数据库系统概述

李文根/Wengen Li
Email: lwengen@tongji.edu.cn

先进数据与机器智能系统实验室
Advanced Data and Machine Intelligence Systems (ADMIS) Lab
https://admis-tongji.github.io

同济大学 计算机科学与技术学院

2026年03月

课程概要

同济大学 TONGJI UNIVERSITY

- **Part 0: Overview**
  - Ch1: Introduction
- **Part 1 Relational Languages**
  - Ch2: Relational model
  - Ch3&4: SQL
  - Ch5: Advanced SQL
- **Part 2 Database Design**
  - Ch6: Database design via E-R model
  - Ch7: Relational database design
- **Part 3 Application Design & Development**
  - Ch8: Complex data types
  - Ch9: Application development
- **Part 4 Big Data Analytics**
  - Ch10: Big data
  - Ch11: Data analytics
- **Part 5 Storage Management & Indexing**
  - Ch12: Physical storage systems
  - Ch13: Data storage structures
  - Ch14: Indexing
- **Part 6 Query Processing & Optimization**
  - Ch15: Query processing
  - Ch16: Query optimization
- **Part 7 Transaction Management**
  - Ch17: Transactions
  - Ch18: Concurrency control
  - Ch19: Recovery system
- **Part 8 Parallel & Distributed Database**
  - Ch20: Database system architecture
  - Ch21-23: Parallel & distributed storage, query processing & transaction processing
- **Advanced topics**
  - DB Platform: **OceanBase**, MongoDB, Neo4J
  - RAG, Multimodal retrieval, …

关系表

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

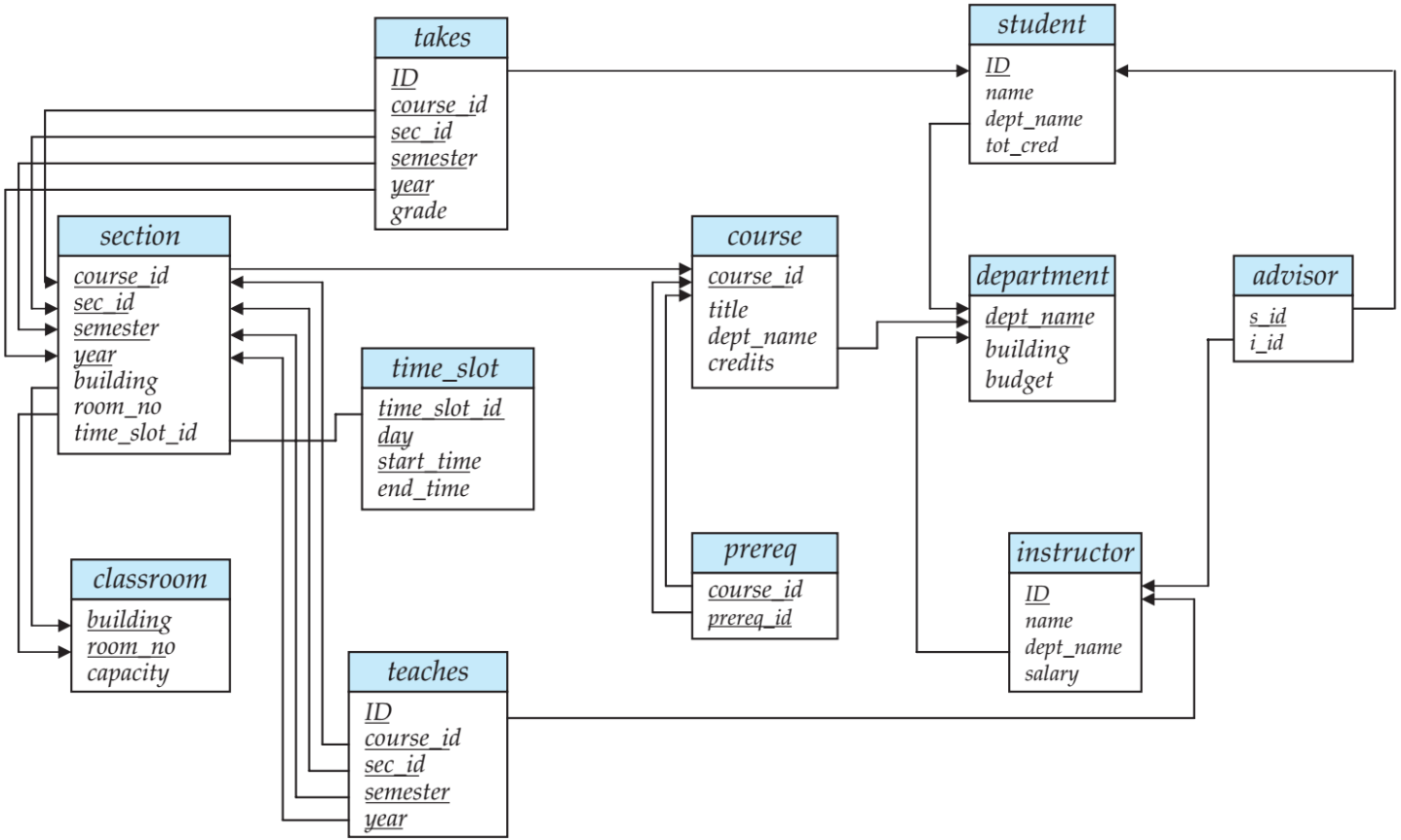
Instructor table

ID	*name*	*dept_name*	*tot_cred*
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

Student table

思考：在大学里，用于存储教学相关的关系表可能有哪些？这些表之间有什么关系？

University DB



- Database Design
- Storage & Indexing
- Query & SQL
- Transactions

目录

传统文件处理系统的缺点

- **数据的冗余和不一致性 （data redundancy and inconsistency)**
  - Multiple file formats (different programmers/languages/structures), duplication of information in different files
- **数据访问困难（difficulty in accessing data)**
  - Need to write a new program to carry out each new task
- **数据孤立（data isolation)**
  - Multiple files and formats
- **完整性问题（integrity problems)**
  - Integrity constraints（e.g., account balance > 0) become “buried” in program code rather than being stated explicitly
  - Difficult to add new constraints or change existing ones
- **原子性问题（atomicity problem)**
  - Failures may leave database in an inconsistent state with partial updates carried out
    - E.g., transfer of funds from one account to another should either complete or not happen at all
- **并发访问异常（concurrent-access anomaly)**
  - Concurrent access is needed for better performance but uncontrolled concurrent accesses can lead to inconsistencies
- **安全性问题（security problems)**
  - Difficult to provide user access to some, but not all, data

数据库（Database, DB）

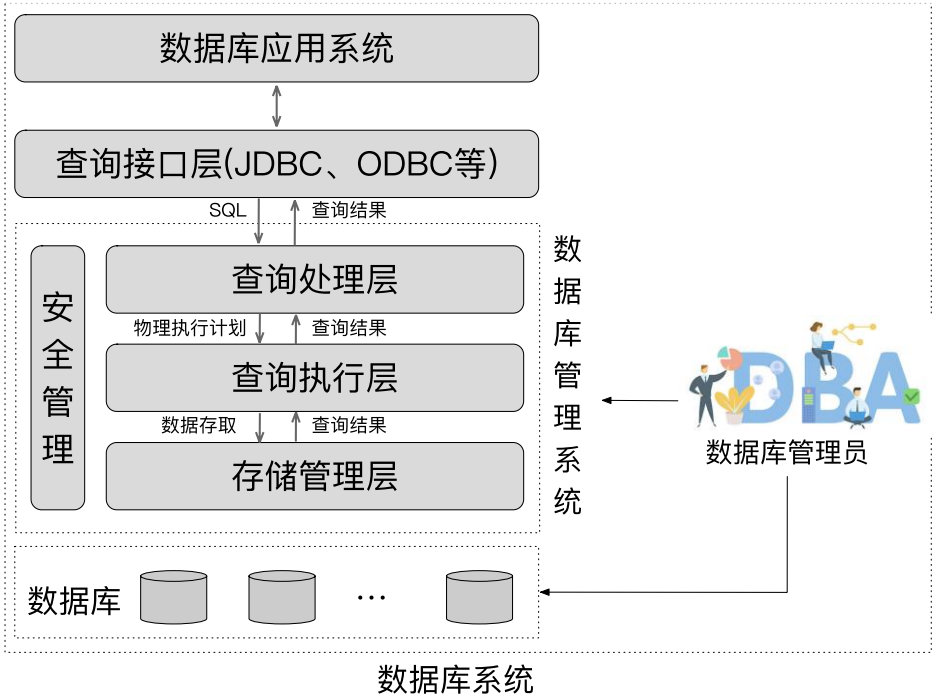
- **什么是数据库?**
  - 一组相互关联的、有组织的、可管理和可共享的数据集合

- **数据库的基本特征**
  - 数据按一定的数据模型组织、描述和储存
  - 支持数据的增、删、改、查
  - 支持并发查询处理

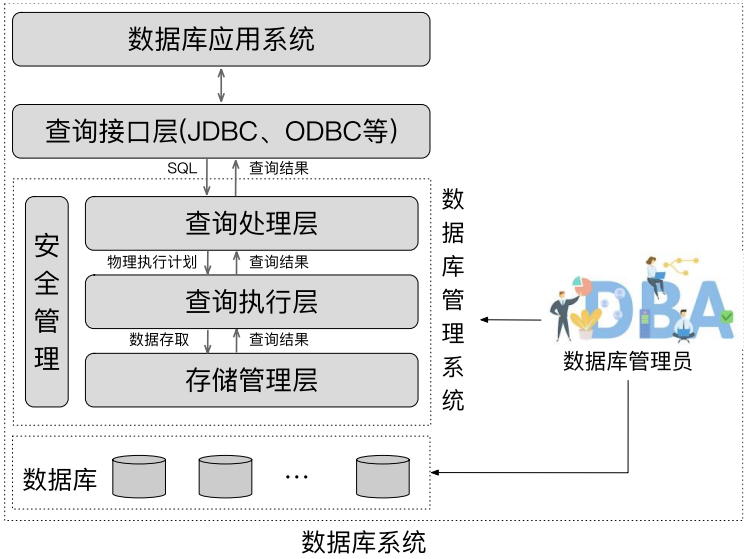
数据库管理系统(DB Management System, DBMS)

同济大学 TONGJI UNIVERSITY

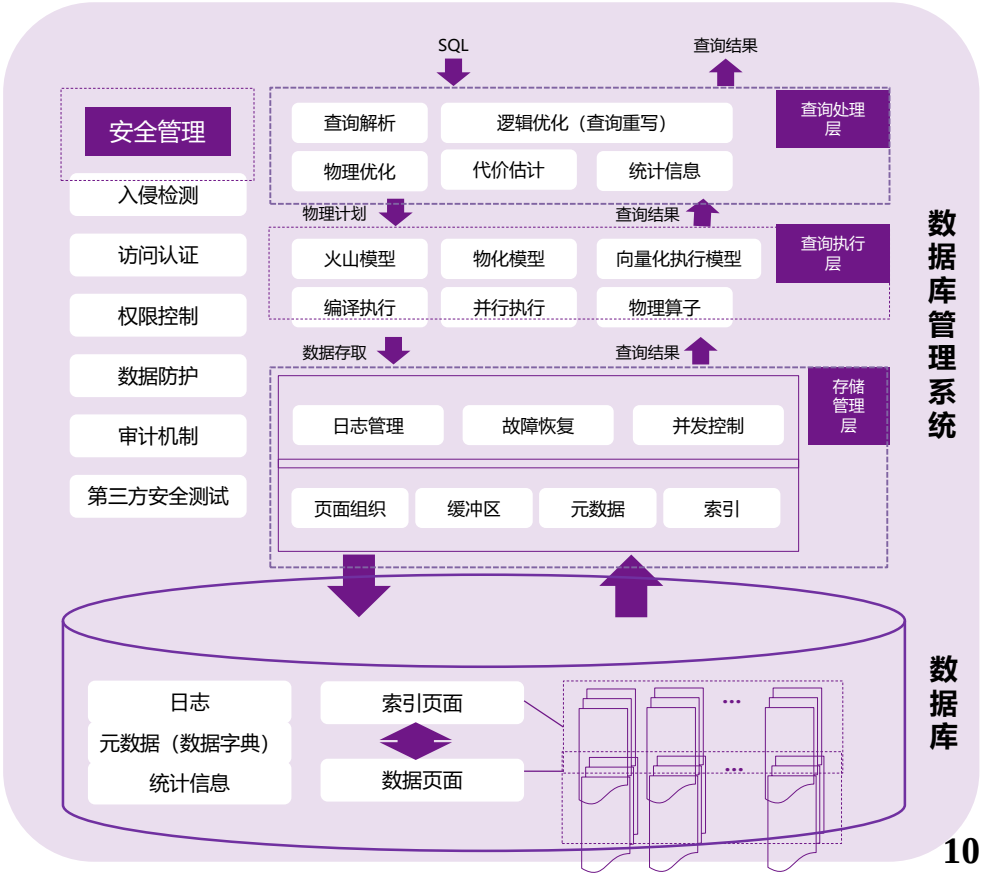
- **什么是数据库管理系统?**
  - **定义1**：用户（应用程序）与操作系统之间的数据库管理软件
  - **定义2**：一个管理数据的大型复杂基础软件系统
- **数据库管理系统的用途**
  - 优雅查询和数据抽象
  - 高效组织和存储数据
  - 正确一致的并发更新
  - 低时延高吞吐的查询
  - 并行高效的有序执行
  - 可用性和高可靠保证
  - 安全可信的统一控制
  - 方便易用的用户接口



数据库管理系统（DBMS）

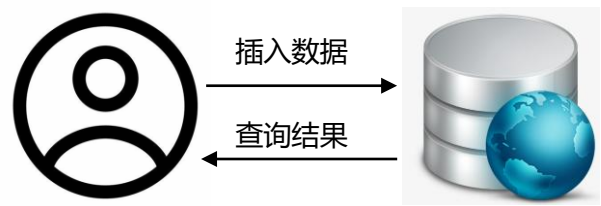


细化



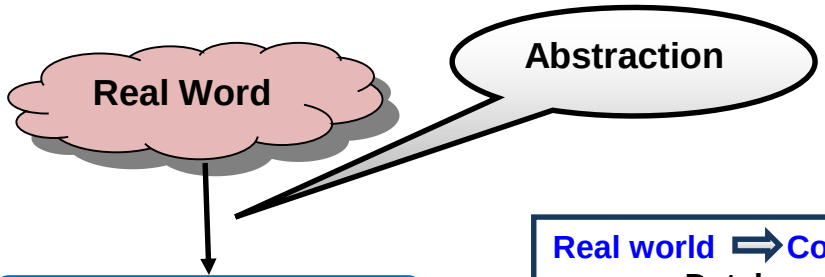
数据库的核心用途

- 数据库是数据基础设施
- 数据库是核心基础软件
- 数据库已赋能千行百业



目录

数据抽象



Information world

Machine world

Conceptual Model

Data model supported by DBMS

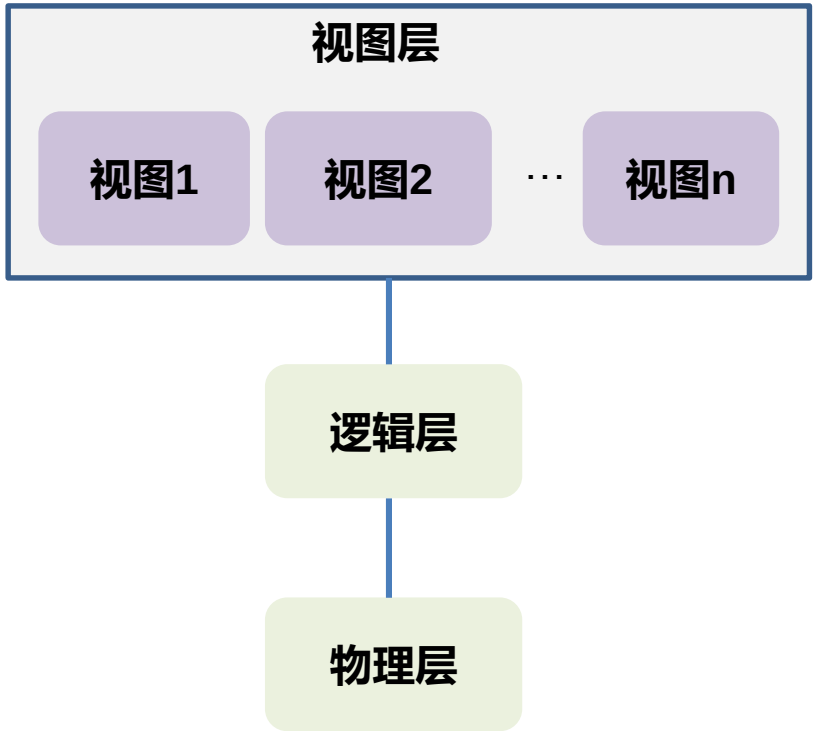
Real world ⇨ Conceptual model
Database Designers

Conceptual model ⇨ Logical model
Database Designers

Logical model ⇨ Physical model
DBMS

数据抽象的层次

- **视图层（View level）**
  - Users only need to access a part of the database. The view level simplifies the interaction with the system
  - Views can hide info (e.g., an employee's salary) for security purposes
- **逻辑层（Logical level）**
  - Describe what data stored in database, and the relationships among the data
- **物理层（Physical level）**
  - Describe how a record is stored on storage devices



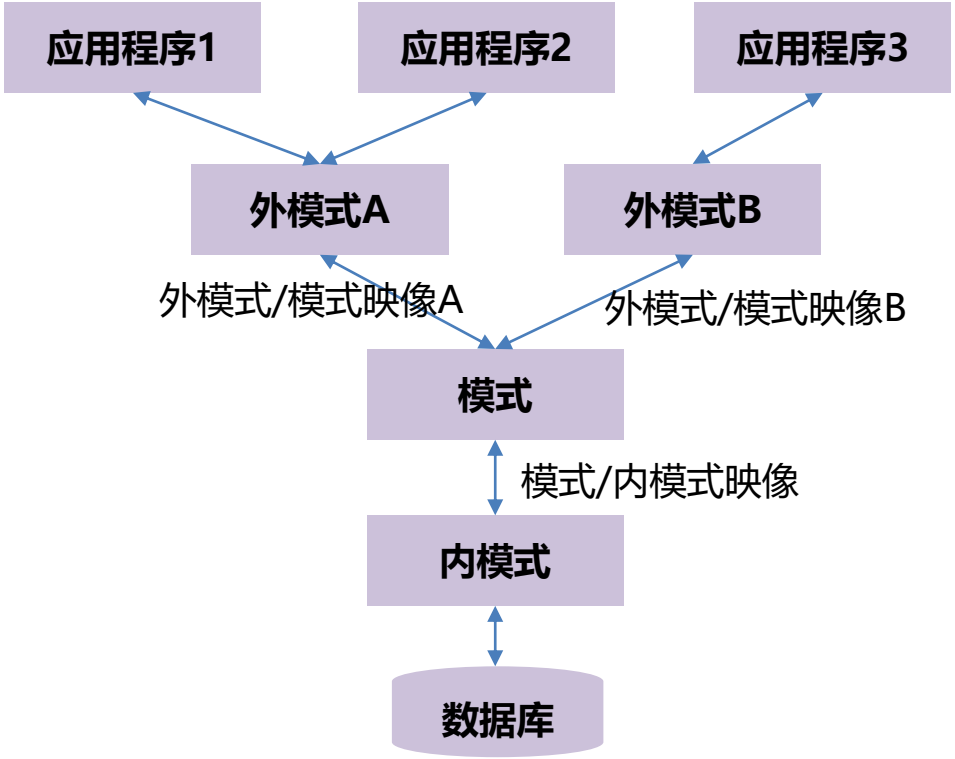
模式与实例（Schema and Instance）

- **模式(Schema):** 数据库的结构
  - 例如：the database contains the information about instructors and students, and the relationship between them
  - **三种模式:**
    - External schema(外模式/子模式): define at the app level
    - Logical schema(概念模式/逻辑模式/模式): design at the logical level
    - Physical schema(物理模式/内模式/存储模式): design at the physical level
- **实例(Instance)**
  - **特定时刻**存储在数据库中信息的集合称为数据库的一个实例

- 模式和实例之间的关系类似编程语言中**数据类型**和**变量**之间的关系

模式与实例（Schema and Instance）

- **物理数据独立性(Physical data independence)**
  - The ability to modify the physical schema without changing the logical schema
- **逻辑数据独立性(Logical data independence)**
  - The ability to modify the logical schema without changing the external schema



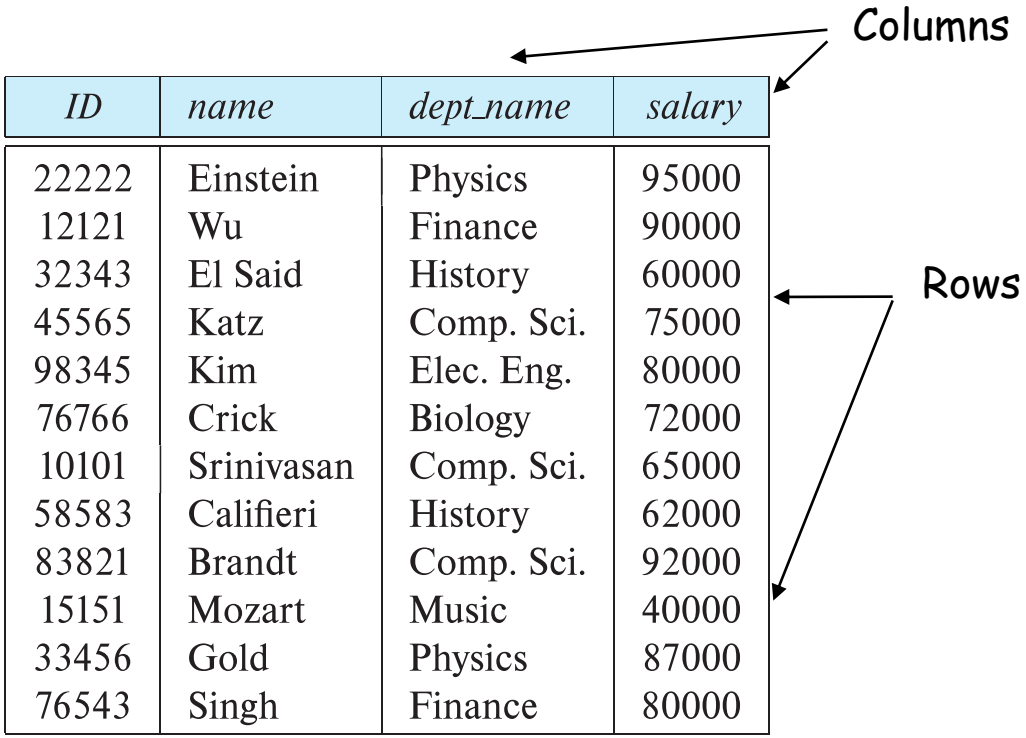
数据模型(Data Models)

同济大学 TONGJI UNIVERSITY

- **数据模型**：数据如何被组织、存储、处理以及关联起来的一套方法、规则和工具
  - **关系模型(Relational model)**
    - Record-based model, use tables as relations to represent data and their relationships
  - **实体-联系模型(Entity-Relationship data model)**
    - Mainly for database logical design
  - **半结构化数据模型(Semi-structured data model)**
    - JSON, XML, etc.
  - **基于对象的数据模型(Object-based data models)**
    - Extend the E-R model with the notions of encapsulation(封装), methods(方法), and object identity (对象标识)
  - **其他数据模型**
    - Network model (e.g., Honeywell IDS, 1963)
    - Hierarchical model (e.g., IBM IMS, 1968)

关系模型

- 关系模型中的**表格数据**



ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

关系模型

- 关系模型中的**数据关系**

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

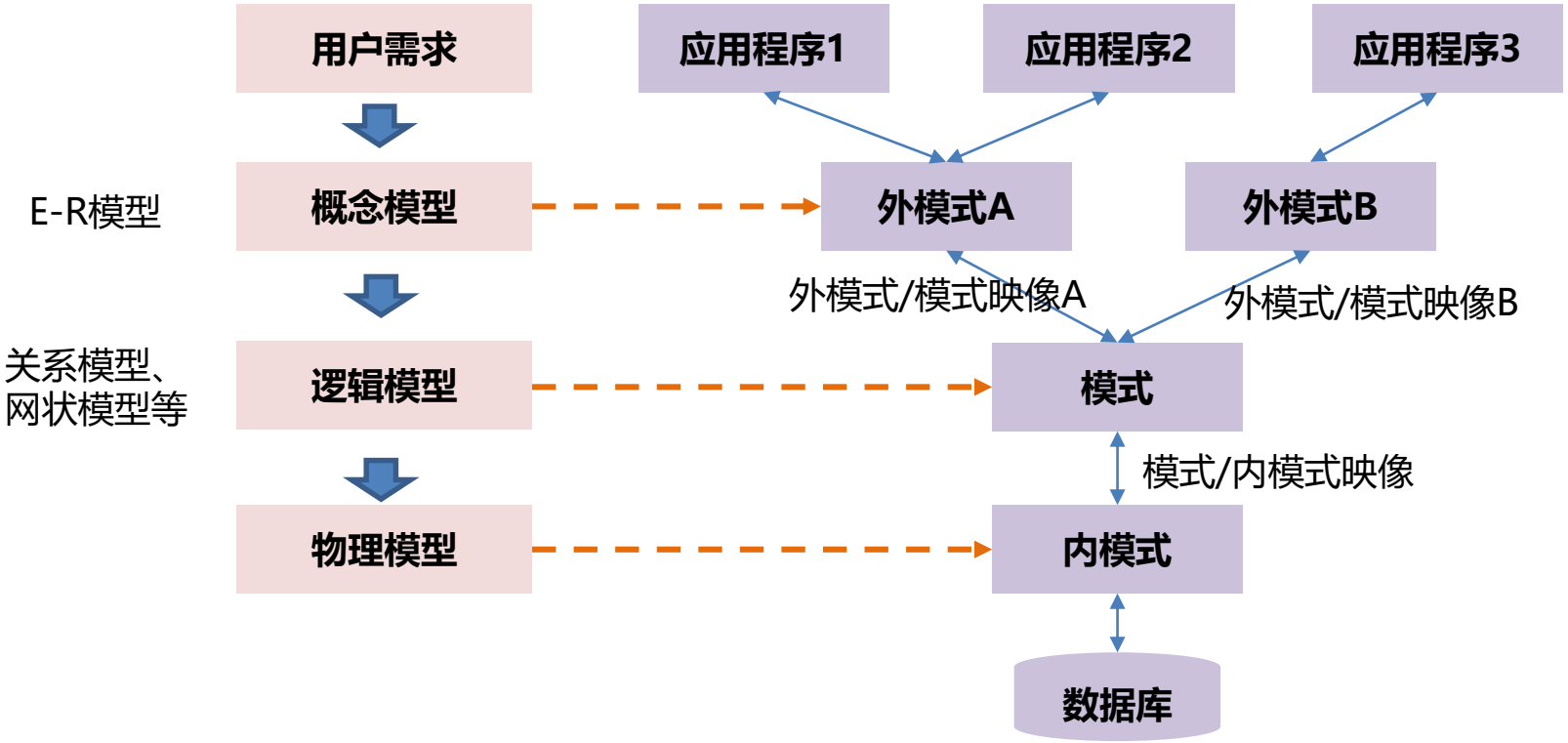
(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

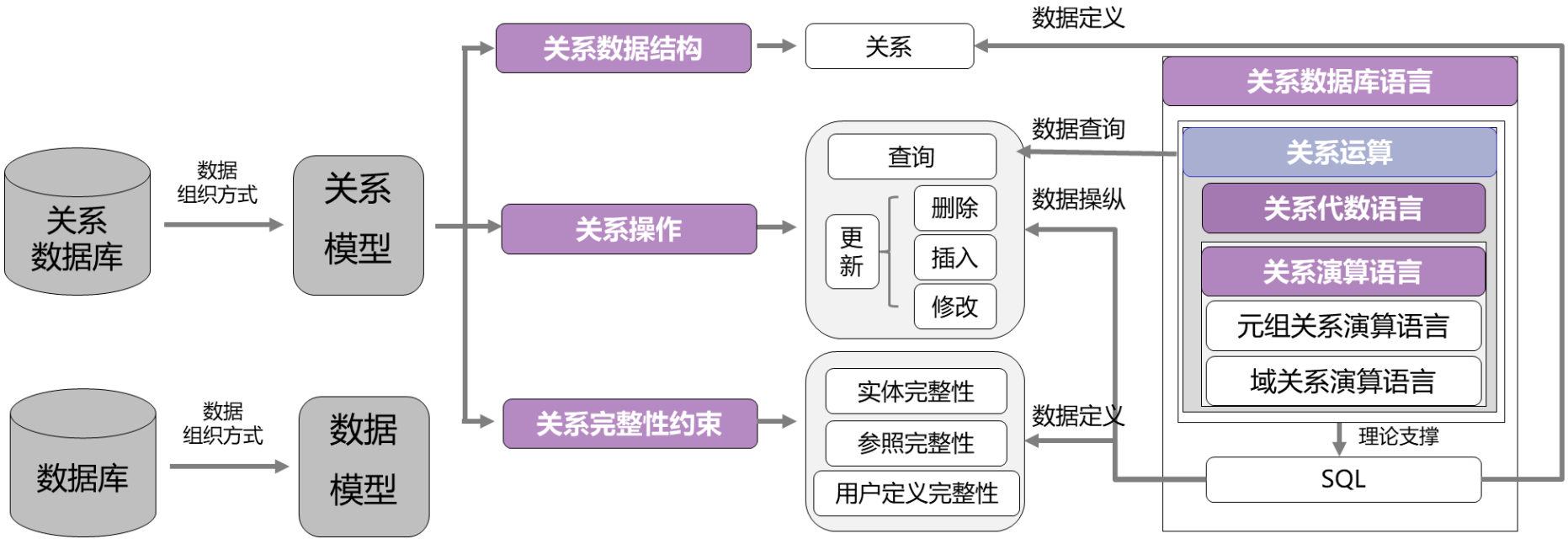
数据模型和数据库模式

- 数据模型和数据库模式之间的关系



目录

数据库语言


关系数据库
数据组织方式
关系模型
数据库
数据组织方式
数据模型
关系数据结构
关系操作
关系完整性约束
关系
查询
更新
删除
插入
修改
实体完整性
参照完整性
用户定义完整性
数据定义
数据查询
数据操纵
数据定义
关系数据库语言
关系运算
关系代数语言
关系演算语言
元组关系演算语言
域关系演算语言
理论支撑
SQL

数据库语言

- **数据定义语言**（Data-definition language, DDL）
  - Specify the database schema and various constraints
- **数据操纵语言**（Data-manipulation language, DML）
  - Express database queries and updates

- SQL同时包含DDL和DML

数据定义语言

- DDL: 定义数据库模式
  - 例如：

```
create table department (
dept_name    char(20),
building     char(15),
budget       numeric(12,2));
```

- DDL compiler generates a set of tables stored in a **data dictionary (数据字典)**
- Data dictionary contains metadata (i.e., data about data)
  - Database schema
  - Data storage
    - specify the storage structure and access methods
  - Consistency constraints
    - domain constraints
    - referential integrity (reference constraints in SQL)
    - authorization

数据操纵语言

- **DML**: 访问或操纵按照某种适当数据模型组织的数据
  - also known as query language

- **两种类型的DML**
  - **Procedural DML** -- specify what data are needed and how to get those data.
  - **Declarative DML** -- specify what data are needed without specifying how to get those data.
    - also referred to as non-procedural DMLs
    - easier to learn and use than procedural DMLs

SQL

- SQL: Structured Query Language, widely used **non-procedural language**
  - E.g., find all the instructors in the department of Computer Science

```
select   instructor.name
from     instructor
where    instructor.dept_name = 'Computer Science'
```

  - E.g., find the IDs of all the instructors from those departments with a budget larger than 95000

```
Select   instructor.ID
from     instructor, department
where    instructor.dept_name = department.dept_name and
department.budget > 95000
```

- **应用程序访问数据库**
  - Embedded SQL
  - Application program interface (e.g., ODBC/JDBC)

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

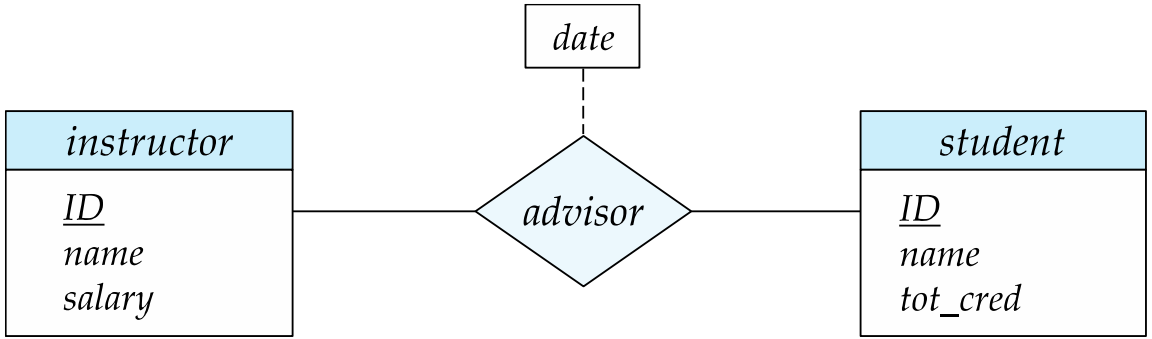
目录

数据库设计

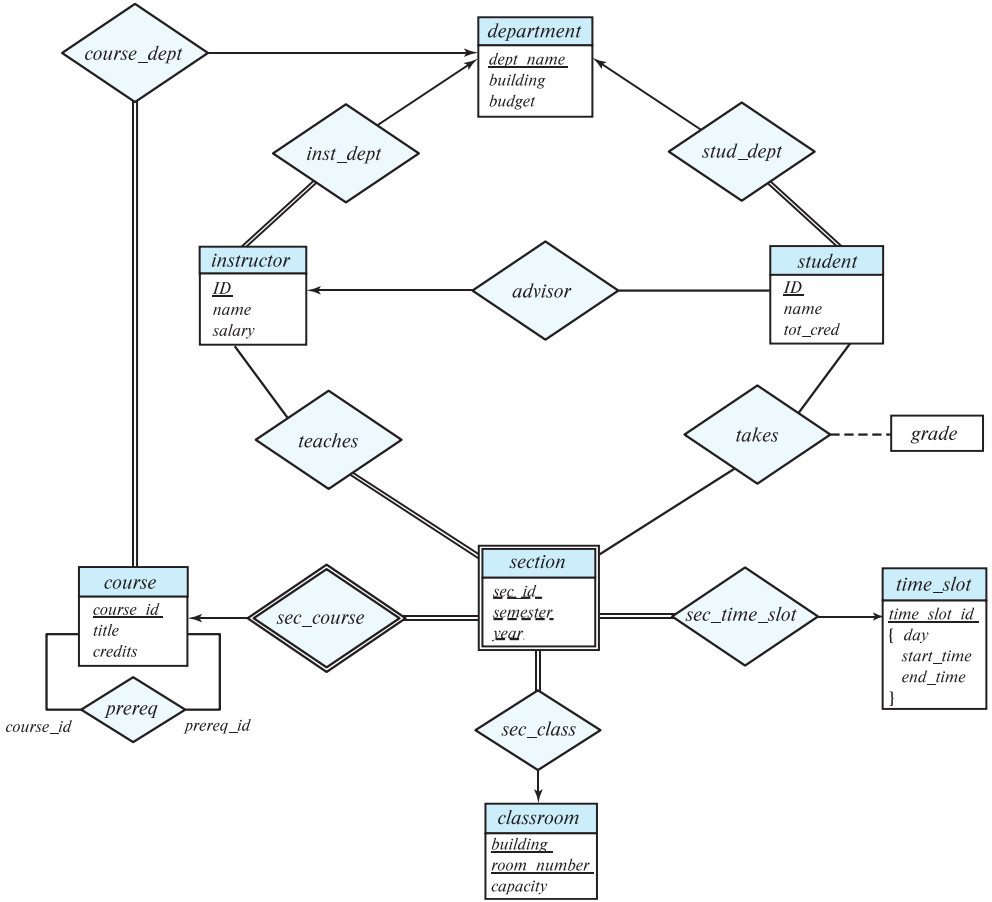
- **数据库结构设计过程**
  - **概念设计(Conceptual design)**: Decide what should be covered and their relations
    - Use the entity-relationship (E-R) model
  - **逻辑设计(Logical design)**: Decide the database schema, find a “good” collection of relation schemas (关系模式)
    - Business decision – What attributes should be recorded in the database?
    - Computer science decision – What relation schemas should we have and how should the attributes be distributed among the various relation schemas?
      - employ a set of algorithms (collectively known as normalization) that takes as input the set of all attributes and generates a set of tables
  - **物理设计(Physical design)**: Decide the physical layout of the database

E-R模型

- **E-R模型通过实体和关系建模数据**
  - **实体(Entity):** a “thing” or “object” that is distinguishable from other objects
    - Described by a set of attributes
  - **关系(Relationship):** an association among two or more entities



大学数据库的E-R模型


course_dept
department
dept_name
building
budget
inst_dept
stud_dept
instructor
ID
name
salary
advisor
student
ID
name
tot_cred
teaches
takes
grade
course
course_id
title
credits
sec_course
section
sec_id
semester
year
sec_time_slot
time_slot
time_slot_id
{ day
start_time
end_time
}
prereq
course_id
prereq_id
sec_class
classroom
building
room_number
capacity

目录

数据库引擎

- **存储管理器(Storage manager)**
  - Minimize the need to move data between disk and main memory
- **查询处理器(Query processor)**
  - Translate updates and queries written in a non-procedural language, at the logical level, into an efficient sequence of operations at the physical level to help the database system to facilitate the access to data
- **事务管理器(Transaction manager)**
  - Ensure that the database remains in a consistent (correct) state despite system failures, and that concurrent transaction executions proceed without conflicting

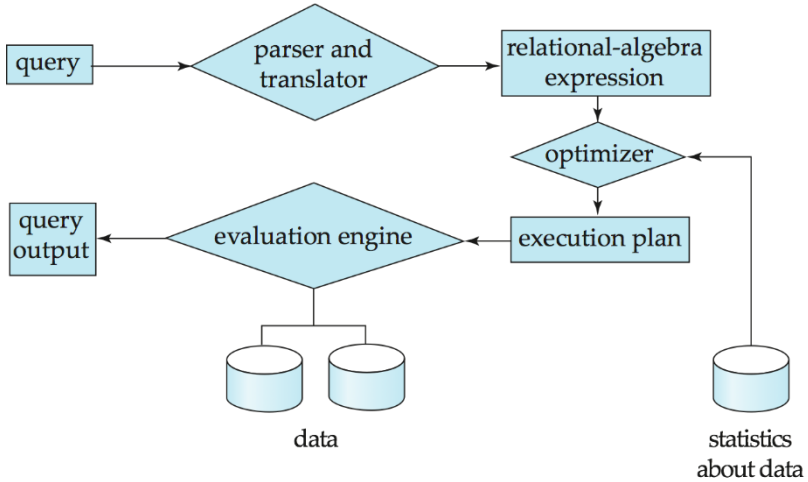
数据库引擎

- **存储管理器**
  - 数据库中存储的低层数据和提交查询之间交互的程序模块，负责数据的存储、检索和更新
    - The raw data are stored on the disk using the file system provided by the operating system.
    - The storage manager translates various DML statements into low-level file-system commands.

- **关键问题**
  - 文件组织
  - 存储访问
  - 索引结构

查询处理器

- **DDL解释器**
  - Interprets DDL statements and records the definitions in the data dictionary.
- **DML编译器**
  - Translate DML statements into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.
  - Query optimization
- **查询执行引擎**
  - Execute low-level instructions generated by the DML compiler



事务管理器

同济大学 TONGJI UNIVERSITY

- **事务(Transaction)**
  - A collection of operations that perform a single logical function in a database application
  - **ACID**: Atomicity(原子性)、consistency(一致性)、Isolation(独立性)、durability(持久性)

- **事务管理器(Transaction manager)**
  - Ensure that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures
  - **并发控制管理器**: Control the interaction among the concurrent transactions to ensure the consistency of the database.
  - **恢复管理器**: Failure recovery

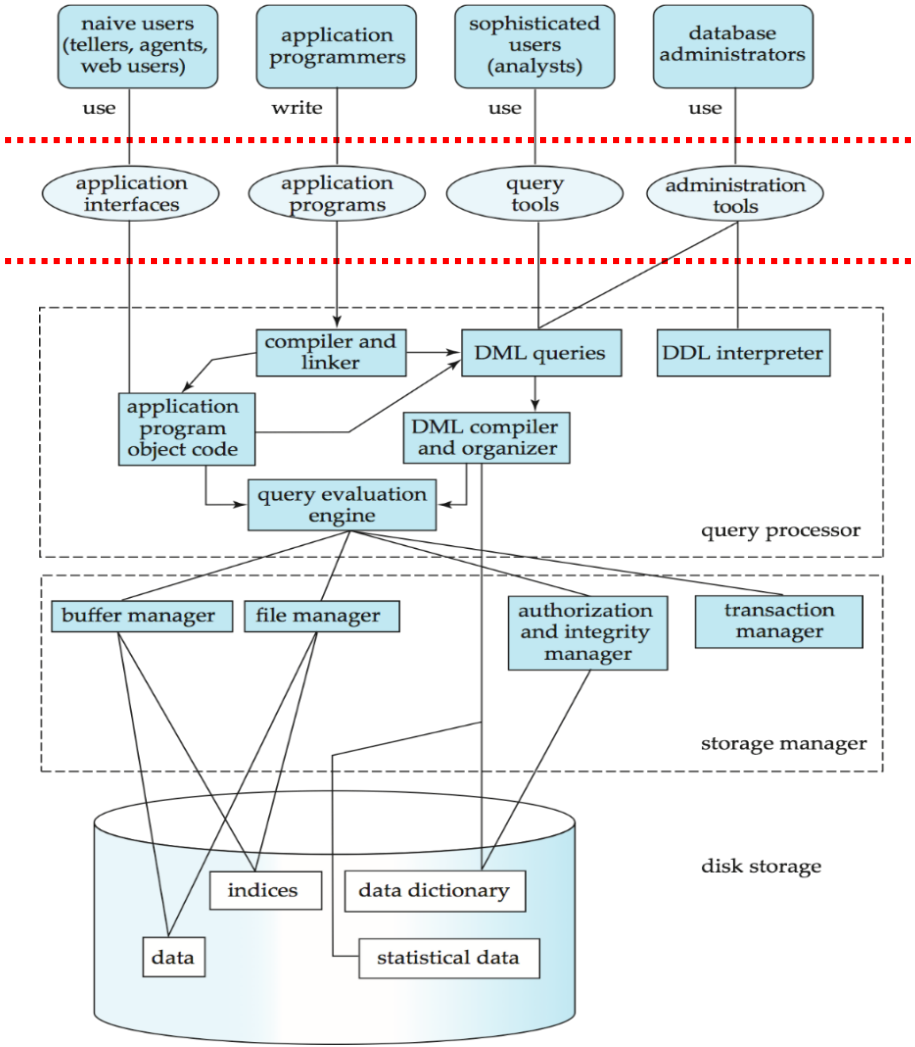
思考?

- One transaction that transfers funds from account A to account B could is composed of two separate steps:
  - debits (借记) account A,
  - credits account B.
- Will the execution of these two steps one after the other preserve consistency?
- Are steps A and B transactions?

目录

- 数据库系统
- 数据视图
- 数据库语言
- 数据库设计
- 数据库引擎
- 数据库和应用体系结构
- 数据库用户和管理员
- 数据库系统的历史
- 前沿发展方向

数据库系统结构



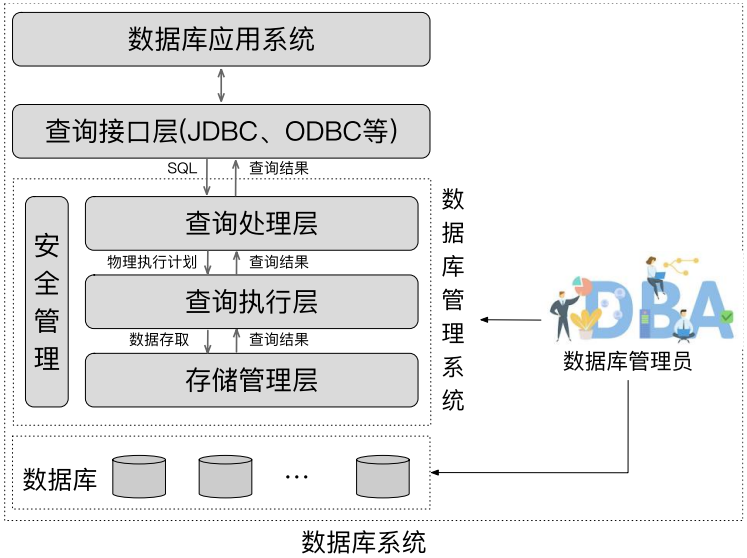
用户

应用和工具

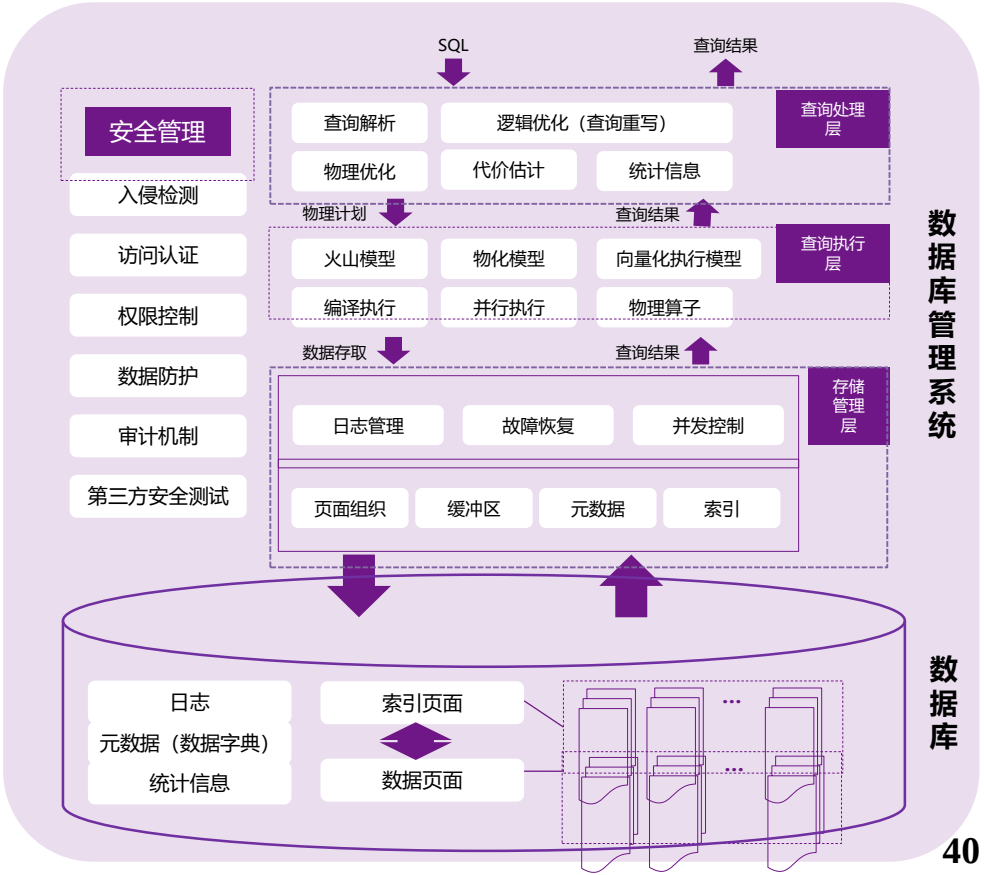
数据库管理系统

数据库

数据库系统架构

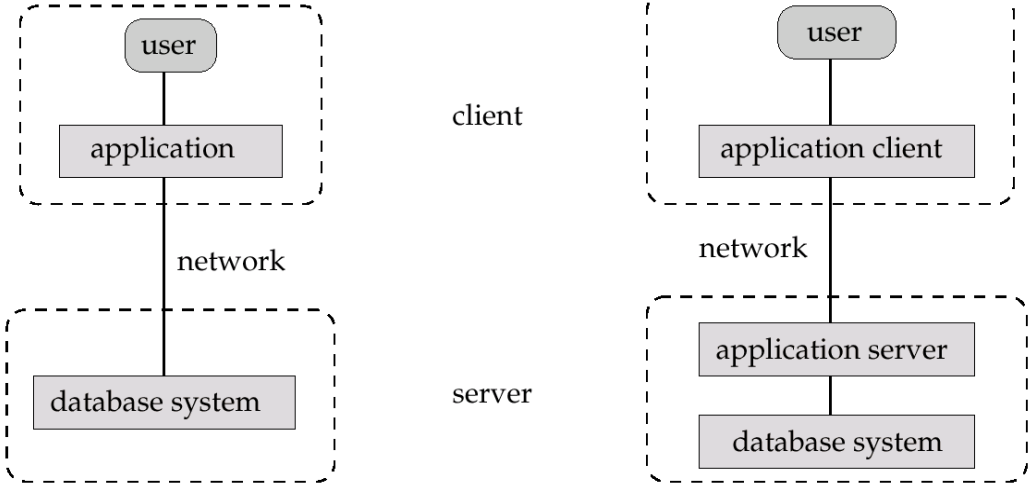

数据库应用系统
查询接口层(JDBC、ODBC等)
SQL
查询结果
安全管理
查询处理层
物理执行计划
查询结果
查询执行层
数据存取
查询结果
存储管理层
数据库管理系统
DBA
数据库管理员
数据库
…
数据库系统

细化


SQL
查询结果
安全管理
查询解析
逻辑优化（查询重写）
查询处理层
物理优化
代价估计
统计信息
入侵检测
物理计划
查询结果
访问认证
火山模型
物化模型
向量化执行模型
查询执行层
权限控制
编译执行
并行执行
物理算子
数据存取
查询结果
数据防护
存储管理层
审计机制
日志管理
故障恢复
并发控制
第三方安全测试
页面组织
缓冲区
元数据
索引
数据库管理系统
日志
索引页面
元数据（数据字典）
统计信息
数据页面
数据库

数据库应用系统结构

- **两层体系结构**: e.g., client programs using ODBC/JDBC to communicate with a database directly
- **三层体系结构**: e.g., web-based applications, and applications built using “middleware”



a. two-tier architecture

b. three-tier architecture

目录

数据库用户

- **数据库用户分类**
  - **无经验用户(Naive users)**
    - Use application programs that have been developed previously
    - E.g., people access database over the web, ATMs, various apps
  - **应用程序员(Application programmers)**
    - Interact with system through DML calls
  - **熟练用户(Sophisticated users)**
    - Form requests using a database query language

数据库管理员(Database Administrator, DBA)

- DBA是数据库系统中所有活动的管理者和协调者，对DBMS以及企业的信息资源和需求有较好的理解
- **DBA's duties**
  - schema definition
  - storage structure and access method definition
  - schema and physical organization modification
  - granting of authorization for data access
  - routine maintenance
  - specify integrity constraints
  - act as liaison with users
  - monitor performance, and respond to changes in requirements

目录

数据管理的三个阶段

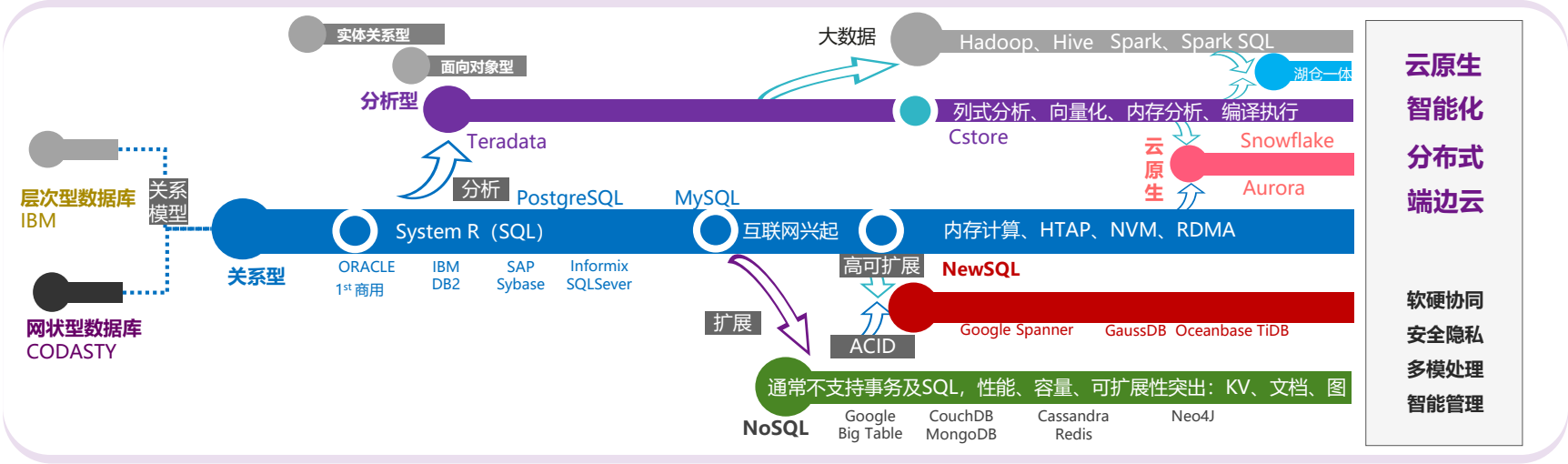
— 20世纪50年代中期　20世纪50年代中期 — 60年代中期　20世纪60年代末 — 至今

人工管理 → **文件系统管理** → **数据库管理**

	人工管理	文件系统管理	数据库管理
应用背景	满足科学计算	满足科学计算和数据管理	大规模数据管理
硬件条件	无直接存取存储设备	磁盘、磁鼓	大容量磁盘
软件条件	没有操作系统	有文件系统	数据库管理系统
处理方式	批处理	联机实时处理、批处理	联机实时处理、批处理、分布式处理
实现方式	通过卡带、卡片和磁带等计算机间接存取设备实现	通过磁盘和磁鼓等计算机直接存取设备实现	通过数据库管理系统与数据库进行交互以实现数据管理操作
特点	• 人力维护成本极高 • 数据与应用程序深度绑定 • 数据间缺乏组织和独立性 • 数据无共享、冗余度极大	• 人力维护成本很高 • 数据文件与应用程序耦合 • 数据共享性差、冗余度大 • 难以支持多用户并发访问	• 使用数据模型描述，无需应用程序定义，组织程度和共享程度高 • 编程友好，无需关心数据库文件的物理操作和系统控制 • 支持多用户并发访问

数据库发展史

60s 诞生 | **70s 兴起** • 关系型数据库 • SQL语言 | **80s 发展** • OLTP | **90s 分析** • OLAP | **00s 大数据** • NoSQL | **10s 云演进** • 云原生 • 分布式 | **20s？？**



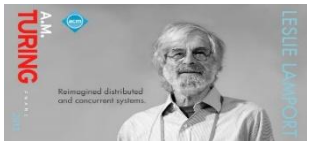
数据库概念
图灵奖Charles Bachman

关系代数
图灵奖： Edgar Codd

事务处理
图灵奖：Jim Gray

数据库系统
图灵奖Mike.Stonebraker

分布式一致性仲裁算法
图灵奖： Leslie Lamport



主流数据库管理系统

- **商用DBMS**
  - Oracle
  - IBM DB2 (from System R, System R*, Starburst)
  - Microsoft SQL Server
  - Sybase
  - Informix (acquired by IBM), …
- **开源DBMS**
  - PostgreSQL (from UC Berkeley’s Ingres, Postgres)
  - MySQL
- **其他DBMS**
  - NoSQL (Not only SQL, for big data): HBase, MongoDB, Neo4J, Redis, Cassandra
  - NewSQL: OceanBase, openGauss, etc.

DB-Engines Ranking

同济大学 TONGJI UNIVERSITY

Rank			DBMS	Database Model
Feb 2026	Jan 2026	Feb 2025		
1.	1.	1.	Oracle	Relational, Multi-model
2.	2.	2.	MySQL	Relational, Multi-model
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model
4.	4.	4.	PostgreSQL	Relational, Multi-model
5.	5.	5.	MongoDB	Document, Multi-model
6.	6.	↑ 7.	Snowflake	Relational
7.	7.	↓ 6.	Redis	Key-value, Multi-model
8.	8.	↑ 13.	Databricks	Multi-model
9.	9.	9.	IBM Db2	Relational, Multi-model
10.	10.	↓ 8.	Elasticsearch	Multi-model

https://db-engines.com/en/ranking

目录

数据库的3次主要变革

- **在过去50年里, 数据库主要经历了3次变革**:
  - 第1代是**单机数据库**, 解决了数据存储、数据管理和查询处理等问题, 代表系统包括Oracle, PostgreSQL和MySQL等
  - 第2代是**集群数据库**, 旨在为企业关键业务提供高可用性和可靠性保障, 代表系统包括Oracle RAC, IBM DB2和Microsoft SQL Server
  - 第3代是**分布式数据库**和**云原生数据库**, 旨在解决大数据时代的**弹性计算**和**动态数据迁移**问题, 代表系统包括亚马逊Aurora、华为openGauss和蚂蚁科技的OceanBase

数据库技术不断发展，向四大目标演进，具体分为九个方向

大数据时代，数据量不断爆炸式增长，数据存储结构也越来越灵活多样，日益变革的新兴业务需求催生数据库及应用系统的存在形式愈发丰富，这些变化均对数据库的各类能力不断提出挑战，推动数据库技术不断演进。

1 降低成本 提升易用性
- HTAP数据库
- 多模数据管理
- 数据库+AI

2 保障数据 安全可信
- 防篡改数据库
- 全密态数据库

3 提升功能 增强性能
- 软硬件协同发展
- 云原生数据库

4 满足新兴 业务需求
- 大规模图数据处理
- 实时湖仓

内容来自中国数据库年会（NDBC）

技术方向1：通过HTAP实现架构简化，成本降低

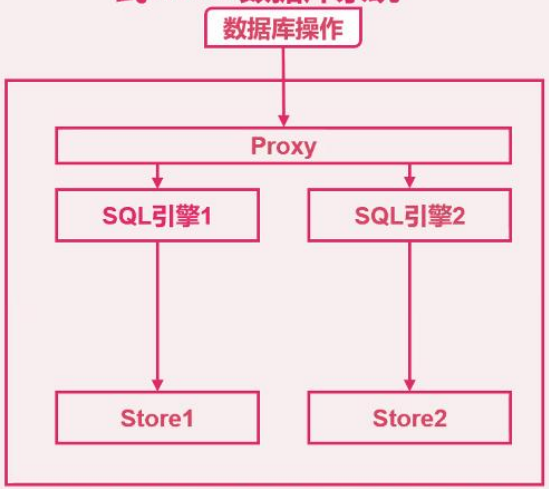
CAICT 中国信通院

HTAP数据库能够基于统一套引擎同时支撑业务系统运行和分析决策场景，避免在传统架构中，在线与离线数据库之间大量的数据交互，目前业界主要有三种实现方案。

方案一：紧耦合的多引擎多存储模式HTAP数据库系统

HyPer　SingleStore　ORACLE
SQL Server　openGauss　OceanBase
阿里云　Pivotal Greenplum　GBASE

方案二：松耦合的多引擎多存储模式HTAP数据库系统



TiDB　SequoiaDB 巨杉数据库
TRANSWARP 星环科技　MySQL

方案三：未来平台模式

这种方案主要针对未来的云原生模式。以操作算子的粒度确定响应何种查询。采用何种执行算子由云计算调度，主要采用Serverless的技术融合不同的执行算子满足HTAP的需求。

阿里云　Microsoft Azure
ORACLE　HUAWEI

技术方向2：多模数据管理备受关注，面临多重挑战

多模数据管理能力是指一个数据管理系统支持多种数据模型和语义访问。多模数据库拉近了多样化应用与数据库之间的距离。在大数据时代，多模的数据管理能够优化对于各种数据结构的管理。

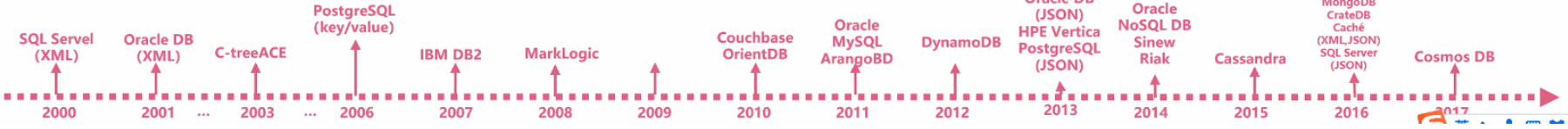
- 多模数据库可以将各种类型的数据进行集中存储、查询和处理
- 满足应用程序对于结构化、半结构化和非结构化数据的统一管理需求

存储类型	代表数据库	支持的数据类型	存储策略	查询语言	索引
关系型数据库	SQL Server	relational，JSON,XML	TEXT，关系型表	SQL扩展	B-tree，full-text
	IBM DB2	relational,XML	原生XML类型	SQL/XML扩展	XML paths/B+ tree，full-text
	Oracle	relational,XML，JSON，RDF	关系型	SQL/XML 或者SQL对JSON的扩展	bitmap，B+ Tree，函数索引，XML索引
宽表	Cassandra	text，UDT	稀疏表	SQL-like CQL	IR，B+
键值对	Riak	Key/Value,XML ,JSON	key/value对	Solr	Solr
	Redis	flat lists, sets, hash tables	key/value对	API	-
文档存储	ArangoDB	key/value,document,graph	文档存储	SQL-like AQL	hash
图	OrientDB	graph, doc- ument, key/value, object	key/value对和面向对象的连接	Gremlin, ex- tended SQL	SB-tree, ex- tendible hash- ing, Lucene
对象数据库	Cache´	object, SQL or multi- dimensional, document (JSON, XML) API	多维数组	SQL扩展	bitmap, bit- slice, standard

阿里云　ORACLE

SequoiaDB 巨杉数据库　mongoDB

TRANSWARP 星环科技　Microsoft Azure



技术方向3：AI助力数据库升级，数据库助力AI加速

CAICT 中国信通院

正在发言：

随着智能时代到来，AI 和数据库的结合也成为了研究热点，一方面通过 AI 技术优化数据库的设计和管理（例如学习型索引、学习型代价估计、智能参数调优等），另外一方面通过数据库系统和技术降低 AI 使用门槛，普惠 AI。

Database For AI：SQL 算子支持AI库内推理和训练

DB4AI

- 模型推理
  - 算子支持
  - 算子选择
  - 执行加速
- 模型训练
  - 特征选择
  - 模型选择
  - 模型管理
  - 硬件加速
- 数据自治
  - 数据发现
  - 数据打标
  - 数据清理
  - 数据谱系
- 数据库安全
  - 数据发现
  - 访问控制
  - SQL 注入

AI For Database：AI算法辅助数据库的智能优化和运维

AI4DB

- 数据库配置
  - 旋钮调节
  - 索引顾问
  - 视图顾问
  - SQL 重写
- 数据库优化
  - 基线估计
  - 成本估计
  - 连接顺序选择
  - 端到端优化
- 数据库设计
  - 学习索引
  - 数据结构
  - 事务管理
- 数据库监控
  - 健康监控
  - 活动监控
  - 性能预测

阿里云 数据库自治服务DAS　HUAWEI　openGauss　ORACLE　ORACLE Autonomous Database

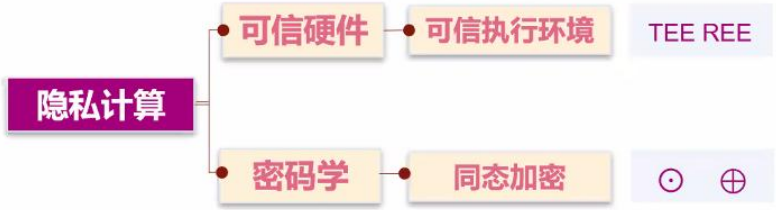
技术方向4：防篡改数据库实现全生命周期数据可信

CAICT 中国信通院

防篡改数据库是对指定数据具有高可信能力的数据库，即是具备一定阻止恶意修改、恶意修改可识别、数据恢复和不可抵赖能力的数据库，能够实现整个数据全生命周期的高度高可信、数据回溯。具备防篡改、可追溯、高性能、高可靠、透明可信等特质。

技术方向5：全密态数据库保障密文数据有效流通

全密态数据库能够提供对应用透明的加解密能力，在数据库系统中数据的全生命周期以密文形式进行处理，同时密钥掌握在授权用户手中的数据库管理系统。全密态数据库能够在数据密文的状态下对数据进行处理与使用，极大程度上保护了数据安全。



硬件方案的核心思路是将存放于普通环境（REE）的加密数据传递给TEE侧，并在TEE侧完成数据解密和计算任务

密码学方案的核心思路是整个运算过程都是在密文状态，通过基于数学理论的算法来直接对密文数据进行检索与计算

面临挑战

- 基于存储和计算开销的效率挑战
- 基于数据操作过程的安全性挑战

发展趋势

- 软硬结合的密态数据处理系统
- 实现动态数据的安全存储
- 支持范围查找的密态索引

技术方向6：数据库一体机促进软硬协同高效发展

CAICT 中国信通院

数据库技术的发展始终是追求更高的计算密度、更高的资源利用率和模块化、更少的定制化开发、更便捷的使用目标。数据库一体机是一套高度集成，专为数据库设计的高性能数据库底层平台。贯彻“业务应用快速部署”的理念，适用于中大型企业内部，对业务连续性、性能和可用性有极高要求的核心数据库系统。

产品架构

数据库一体机

核心部分

- 计算资源池
- 存储资源池
- 高速互联网络

高可用

计算节点、存储节点、网络交换链路可设计为全冗余架构设计及配置，无单点故障，采用负载均衡的多路径机制，保证系统各节点卓越的性能和高可用性。

采用分片的数据多副本容错机制，将数据冗余副本同时随机存放到不同的存储节点，避免单点故障，提高数据并发处理性能。

产品性能

闪存介质　高速光纤通道交换机

通过池化大量节点中的资源提供横向扩展服务，数据分布在多个可用节点上，每个节点具有2x100Gb高性能网络连接。可提供随存储资源和节点线性扩展的出色 IO 性能。

存储容量

数据库一体机支持压缩功能，能够帮助用户提升2-3倍的存储容量

ORACLE　TERADATA　沃趣 WOQUTECH　达梦数据库　云和恩墨 ENMOTECH　天玑科技 Pioneer In IT Service

技术方向7：云原生数据库继续进化，实现新突破

企业云化从“以资源为中心”演进到“以应用为中心”。下一代云原生数据库在架构和技术能力上能感知应用特征，将计算、内存、存储分层池化，实现三者的解耦，从而可以实现独立伸缩，进一步提升性价比。

传统云数据库存在瓶颈

- 存储空间浪费
- 较大的恢复时间目标（RTO）及数据滞后
- 计算资源浪费
- 系统性能受限
- 网络带宽资源消耗大

云原生数据库优势

- 应用透明
- 弹性成本
- 一站式数据处理
- 安全可靠

典型应用场景

- 时效性要求高
- 业务流量变化大
- 混合负载应用
- 追求高性价比
- 可靠性要求高
- 提升云资源利用率

云原生数据库未来方向

架构上云原生数据库未来要实现内存池化和全栈解耦。

在OLTP和OLAP能力融合的基础上，云原生数据库未来更进一步结合Memory Fabric软硬协同，实现网络吞吐的大幅度缩减，更极致的性能提升。

阿里云 aws HUAWEI 腾讯云 TRANSWARP 星环科技 Microsoft Azure Google Cloud

技术方向8：图数据库处理能力迈向万亿超大图规模

CAICT 中国信通院

正在发言：姜春宇-信通

《“十四五”软件和信息技术服务业发展规划》，明确提出“突破大规模并行图数据处理关键技术”，伴随全球数字化转型日趋火热，海量图数据的挖掘利用将直接影响企业数字化、智能化进程。

图数据库在新兴领域有巨大价值

图数据库作为以图论为设计原理的数据库管理系统，将现实世界的实体和实体关系抽象表达为顶点和边，擅长图数据的高效存储、查询、计算、分析，能有效解决关系型数据库难以解决的大数据关联难题。

- 金融风险
- 精准零售
- 物流优化
- 能源调度
- 生物制药
- 智能交通
- 疫情防控
- 其它新兴领域等

机遇

随着人工智能、图联邦学习、图数据库处理图数据技术不断发展，且数据必然以井喷方式增长，在此背景下，**建立高性能的分布式图计算理论及分布式存储技术，支持万亿规模超级大图数据的存储和计算**，实现实时大图数据增删改查，并完成多跳查询、模式挖掘等典型图查询、图计算等操作尤为关键。

挑战

未来支持万亿级图规模系统应该是基于普通PC机构建的分布式系统，**如何避免使用昂贵的大型机或者小型机，减少企业软硬件成本和运维成本**是亟待解决的问题。

技术方向9：分析型数据库迈入实时湖仓集约阶段

近十年，业界不断总结当前时代的数字业务特点、逐步突破前两代数据分析的技术障碍，兼具数据仓库和数据湖两者优点的“湖仓一体”架构诞生，逐步发展为统一存储、统一元数据管理、存算分离、弹性扩展、多租户、可插拔存储、分级资源管理等功能丰富的统一框架时代。

企业数据分析与应用需求变化趋势

- 企业数据分析的应用场景变得更加广泛
- 企业的总数据量以及实时数据正在以前所未有的速度爆发式增长
- 企业业务和分析系统上云正在加速

传统分析型数据库的主要缺陷

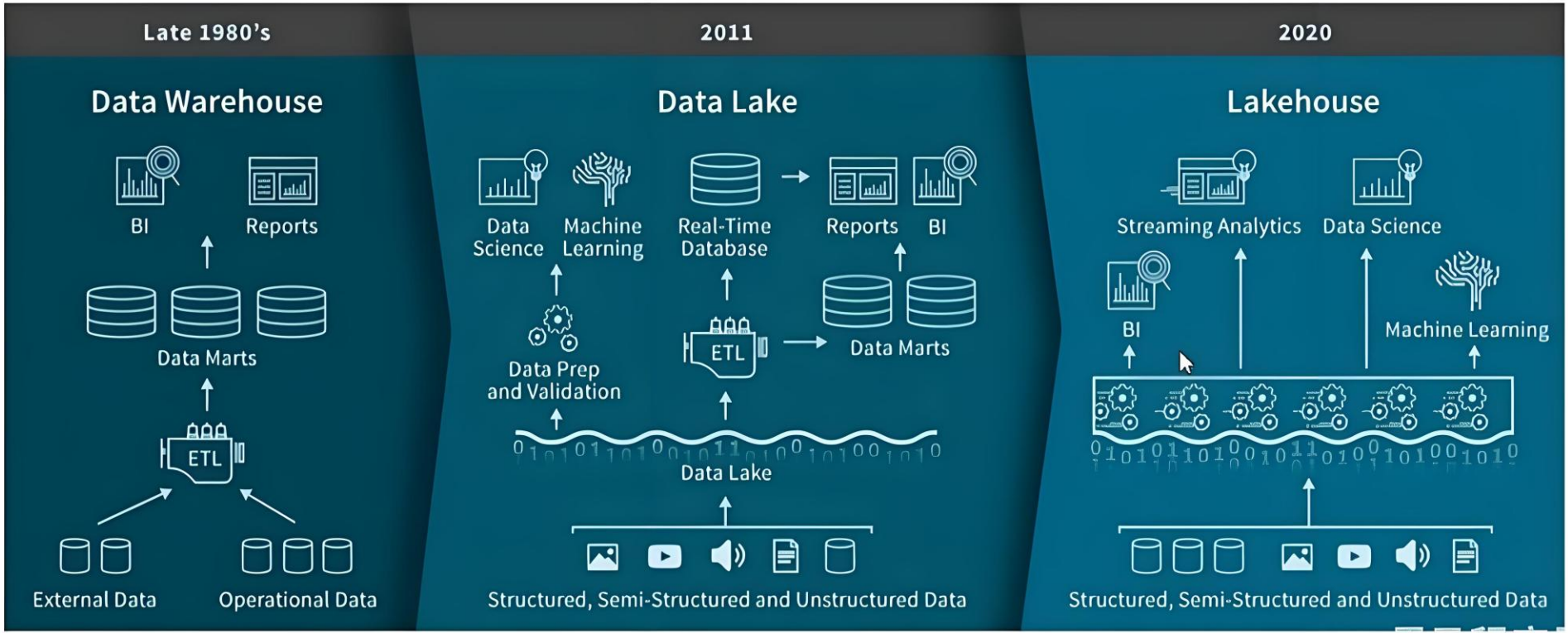
- 存储计算资源难以弹性扩展，制约了大数据量下数据分析的性能和速度
- 缺乏优化的性价比，资源消耗大、成本高
- 对人工智能和机器学习等高级分析的支持不足
- 系统架构复杂，稳定性差，管理和维护成本高

第四代分析型数据库“智能湖仓”诞生

- 通过分析近年来数据分析的应用场景、数据以及计算环境等方面发生的变化，以及现有的分析型数据库在应对这些变化时的主要缺陷，我们认为，下一代的分析型数据库必然会朝着增强分析性能、提升易用性、降低使用成本的方向发展。

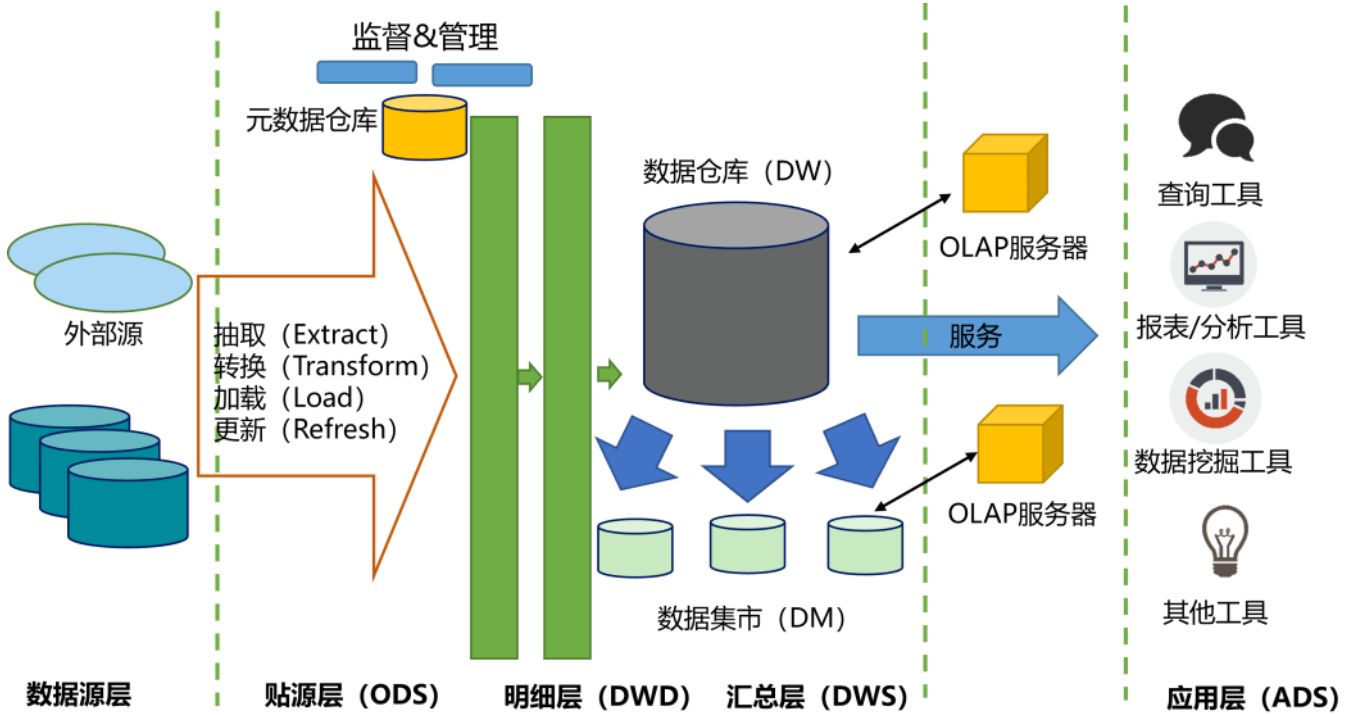
- 通过元数据层在数据湖上实现数据管理功能
- 流批一体，简化系统架构
- 云原生、存算分离

Data Warehouse, Data Lake & Lakehouse



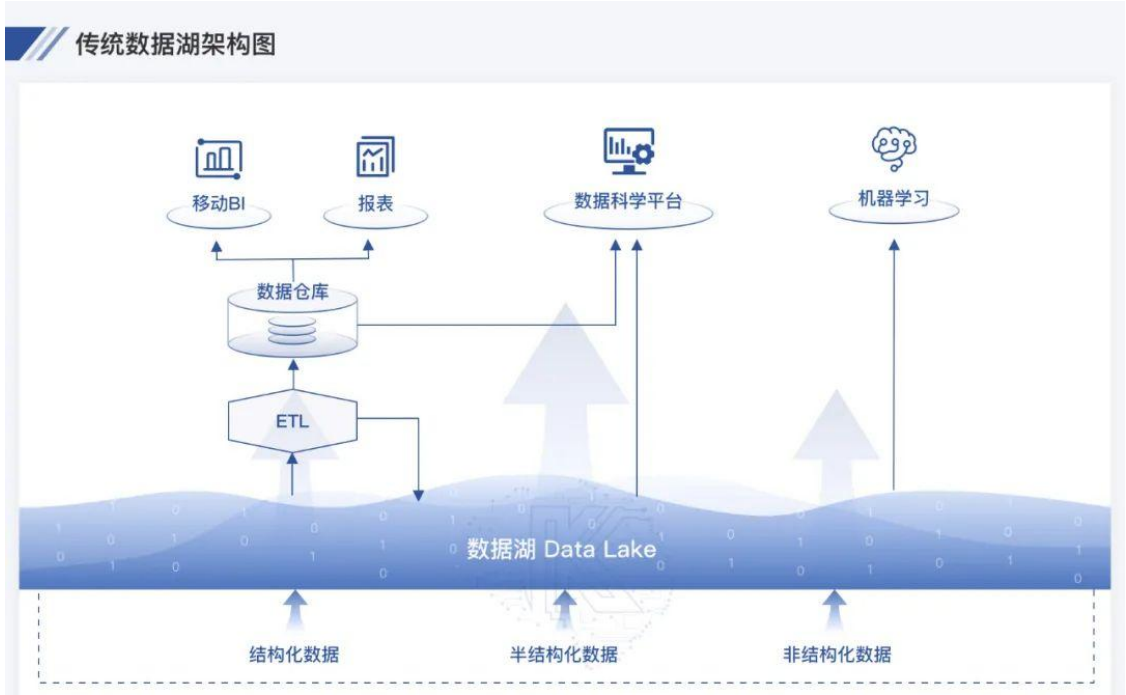
数据仓库（Data Warehouse）

- 一个面向主题的、集成的、相对稳定的、反映历史变化的数据集合，用于支持管理决策和信息的全局共享



数据湖（Data Lake）

- 数据湖是一种不断演进、可扩展的大数据存储、处理、分析的基础设施。它就像一个大型仓库，可以存储任何形式（包括结构化和非结构化）和任何格式（包括文本、音频、视频和图像）的**原始数据**

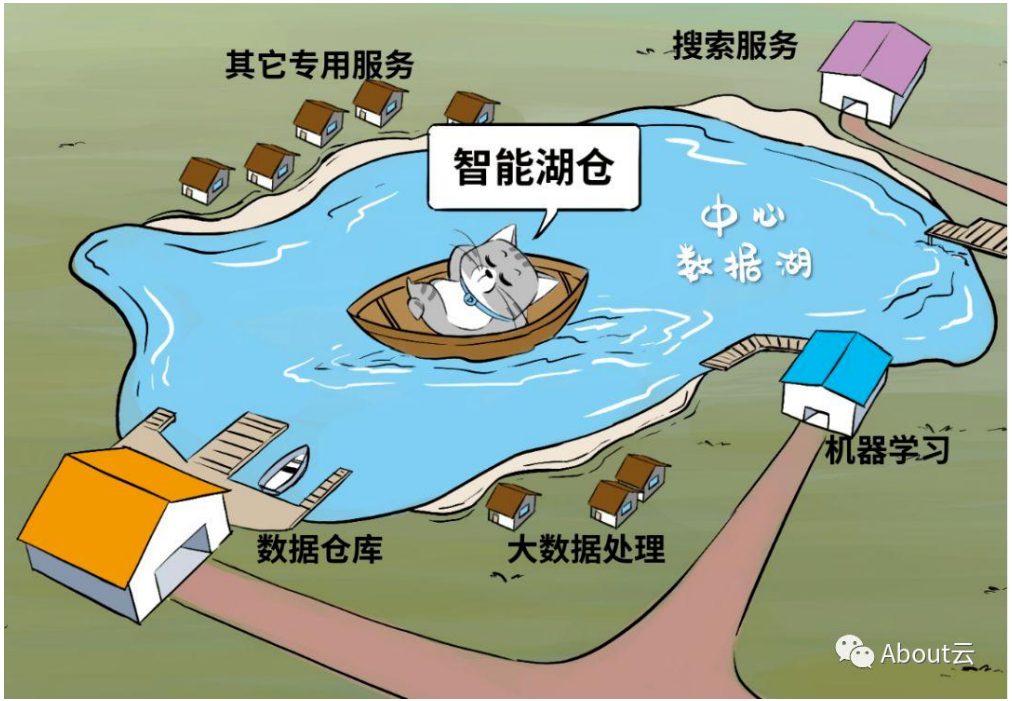


数据仓库 vs. 数据湖

特性	数据仓库	数据湖
数据类型	结构化	结构化，非结构化，半结构化
处理过程	离线	离线，实时
数据质量	高	低
开发效率	低，开发周期	高，随时处理
主要使用人员	业务分析人员为主	数据科学，数据开发
主要支持业务	业务报告，BI分析，可视化，驾驶舱等	AI智能，数据探索，特征处理，标签画像，实时大屏等
是否标准	有标准	没有标准
是否灵活开发	数据开发完成才能使用	灵活的租户开发服务

湖仓一体（Lakehouse）

- **湖仓一体**是一种新型开放式架构，将数据湖和数据仓库的优势充分结合，它构建在数据湖低成本的数据存储架构之上，又继承了数据仓库的数据处理和管理功能，打通数据湖和数据仓库两套体系，让数据和计算在湖和仓之间自由流动



我国数据库将规模化向重点行业深入应用创新

CAICT 中国信通院

金融领域核心系统分布式改造需求旺盛

金融

系统压力大，旧系统需要分布式改造

以**中信银行、邮储银行、国家开发银行、张家港农商行、微众银行、常熟农商行**等为代表的新核心系统相继上线并平稳运行。

电信行业数据库应用创新发展方向

- 未来电信运营商数据库改造将从周边非核心系统过渡到核心应用系统，如CRM和BOSS等；
- 数据库云化趋势明显，通过解耦化、模块化、标准化以PaaS方式为应用提供弹性、自治的服务。

能源行业数据库应用创新发展方向

- 发电侧企业信息系统改革加强，风和光伏为代表的发电侧对趋势预测等数据分析需求提出新要求。
- 电网侧数据技术需求持续增长，电网企业物联网应用场景和传感器数量不断增加，监控系统产生的数据规模与日俱增，对数据库性能和稳定性提出更高要求。

政务行业数据库应用创新发展方向

- 数字化改革效能持续提升。电子政务领域的国产化应用推广，已从以省、部级单位为主的应用示范阶段，深入到以地市、县级单位为主的应用推广阶段。
- 由电子公文系统扩展至其他核心政务应用系统。
- 集中式数据库为主导，其它类型数据库为辅。相比分布式数据库，我国集中式数据库技术较为成熟可控，生态建设日趋完善，能够满足政务领域的应用需求。

我国数据库产业生态持续向好，不断繁荣

CAICT 中国信通院

正在发言：姜春宇-信通院

人才规模持续扩大

人才

数据库内核开发人才缺口较大

随着基础软件利好政策不断出台，资本持续投入，行业待遇日益提升，涌入数据库行业的人才规模将不断提升。

学术创新不断突破

2018-2021年，我国高校和企业在VLDB、SIGMOD和ICDE中稿比例逐年提升；

代表企业：阿里巴巴、中国移动、华为、腾讯、字节跳动、蚂蚁科技集团、百度等；

代表高校&研究机构：清华、港技大、北大、港中文、浙大、人大、华中师大、华东师大、中科院、深算院等。

软硬技术深度融合

数据库将持续与云计算、区块链、人工智能、隐私计算、新型硬件等技术领域融合发展。

数据库将向着更高效、更安全和更智能的目标发展，推动数据库形态和架构持续变革。

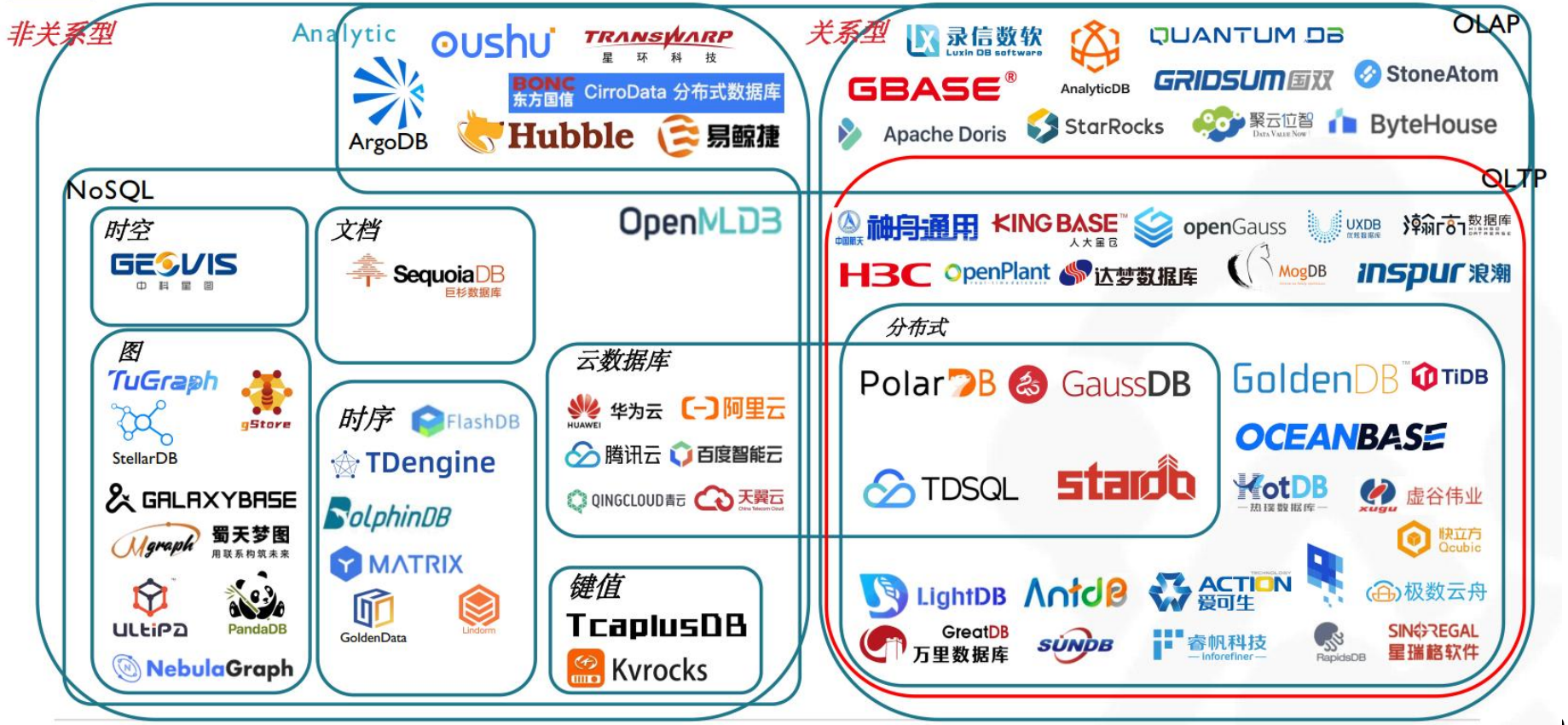
产业变革悄然而至

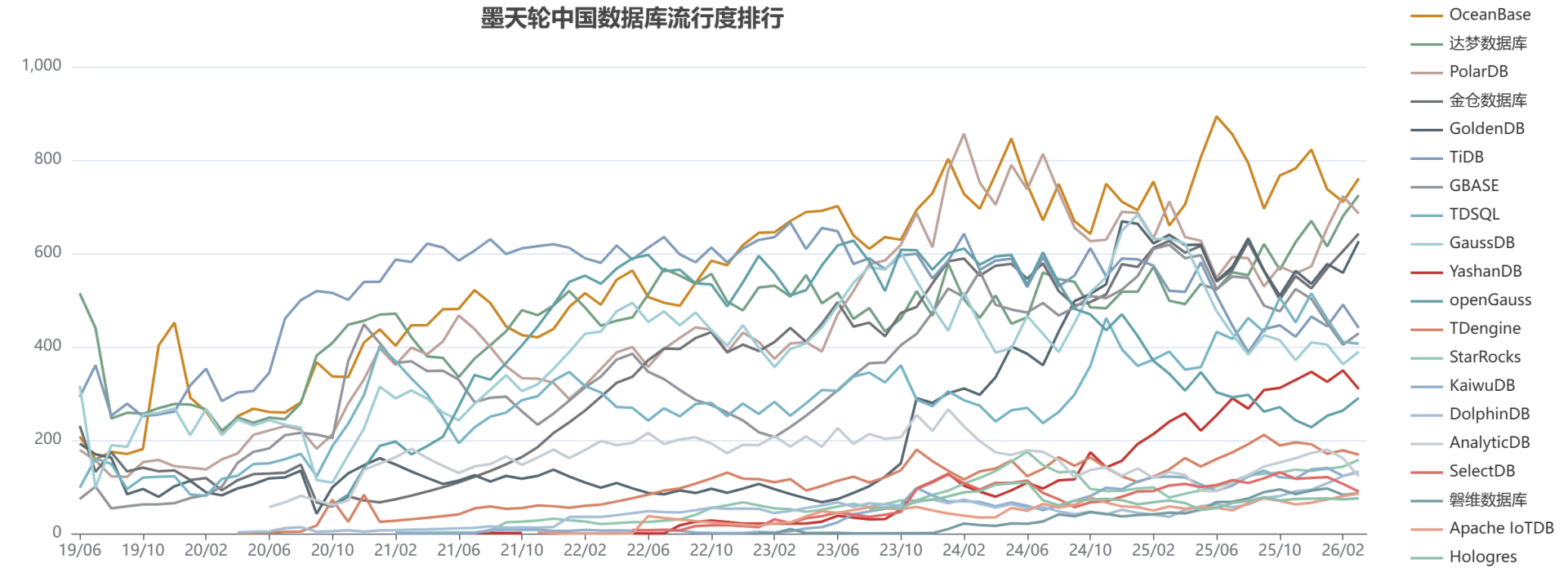
全球数据库产业格局加速变革，以美中为主的数据库产品数量和人才规模不断提升。

我国数据库产业正经历由“数量型”向“质量型”转变。

产业图谱 - 中国数据库产品图谱家族

墨天轮





https://www.modb.pro/dbRank

孰弱孰强 - 中国数据库论文数盘点

墨天轮

数据库论文能够展现数据库行业最新研究成果，发文数量能够一定程度上体现各发文单位在数据库学术领域、技术钻研上的成就。目前中国数据库论文数占全球总数12%，并呈现发文数逐年递增的趋势，这标志着中国数据库领域自主创新能力明显提升。
以下数据基于Web of Science核心合集，检索主题为Database的检索机构，检索日期为2022年7月。

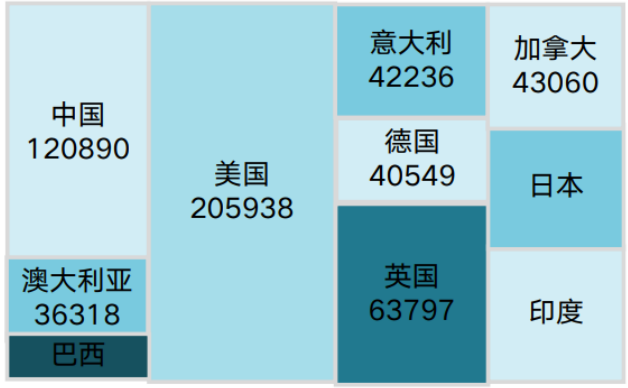
数据库行业支撑体系

数据库顶级会议
VLDB、SIGMOD、ICDE

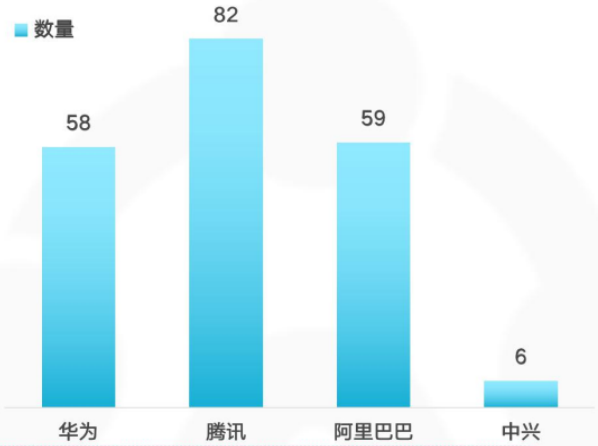
高校
清华大学、中国科学院大学、华中科技大学
浙江大学等

企业
阿里巴巴、华为、腾讯
百度、平凯星辰、达梦、
人大金仓等

各国数据库论文数



中国数据库厂商论文数



学术机论文数

北京大学	中国科学院	清华大学	香港中文大学	香港科技大学	武汉大学
4036	10032	4036	4036	4036	2396

来源：https://www.webofscience.com/

影响数据库发展的因素

关系，键值, 文档, Json，
图，向量

SMP, NUMA
RAM, NVM
RDMA
GPU, FPGA
SGX, TEE

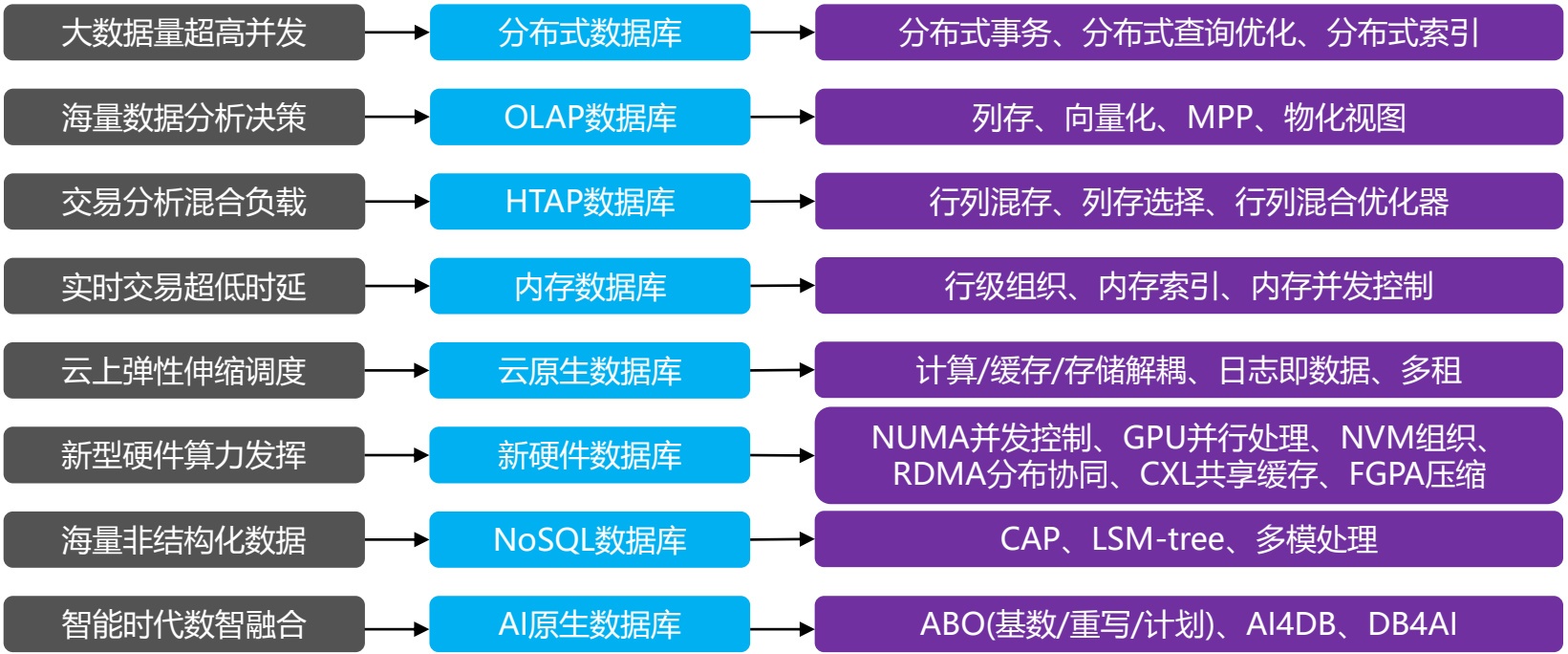
线下On premise
云Cloud
Serverless
端边云End-Edge-Cloud

单机，单机主备，多主，集群，分布式

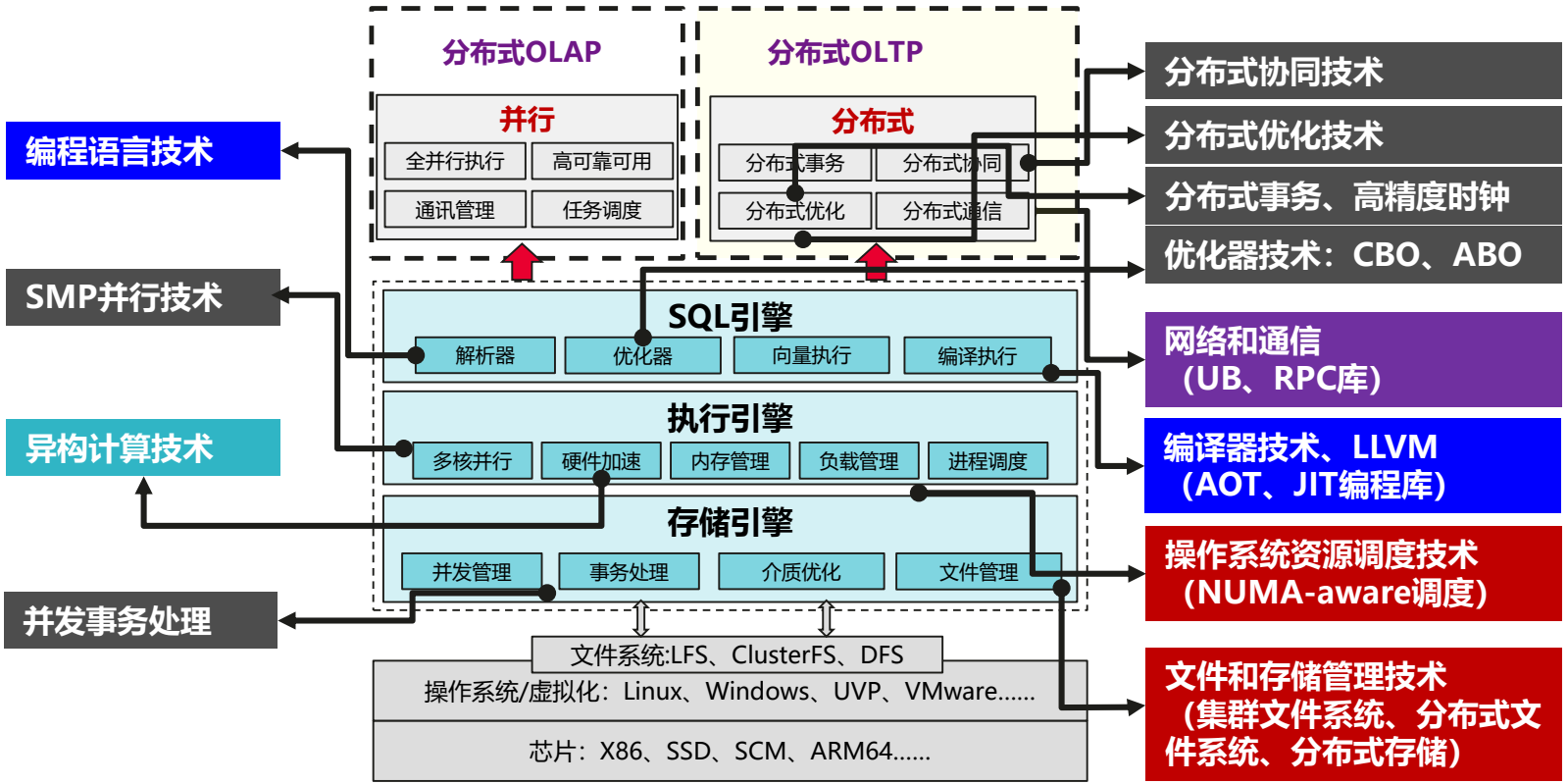
OLAP
OLTP
HTAP
时序、流、AI
安全隐私

2PC+Paxos
DB+LLVM
大数据
区块链
AI

高级数据库技术


大数据量超高并发
分布式数据库
分布式事务、分布式查询优化、分布式索引
海量数据分析决策
OLAP数据库
列存、向量化、MPP、物化视图
交易分析混合负载
HTAP数据库
行列混存、列存选择、行列混合优化器
实时交易超低时延
内存数据库
行级组织、内存索引、内存并发控制
云上弹性伸缩调度
云原生数据库
计算/缓存/存储解耦、日志即数据、多租
新型硬件算力发挥
新硬件数据库
NUMA并发控制、GPU并行处理、NVM组织、
RDMA分布协同、CXL共享缓存、FGPA压缩
海量非结构化数据
NoSQL数据库
CAP、LSM-tree、多模处理
智能时代数智融合
AI原生数据库
ABO(基数/重写/计划)、AI4DB、DB4AI

多方向交叉融合创新


分布式OLAP
分布式OLTP
并行
全并行执行
高可靠可用
通讯管理
任务调度
分布式
分布式事务
分布式协同
分布式优化
分布式通信
SQL引擎
解析器
优化器
向量执行
编译执行
执行引擎
多核并行
硬件加速
内存管理
负载管理
进程调度
存储引擎
并发管理
事务处理
介质优化
文件管理
文件系统:LFS、ClusterFS、DFS
操作系统/虚拟化：Linux、Windows、UVP、VMware.....
芯片：X86、SSD、SCM、ARM64.....
编程语言技术
SMP并行技术
异构计算技术
并发事务处理
分布式协同技术
分布式优化技术
分布式事务、高精度时钟
优化器技术：CBO、ABO
网络和通信（UB、RPC库）
编译器技术、LLVM（AOT、JIT编程库）
操作系统资源调度技术（NUMA-aware调度）
文件和存储管理技术（集群文件系统、分布式文件系统、分布式存储）

小结

- **数据库相关概念**
  - 数据库、数据库管理系统、数据库系统
  - 文件处理系统的缺点
- **数据视图**
  - **数据抽象**：物理层、逻辑层、视图层
  - 物理/逻辑数据独立性
  - **数据模型**：关系模型、E-R模型
- **数据库语言**
  - 数据定义语言
  - 数据操纵语言
    - 过程式、声明式
- **数据库设计**
  - 概念设计（E-R模型）
  - 逻辑设计
  - 物理设计
- **数据库引擎**
  - 存储管理器
  - 查询处理器
  - 事务管理器
- **数据库系统的结构**
- **数据库用户**
  - DBA
- **数据库前沿发展方向**