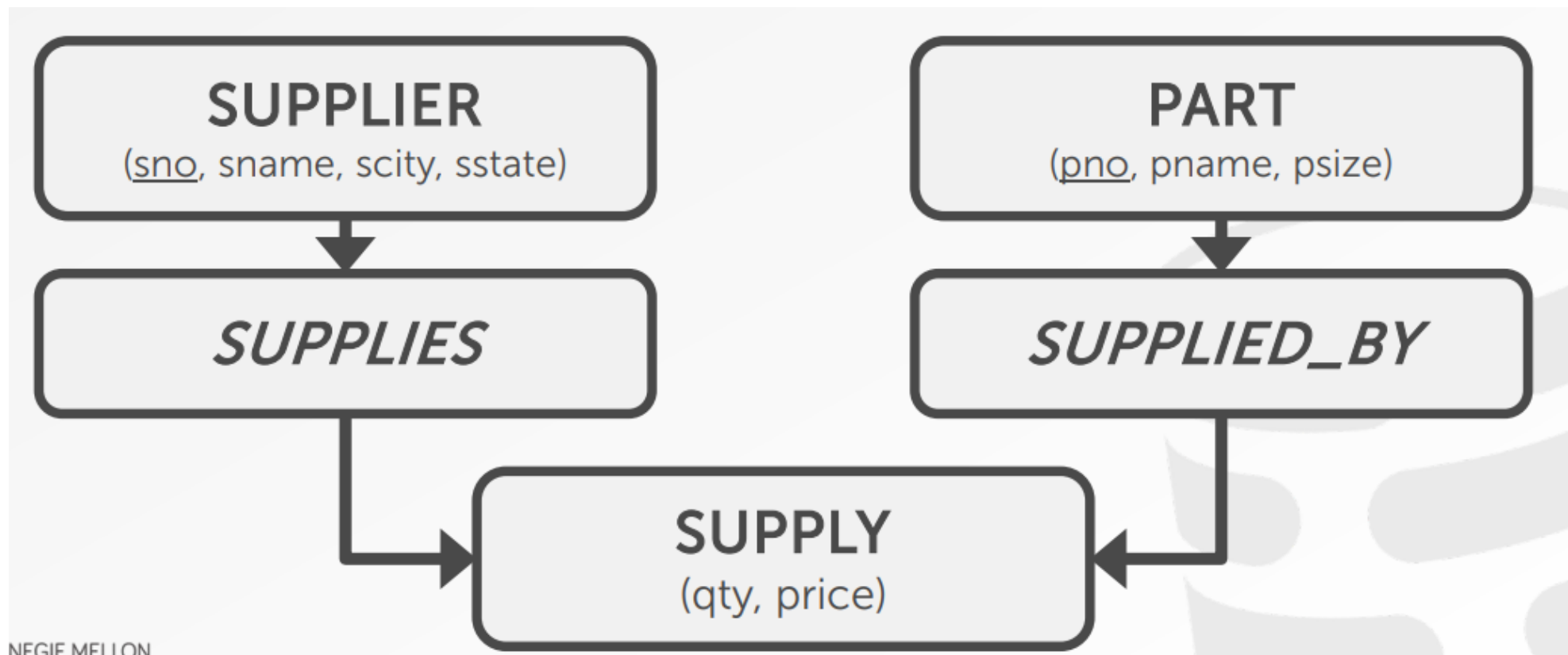


Charles Bachman

1970s



分布式存储与计算实验室



NEGIE MELLON

网状数据模型

1970s



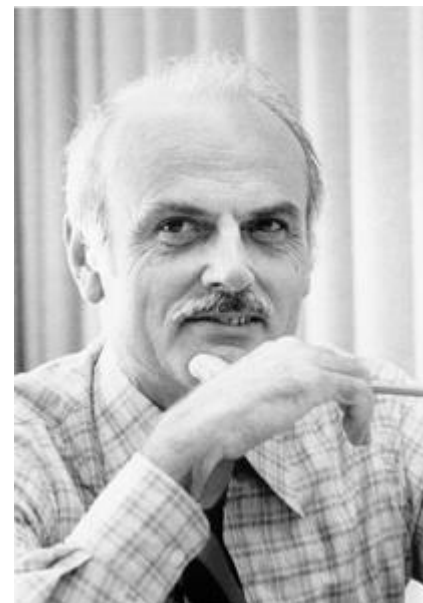
分布式存储与计算实验室

RELATIONAL MODEL

IBM研究院数学家Ted Codd发现，每次数据库的schema或layout发生变化时，开发人员都要花时间重写IMS和Codasy1程序。

提出将数据库抽象化以避免重写：

- 将数据存储在简单的数据结构中
- 使用高级语言访问数据
- 让数据库自己决定具体存储格式

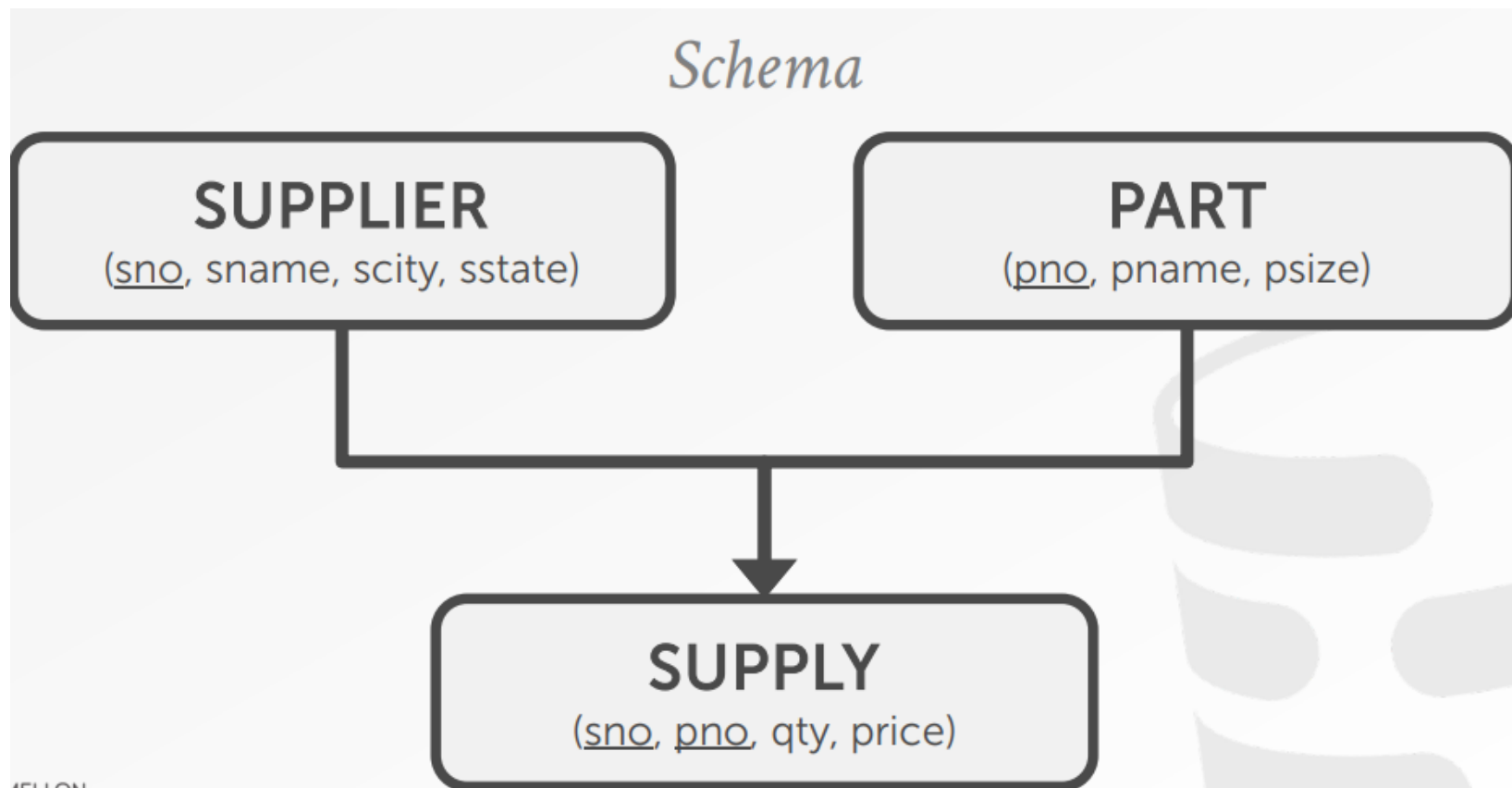


Ted Codd

1970s



分布式存储与计算实验室



关系数据模型

1980s



分布式存储与计算实验室

OBJECT-ORIENTED DATABASES

为了避免 “relational-object impedance mismatch”现象, OBJECT-ORIENTED DATABASES将对象和数据库紧密耦合。

这些20世纪80年代的DBMS大多已经不在, 但很多技术以其他形式存活了下来, 如JSON, XML。

1980s



分布式存储与计算实验室

Application Code

```
class Student {  
    int id;  
    String name;  
    String email;  
    String phone[];  
}
```



Student

```
{  
    "id": 1001,  
    "name": "M.O.P.",  
    "email": "ante@up.com",  
    "phone": [  
        "444-444-4444",  
        "555-555-5555"  
    ]  
}
```

OBJECT-ORIENTED MODEL

2000s



分布式存储与计算实验室

DATA WAREHOUSES

为了满足数据分析的需求，OLAP的数据库开始崛起。

- Distributed / Shared-Nothing
- Relational / SQL
- Usually closed-source

通常使用列式存储来加快查询速度。



2000s



分布式存储与计算实验室

NoSQL SYSTEMS

专注于高可用性和高扩展性。

- 没有Schema
- 非关系型数据模型
- 不支持事务
- 使用特定的API来代替SQL
- 通常是开源数据库
- 加快查询速度。



2010s



分布式存储与计算实验室

NewSQL

在OLTP的场景下，可以提供与NoSQL数据库相似的性能，同时提供事务支持。

- 关系型/SQL
- 分布式



HYBRID SYSTEMS

Hybrid Transactional-Analytical Processing

在OLTP场景下，提供与NewSQL相当的性能；而在OLAP场景下，可以像数据仓库一样进行复杂查询。

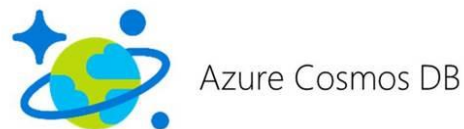
- Distributed / Shared-Nothing
- Relational / SQL
- Mixed open/closed-source



Multi-Model Database

一个数据库支持多个存储引擎，可以同时满足应用程序对于结构化、半结构化和非结构化数据的统一管理需求。

- 具有多种数据模型数据存储能力
- 提供同时查询多种数据模型数据的查询语言
- Single-engine\Muti-engine



- [What is a multi model database - YouTube](#)
- [《openGauss 数据库核心技术》](#)
- [《数据库发展研究报告（2021）》](#)
- [Course Information & History of Databases \[CMU Database Systems Spring 2017\]](#)