Database Systems

College of Computer Science and Technology

Zhejiang University

2024

- Instructor: Ling CHEN 陈岭
- Homepage: https://person.zju.edu.cn/lc
- Class Time: Mon. 13:25-15:50
- Practice Hours: Mon. 16:15-17:50
- Office Add: 玉泉校区曹光彪楼主楼 616B
- Tel: 13606527774
- Email: lingchen@cs.zju.edu.cn
- TA: Xiaofeng WANG 王晓峰
- Tel: 13255765087
- Email: 1634230100@qq.com

- **WeChat Group:** 数据库系统 -2024- 陈岭

群聊: 数据库系统-2024-陈岭

该二维码7天内(2月27日前)有效，重新进入将更新

Textbook:

- Database Systems Concepts 7th edition
- By Abraham Silberschatz, Henry F. Korth, and S. Sudarshan
- Higher Education Press, McGraw-Hill Companies

Reference books:

- Database Management Systems 3rd edition

  By Ramakrishnan and Gehrke
- Database Systems: The Complete Book

  By Garcia-Molina, Ullman, and Widom
- 数据库系统概论（第四版），萨师煊 王珊，高等教育出版社， 2006
- 数据库系统原理教程，王珊 陈红，清华大学出版社， 2003
- 数据库课程设计，陈根才 孙建伶 林怀中 周波，浙江大学出版社， 2007

  （实验参考书）

Grading Policy:

- 期末考试占 **50%**
- 实验和 **project** 占 **30%**
- 课程作业和课堂测验占 **20%**

【注： **Close book test, allow taking one handwritten A4 page note** 】

实验和大程

- 紫金港机房，专用服务器和用户
- 《实验和大程要求》
- 实验由 **5** 部分组成，每位同学单独进行，完成实验报告
- 大程推荐 **3-5** 人一组，具体参看《 **MiniSQL** 大程大纲》
- 实验 **5** 和大程需要现场运行验收

- **Lectures**
  - **Lecture slides in PPT format will be posted shortly before or after the lecture.**
  - **Many issues discussed in the lectures will be covered in the exams and assignments.**
  - **Please attend lectures regularly!**
- **Assignments**
  - **Paper-based + some programming.**
  - **Will be collected at the end of class on the due date.**
  - **Homework handed in by the due time will be graded for full credit, then a 30% late charge will be applied.**
  - **No late homework is accepted after on-time papers are returned, or the sample solution is made public, whichever occurs first.**

Chapter 1: Introduction

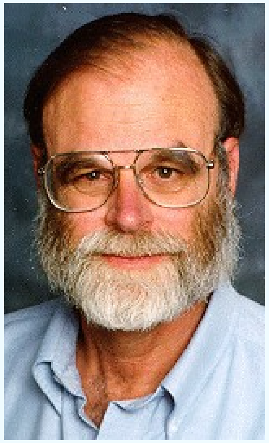
- Purpose of Database Systems
- View of Data
- Data Models
- Database Language
- Database Administrator
- Database Users
- Transaction Management
- Storage Management
- Overall System Structure

1.0 Why should we learn Database?

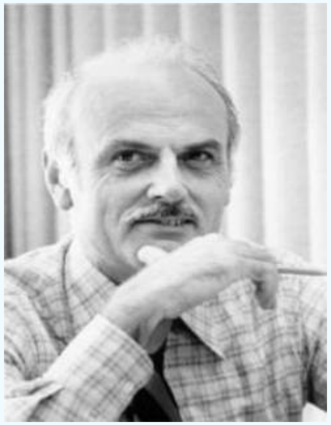
- **Data processing and management are the most important fields of computer applications, and the knowledge of database concepts is essential for computer scientists.**
- **Databases touch all aspects of our lives in information society, even when you don't see them**
  - **Banking: all transactions**
  - **Airlines: reservations, schedules**
  - **Universities: registration, grades**
  - **Sales: customers, products, purchases**
  - **Manufacturing: production, inventory, orders, supply chain**
  - **E-Government, E-Business**
  - ……
- **Of all courses you have taken, this may be the one that gets you a job ;o)**

Turing Award Winners Regarding Database Systems

2014, Michael Stonebraker, contributions in Object-Relational DBMS concepts and systems.

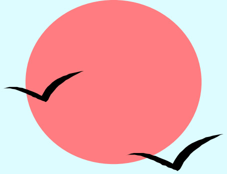


1998, James Gray, contributions in transaction, lock, log, and two-phase commit.



1981, Edgar F. Codd, contributions in Relational DBMS.

1972, Charles W. Bachman, contributions in Network DBMS.



1.1 What should we learn?
- Three Aspects to Study Database

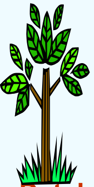
1) **Modeling and design of databases.**

Get (abstract) data models from real world, then translate them into the forms suitable for the target DBMS (*Database Management System*) - *tables, views*.

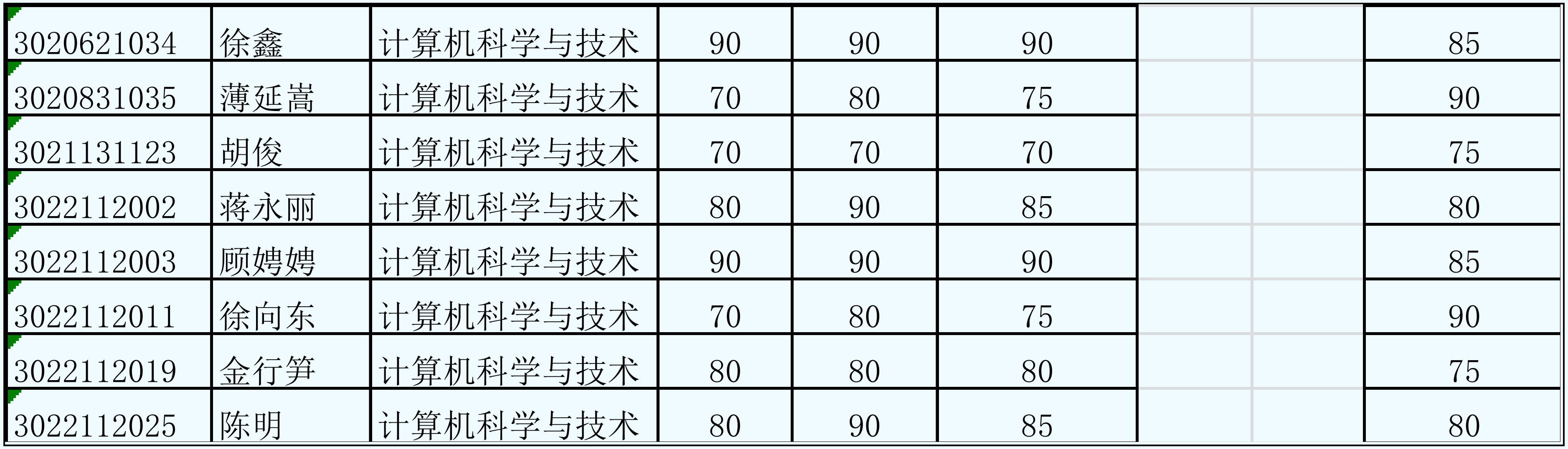
2) **Programming: use database - queries and update of data.**

SQL = “intergalactic data-speak”

3) **DBMS implementation - how does DBMS work, and how to design a DBMS.**



The Design of a Student Score Table



3020621034	徐鑫	计算机科学与技术	90	90	90			85
3020831035	薄延嵩	计算机科学与技术	70	80	75			90
3021131123	胡俊	计算机科学与技术	70	70	70			75
3022112002	蒋永丽	计算机科学与技术	80	90	85			80
3022112003	顾娉娉	计算机科学与技术	90	90	90			85
3022112011	徐向东	计算机科学与技术	70	80	75			90
3022112019	金行笋	计算机科学与技术	80	80	80			75
3022112025	陈明	计算机科学与技术	80	90	85			80

Is this table design good or bad? Why?

Another Design

Students

Sid	Sname	Ssex	Sage	Specialty
3023001093	黄毅照	M	21	No
3011112340	周朝威	F	20	Cs
3020621034	徐鑫	M	18	Cs
3020831035	薄延嵩	M	19	Cs
3021131123	胡俊	F	22	Cs

Courses

cid	Cname	credit
1	DB	4
2	OS	5
3	English	4
4	Math	4

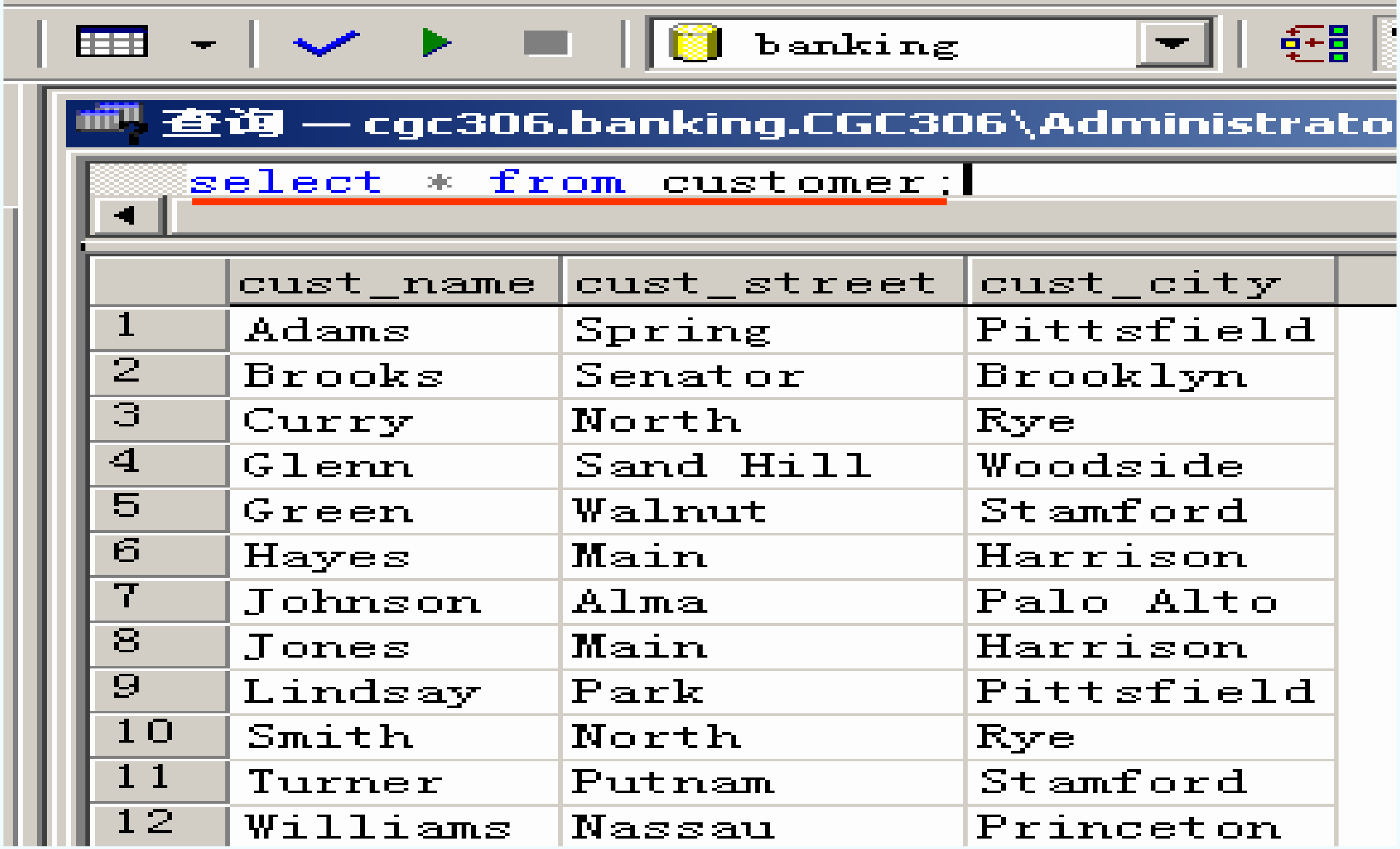
Enrolled

sid	cid	grade1	grade2	grade3
3023001093	1	90		
3023001093	2	85		
3020621034	1	90		
3020831035	1	75		
3021131123	2	75		

Is this table design good or bad? Why?

方法 1：利用数据库管理系统提供的交互工具访问数据库

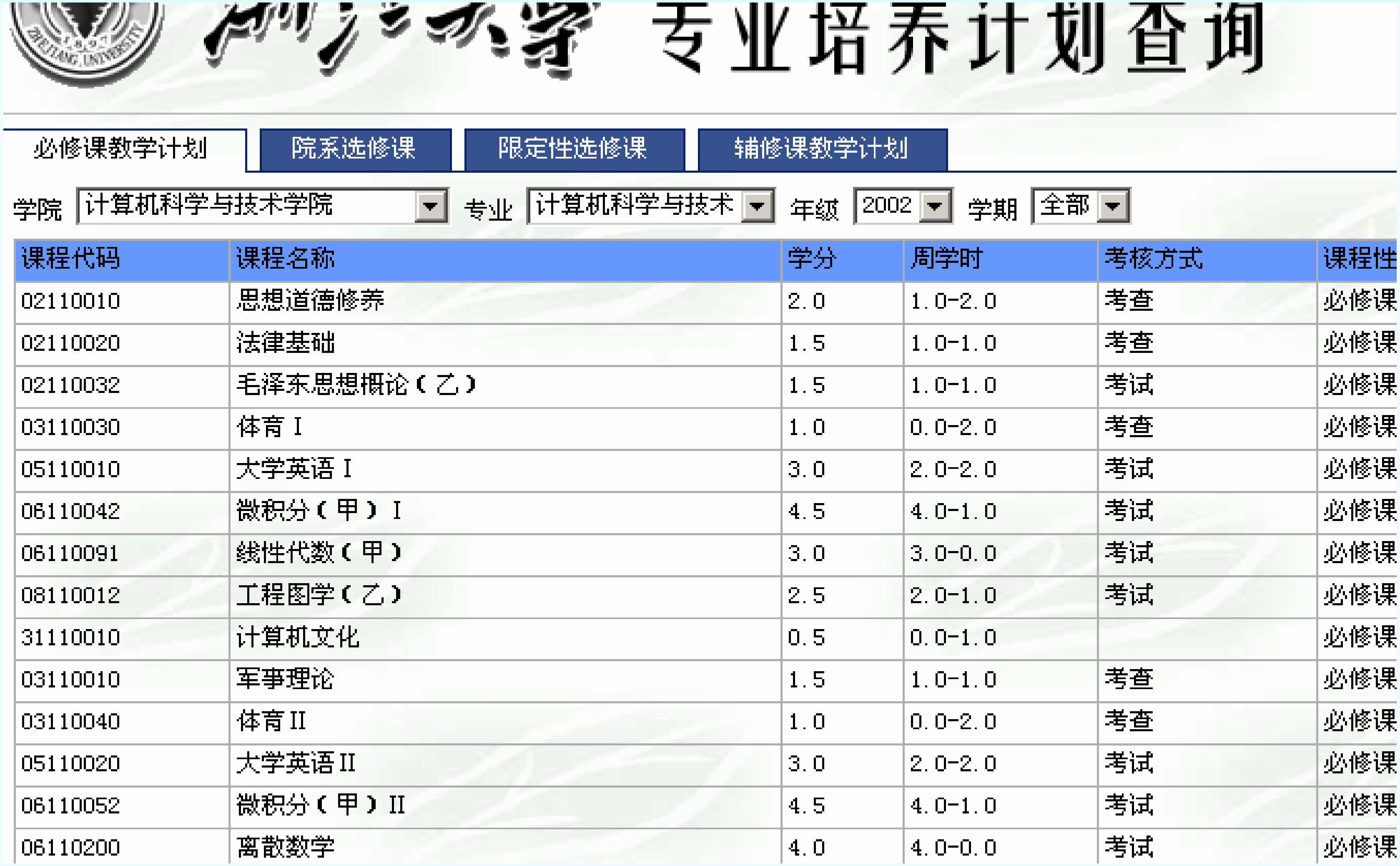
如： SQL Server 的查询分析器， ORACLE 的 Sql*Plus ， Work Sheet



```
select * from customer;
```

	cust_name	cust_street	cust_city
1	Adams	Spring	Pittsfield
2	Brooks	Senator	Brooklyn
3	Curry	North	Rye
4	Glenn	Sand Hill	Woodside
5	Green	Walnut	Stamford
6	Hayes	Main	Harrison
7	Johnson	Alma	Palo Alto
8	Jones	Main	Harrison
9	Lindsay	Park	Pittsfield
10	Smith	North	Rye
11	Turner	Putnam	Stamford
12	Williams	Nassau	Princeton

方法 2：利用开发工具设计界面、处理数据，调用 ODBC 访问数据库，如 ASP, JSP, VC++, PHP, PowerBuilder, Delphi



学院 计算机科学与技术学院 专业 计算机科学与技术 年级 2002 学期 全部

课程代码	课程名称	学分	周学时	考核方式	课程性
02110010	思想道德修养	2.0	1.0-2.0	考查	必修课
02110020	法律基础	1.5	1.0-1.0	考查	必修课
02110032	毛泽东思想概论（乙）	1.5	1.0-1.0	考试	必修课
03110030	体育Ⅰ	1.0	0.0-2.0	考查	必修课
05110010	大学英语Ⅰ	3.0	2.0-2.0	考试	必修课
06110042	微积分（甲）Ⅰ	4.5	4.0-1.0	考试	必修课
06110091	线性代数（甲）	3.0	3.0-0.0	考试	必修课
08110012	工程图学（乙）	2.5	2.0-1.0	考试	必修课
31110010	计算机文化	0.5	0.0-1.0		必修课
03110010	军事理论	1.5	1.0-1.0	考查	必修课
03110040	体育Ⅱ	1.0	0.0-2.0	考查	必修课
05110020	大学英语Ⅱ	3.0	2.0-2.0	考试	必修课
06110052	微积分（甲）Ⅱ	4.5	4.0-1.0	考试	必修课
06110200	离散数学	4.0	4.0-0.0	考试	必修课

1.2 What is a database
---Definition

- **Database:**
  - **A collection of interrelated data, relevant to an enterprise.**
  - **A large collection of integrated and persistent data (DB). [R. Ramakrishnan, J. Gehrhe]**
  - **A collection of information that exists over a long period of time, often many years. [Ullman]**
  - 长期存储在计算机内、有组织的、可共享的数据集合。**[** 萨师煊，王珊 **]**

- **Database Management System (DBMS):**

(Database) + A set of programs used to access, update and manage the data in database.

Major properties of DBMS

- **Efficiency** and **scalability** (可扩展性) in data access;
- **Reduced** application development time;
- Data **independence** (physical data independence / logical data independence) ;
- Data **integrity** (完整性) and **security**;
- **Concurrent access** and **robustness** (recovery).

1.3 DBMS—A Historical Perspective

- **File processing system (1950s-1960s)**
- **Network and hierarchical DBMS (1960s-1970s)**
  - 网状数据模型、层次数据模型 - 网状数据库、层次数据库 (结构复杂、使用很困难)
- **Relational database systems (RDBMS)**
  - **Relational model (1970, E.F. Codd)**
  - **Relational database system developed (late 1970s)**
  - **Relational database systems on the market (1980s)**
  - **Matured relational DBMS technology (1990s)**

DBMS—A Historical Perspective

- **Object-oriented database system (OODBMS)**
- **Object-relational database systems (ORDBMS)**
- **Application-oriented database systems**
  - **Spatial, temporal, multimedia, Web databases**
- **Data Warehousing** (数据仓库)**, Online Analytical Processing** (联机分析处理)**, and Data Mining** (数据挖掘) **systems (for data analysis)**

Database systems VS File Processing Systems

- **File processing system supported by a conventional OS:**
  - **New application programs must be written when needed, and new data files are created as required.**
  - **But over a long period of time, data files may be in different formats. Data files are independent of each other.**
- **Drawbacks of using file systems to store data:**
  - **Data redundancy and inconsistency**
    - **Multiple file formats, duplication of information in different files.**

Database system versus File Processing Systems (cont.)

- **Drawbacks of using file systems (cont.)**
  - **Difficulty in accessing data**
    - **Need to write a new program to carry out each new task.**
  - **Data isolation**
    - **multiple files and multiple formats. (Difficult to retrieve, difficult to share)**
  - **Integrity problems**
    - **Integrity constraints (e.g., account balance > 0) become part of program code.**
    - **Hard to add new constraints or change existing ones.**

- **Drawbacks of using file systems (cont.)**
  - **No atomicity of updates (consistent)**
    - **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 – atomicity.**
  - **Difficult to concurrent access by multiple users**
    - **Concurrent accessed needed for performance.**
    - **Uncontrolled concurrent accesses can lead to inconsistencies.**
      - ✓ **E.g., two people reading a balance and updating it at the same time.**
  - **Security problems (Right person use right data)**
- **Database systems offer solutions to all the above problems!**

The DBMS Marketplace

- **Relational DBMS companies :**
  - **Oracle, Sybase - are among the largest database software companies in the world.**
  - **IBM DB2 - is by some accounts the largest DBMS vendor in the world.**
  - **Microsoft SQL-Server, plus Microsoft Access for the cheap DBMS on the desktop, answered by "lite" systems from other competitors.**
- **Relational companies also challenged by "object-oriented DB" companies.**
- **But countered with "object-relational" systems, which retain the relational core while allowing type extension as in OO systems.**
- **Other database products: Ingres, Paradox, Foxbase, FoxPro, dBase,…**

The DBMS Marketplace (cont.)

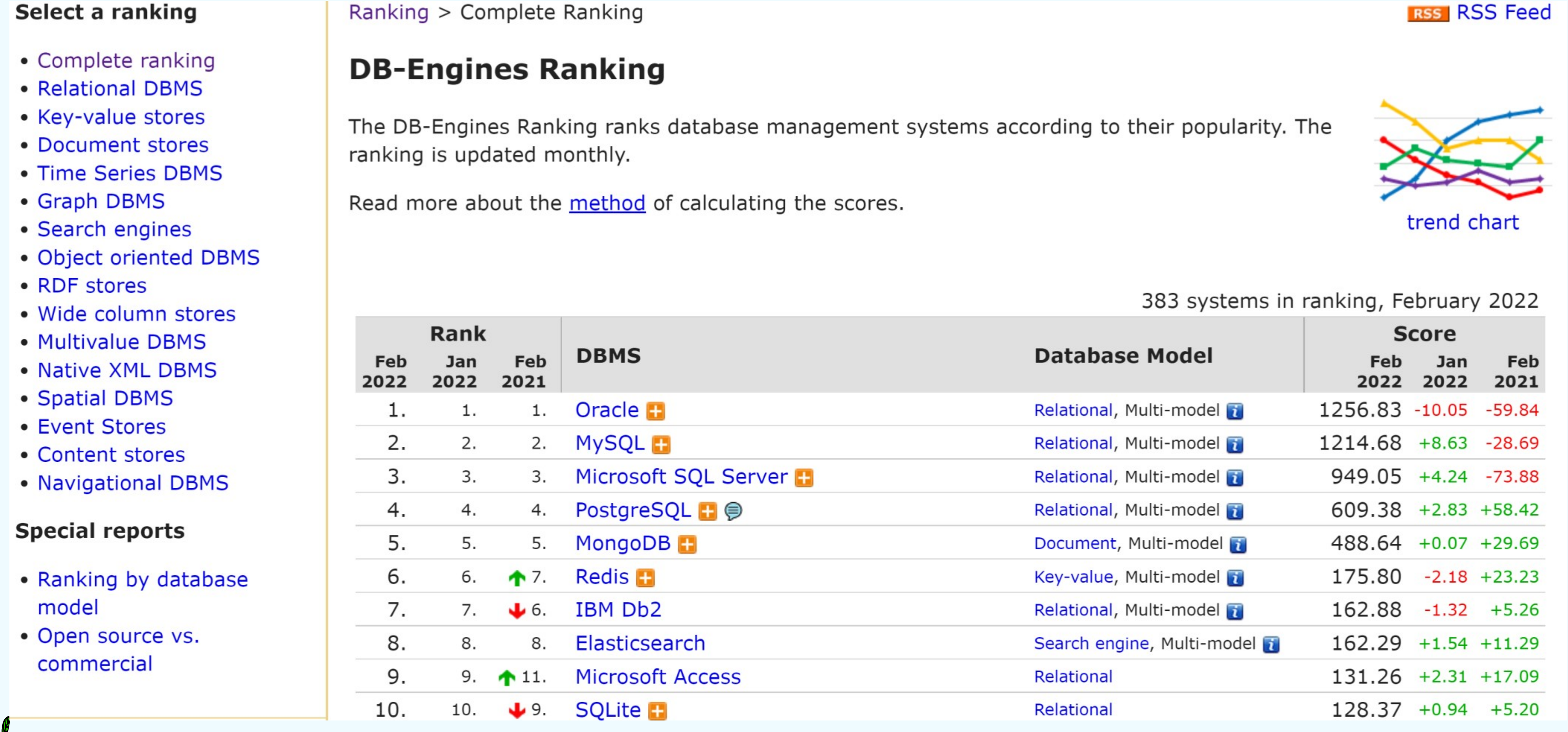
Open source database:

- **MySQL: is the most popular open source database for small system on web sites.**
  - **MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), a fast growing open source enterprise software stack. http://www.mysql.com**
- **PostgreSQL: is a highly scalable, open source object-relational database management system. http://www.postgresql.org**
  - **Originally developed by CS Dept of UC Berkeley called ‘Postgres’**

The DBMS Marketplace (cont.)

Some useful sites:

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



Select a ranking

- Complete ranking
- Relational DBMS
- Key-value stores
- Document stores
- Time Series DBMS
- Graph DBMS
- Search engines
- Object oriented DBMS
- RDF stores
- Wide column stores
- Multivalue DBMS
- Native XML DBMS
- Spatial DBMS
- Event Stores
- Content stores
- Navigational DBMS

Special reports

- Ranking by database model
- Open source vs. commercial

Ranking > Complete Ranking

RSS RSS Feed

DB-Engines Ranking

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

Read more about the method of calculating the scores.

trend chart

383 systems in ranking, February 2022

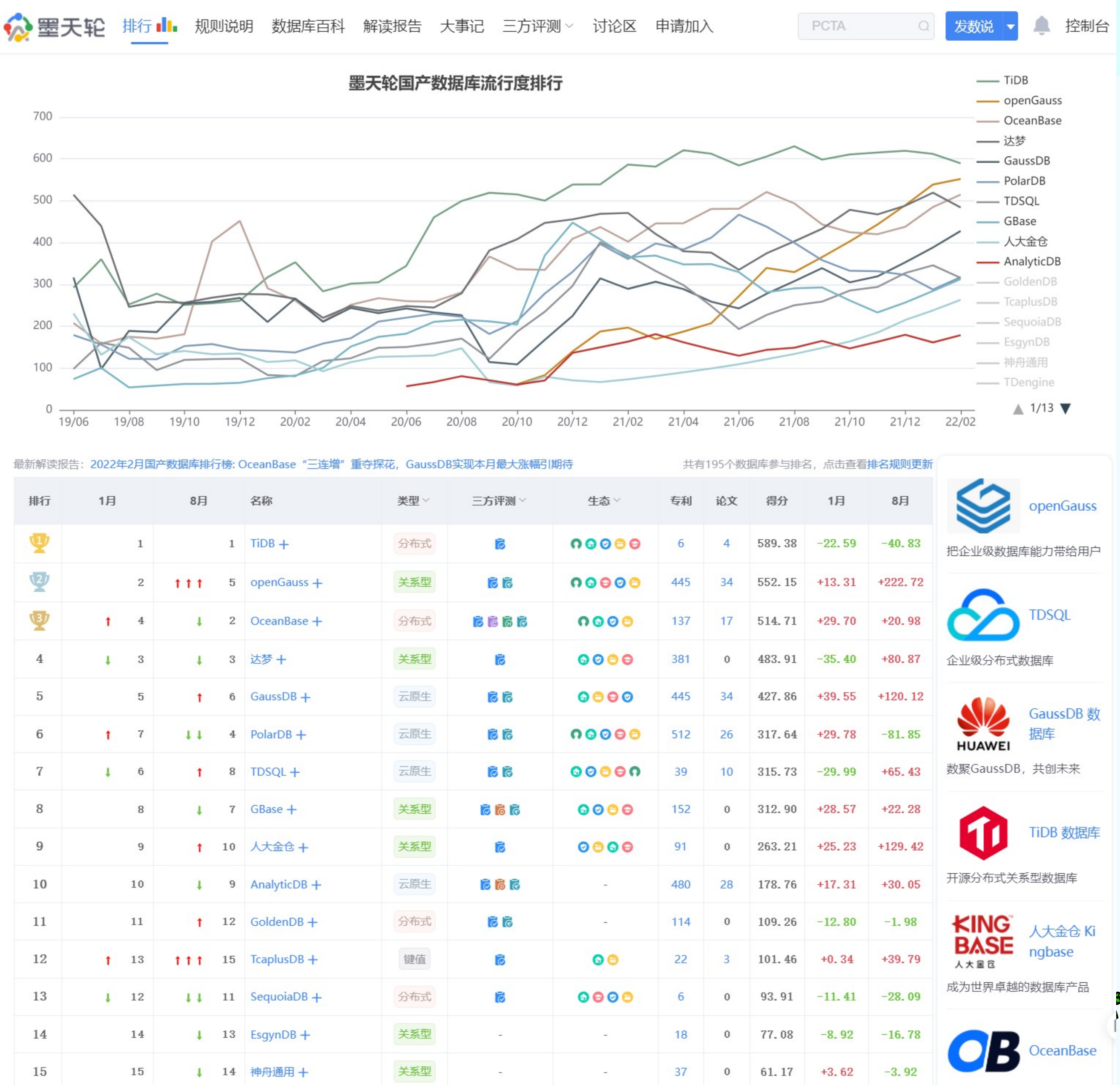
Rank			DBMS	Database Model	Score		
Feb 2022	Jan 2022	Feb 2021			Feb 2022	Jan 2022	Feb 2021
1.	1.	1.	Oracle	Relational, Multi-model	1256.83	-10.05	-59.84
2.	2.	2.	MySQL	Relational, Multi-model	1214.68	+8.63	-28.69
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	949.05	+4.24	-73.88
4.	4.	4.	PostgreSQL	Relational, Multi-model	609.38	+2.83	+58.42
5.	5.	5.	MongoDB	Document, Multi-model	488.64	+0.07	+29.69
6.	6.	↑ 7.	Redis	Key-value, Multi-model	175.80	-2.18	+23.23
7.	7.	↓ 6.	IBM Db2	Relational, Multi-model	162.88	-1.32	+5.26
8.	8.	8.	Elasticsearch	Search engine, Multi-model	162.29	+1.54	+11.29
9.	9.	↑ 11.	Microsoft Access	Relational	131.26	+2.31	+17.09
10.	10.	↓ 9.	SQLite	Relational	128.37	+0.94	+5.20

By 2022/02/22

The DBMS Marketplace (cont.)

https://www.modb.pro/dbRank

By 2022/02/22



最新解读报告：2022年2月国产数据库排行榜: OceanBase "三连增" 重夺探花，GaussDB实现本月最大涨幅引期待

共有195个数据库参与排名，点击查看排名规则更新

排行	1月	8月	名称	类型	三方评测	生态	专利	论文	得分	1月	8月
1	1	1	TiDB +	分布式	●	●●●●●	6	4	589. 38	-22. 59	-40. 83
2	2	↑↑↑ 5	openGauss +	关系型	●●	●●●●●	445	34	552. 15	+13. 31	+222. 72
3	↑ 4	↓ 2	OceanBase +	分布式	●●●●	●●●●	137	17	514. 71	+29. 70	+20. 98
4	↓ 3	↓ 3	达梦 +	关系型	●	●●●●	381	0	483. 91	-35. 40	+80. 87
5	5	↑ 6	GaussDB +	云原生	●●	●●●●	445	34	427. 86	+39. 55	+120. 12
6	↑ 7	↓↓ 4	PolarDB +	云原生	●●	●●●●●	512	26	317. 64	+29. 78	-81. 85
7	↓ 6	↑ 8	TDSQL +	云原生	●●	●●●●●	39	10	315. 73	-29. 99	+65. 43
8	8	↓ 7	GBase +	关系型	●●●	●●●●	152	0	312. 90	+28. 57	+22. 28
9	9	↑ 10	人大金仓 +	关系型	●	●●●●	91	0	263. 21	+25. 23	+129. 42
10	10	↓ 9	AnalyticDB +	云原生	●●●	-	480	28	178. 76	+17. 31	+30. 05
11	11	↑ 12	GoldenDB +	分布式	●●	-	114	0	109. 26	-12. 80	-1. 98
12	↑ 13	↑↑↑ 15	TcaplusDB +	键值	●	●●	22	3	101. 46	+0. 34	+39. 79
13	↓ 12	↓↓ 11	SequoiaDB +	分布式	●	●●●●	6	0	93. 91	-11. 41	-28. 09
14	14	↓ 13	EsgynDB +	关系型	-	-	18	0	77. 08	-8. 92	-16. 78
15	15	↓ 14	神舟通用 +	关系型	-	-	37	0	61. 17	+3. 62	-3. 92

openGauss
把企业级数据库能力带给用户

TDSQL
企业级分布式数据库

HUAWEI GaussDB 数据库
数聚GaussDB，共创未来

TiDB 数据库
开源分布式关系型数据库

KING BASE 人大金仓 人大金仓 Kingbase
成为世界卓越的数据库产品

OceanBase

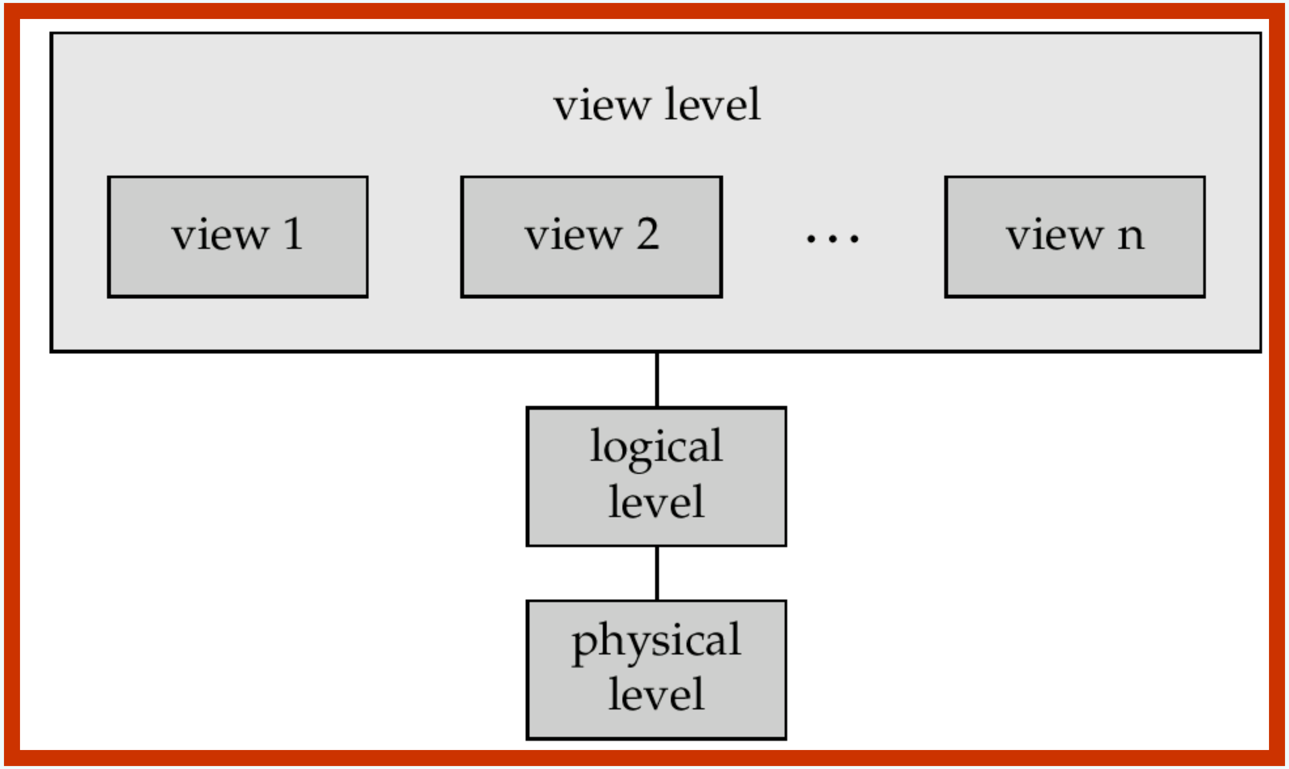
1.4 Levels of Data Abstraction

- **How to use DB - Different usage needs different level of abstraction. (e.g., student score management system)**
  - **Physical level: describes how a record is stored. Files**
  - **Logical level: describes data stored in database, and the relationships among the data on upper level.**

    comparative: **type *student* = record**
    ***sid*:**
    integer;
    ***sname*: string;**
    ***gender*: string;** ***age*:**
    integer; ***deptno*:**
    integer; **end;**
  - **View level: application programs hide details of data types. Views can also hide information (e.g., employee's salary) for security purposes.**

View of Data

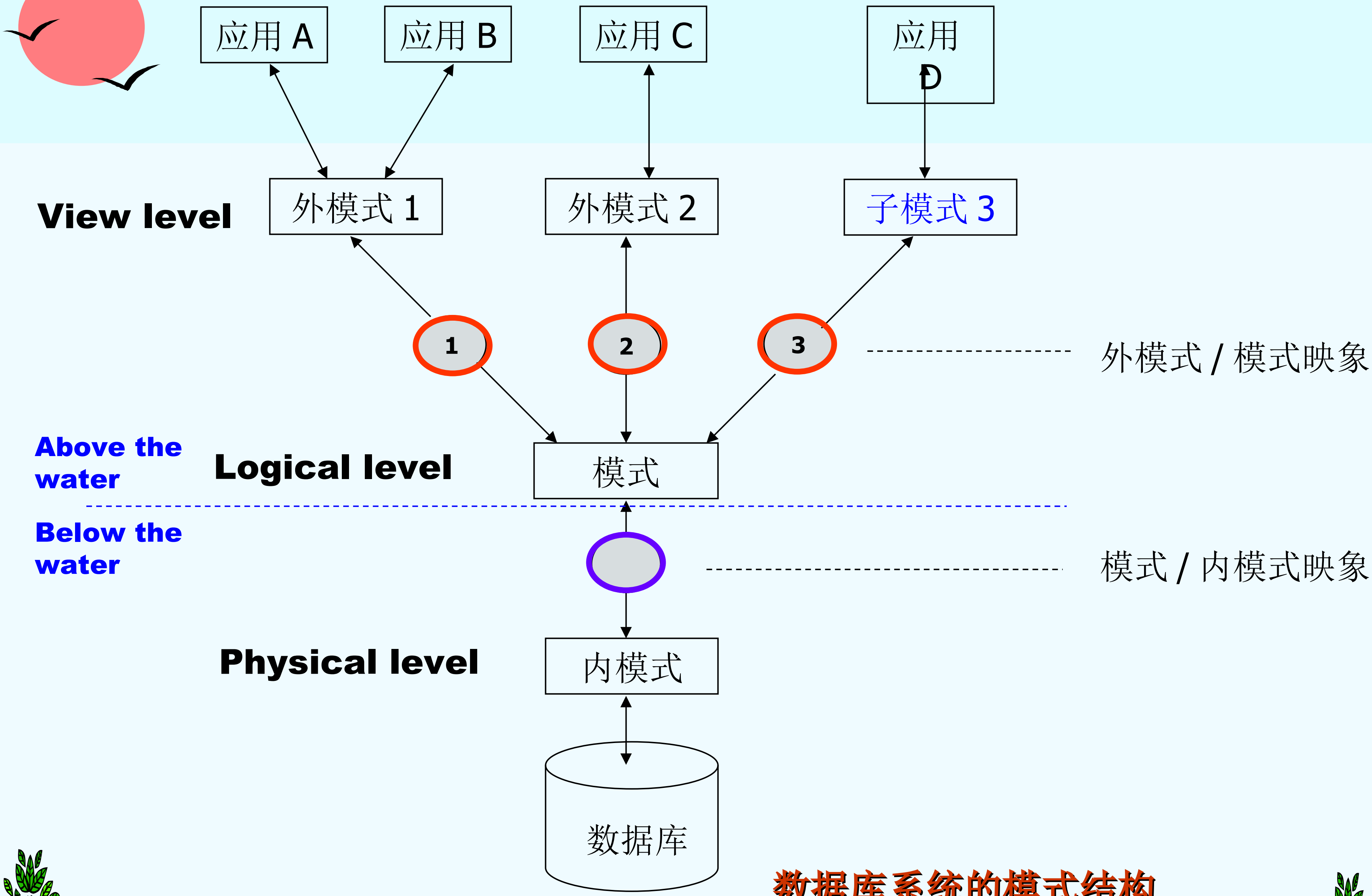
An architecture for a database system



Schemas (模式) and Instances (实例)

- Similar to types and variables in programming languages
  (type ↔ schema, variable ↔ instance)
- Schema - the structure of the database on different level
  - Analogous to type information of a variable in a program
  - Physical schema: database design of structure at the physical level
  - Logical schema: database design of structure at the logical level
  - Subschema: schema at view level
- Instance - the actual content of the database at a particular point in time
  - Analogous to the value of a variable

[schema: a representation of an arrangement or plan.]



应用 A

应用 B

应用 C

应用 D

View level

外模式 1

外模式 2

子模式 3

1

2

3

外模式 / 模式映象

Above the water

Logical level

模式

Below the water

模式 / 内模式映象

Physical level

内模式

数据库

数据库系统的模式结构

Physical Independence vs. Logical independence

- Ability to modify a scheme definition at one level w.o. affecting a scheme definition at a higher level.
- Physical Data Independence - The ability to modify the physical schema without changing the logical schema.
  - Applications depend on the logical schema.
  - Applications insulated from how data is structured and stored.
  - One of the most important benefits of using a DBMS!
- Logical Data Independence - Protect application programs from changes in logical structure of data.
  - Logical data independence is harder to achieve as the application programs are heavily dependent on the logical structure of data.

1.5 Data Models

- **Data model is a collection of conceptual tools for describing:**
  - **data structure**
  - **data relationships**
  - **data semantics**
  - **data constraints**
- **Different data models:**
  - **Entity-Relationship model**
  - **Relational model**
  - **Other models:**
    - **object-oriented model**
    - **semi-structured data models (XML)**
    - **older models: network model and hierarchical model** …

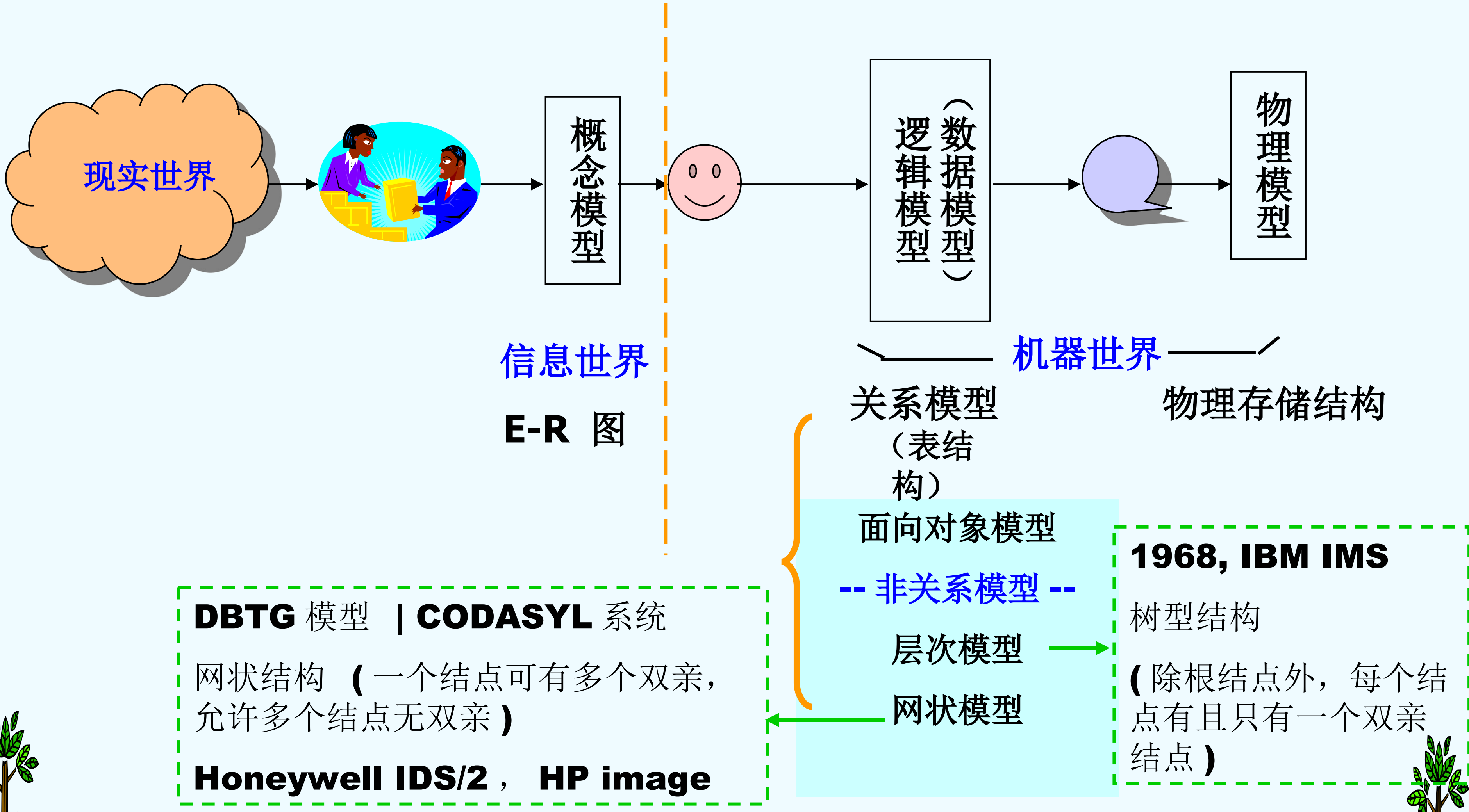
Database design process:

- **Requirement analysis**
- **Conceptual design**
- **Logical design**
- **……**

Steps of Database Design

1) **Requirement analysis**
   - What data, applications, and operations needed
2) **Conceptual database design**
   - A high-level description of data, constraints using E - R model or a similar high level data model
3) **Logical database design:**
   - Convert the conceptual design into a DB schema
4) **Schema refinement**
   - Normalization of relations: Check relational schema for redundancies and related anomalies.
5) **Physical database design :**
   - Indexing, clustering and database tuning
6) **Create and initialize the database & Security design**
   - Load initial data, testing
   - Identify different user groups and their roles

数据库的设计步骤



DBTG 模型 | **CODASYL** 系统

网状结构 **(** 一个结点可有多个双亲，允许多个结点无双亲 **)**

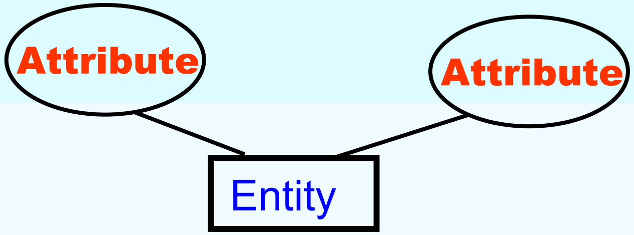
Honeywell IDS/2 ， **HP image**

1968, IBM IMS

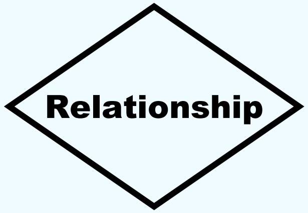
树型结构

(除根结点外，每个结点有且只有一个双亲结点 **)**

(1) Entity Relationship Model (实体 - 联系 模型)



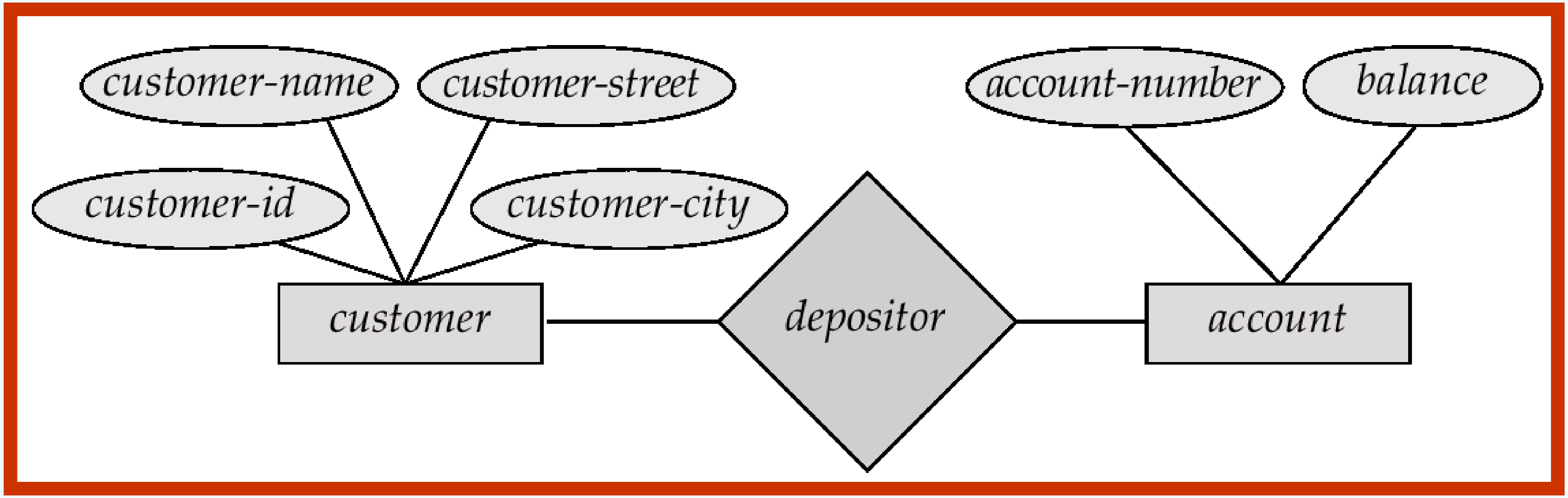
- **E-R model of real world**
  - **Entities (objects)**
    - **E.g., customers, accounts, bank branch**
    - **Entities are described by attributes**
  - **Relationships between entities**
    - **E.g., Account A-101 is held by customer Johnson**
    - **Relationship set depositor associates customers with accounts**



- **E-R Model is widely used for database design**
  - **Database design in E-R model usually converted to design in the relational model .**
  - **Was first proposed by Pete Chen.**

Entity-Relationship Model (cont.)

Example of schema in the entity-relationship model



(2) Relational Model

Transfer E-R diagrams into relational schema

- **Example of tabular data in the relational model**

Attributes

Customer:

schema

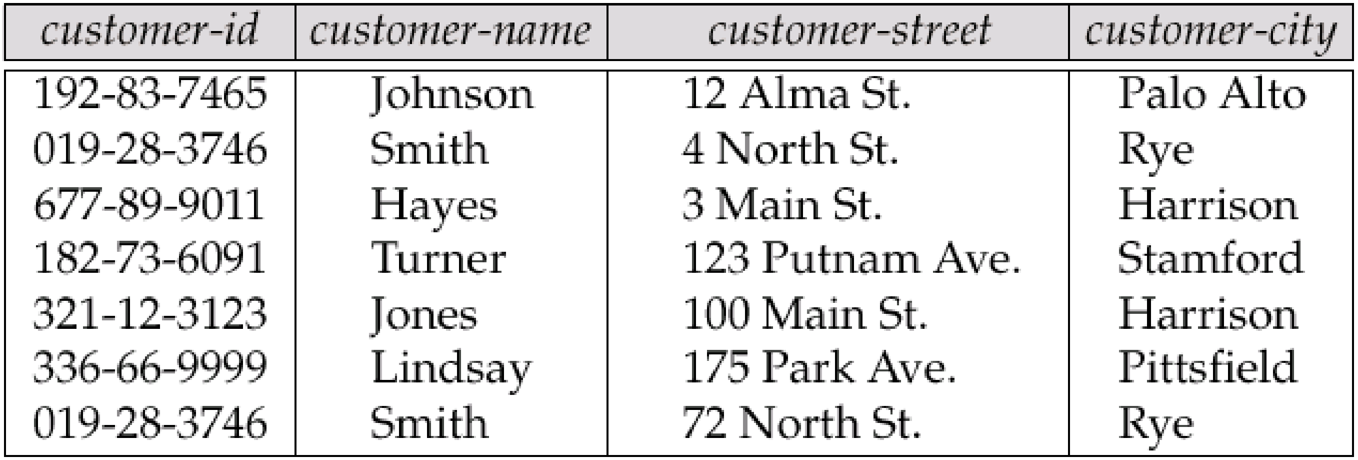
Customer-id	*customer-name*	*customer-street*	*customer-city*	*account-number*
192-83-7465	Johnson	Alma	Palo Alto	A-101
019-28-3746	Smith	North	Rye	A-215
192-83-7465	Johnson	Alma	Palo Alto	A-201
321-12-3123	Jones	Main	Harrison	A-217
019-28-3746	Smith	North	Rye	A-201

Tuple

元组

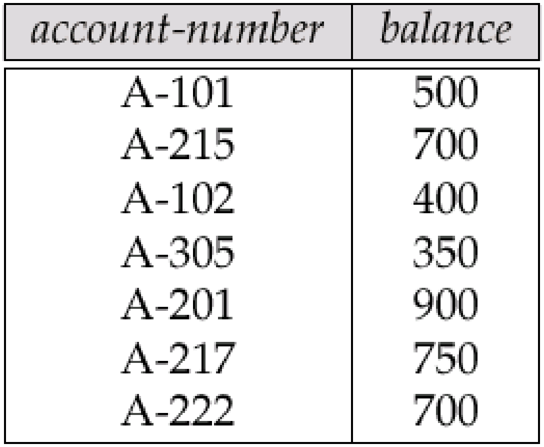
Column (field)

A Sample Relational Database



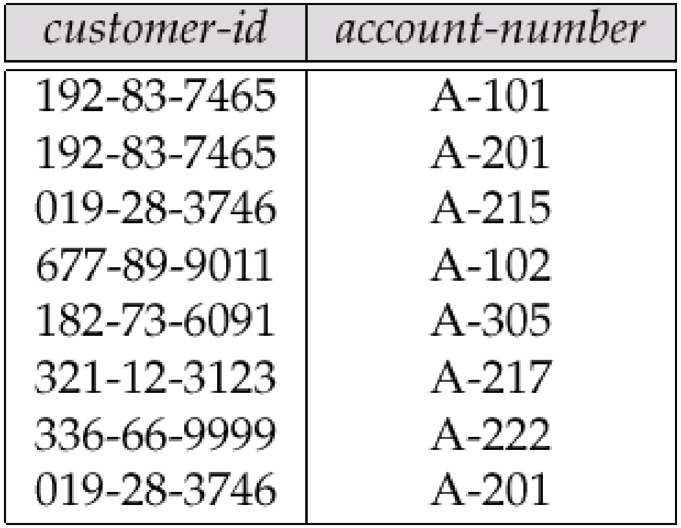
customer-id	*customer-name*	*customer-street*	*customer-city*
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table



account-number	*balance*
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table



customer-id	*account-number*
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

Another example: University Database

Students

Sid	Sname	Ssex	Sage	sdept
3023001093	Tom	M	21	Cs
3011112340	Mary	F	20	Cs
3020621034	Jack	M	18	Cs
3020831035	Smith	M	19	Ma
3021131123	Alane	F	22	Is

Courses

cid	Cname	credit
1	DB	4
2	OS	5
3	English	4
4	Math	4

Enrolled

sid	cid	grade
3023001093	1	92
3023001093	2	88
3020621034	1	70
3020831035	1	85
3021131123	2	95

成绩登记表

view level
(subschema)

Sid	Sname	Cname	credit	grade
3023001093	Tom	DB	4	92
3023001093	Tom	OS	5	88
3020621034	Jack	DB	4	70
3020831035	Smith	DB	4	85
3021131123	Alane	OS	5	95

Logical level (schema)

student

Sid	Sname	Ssex	Sage	sdept
3023001093	Tom	M	21	Cs
3011112340	Mary	F	20	Cs
3020621034	Jack	M	18	Cs
3020831035	Smith	M	19	Ma
3021131123	Alane	F	22	Is

course

cid	Cname	credit
1	DB	4
2	OS	5
3	English	4
4	Math	4

Enrolled

sid	cid	grade
3023001093	1	92
3023001093	2	88
3020621034	1	70
3020831035	1	85
3021131123	2	95

1.6 Database Language

Database Language:

- **Data Definition Language (DDL, 数据定义语言)**
- **Data Manipulation Language (DML, 数据操纵语言)**
- **Data Control Language (DCL, 数据控制语言)**

(1) Data Definition Language (DDL)

- **Specifies a database scheme as a set of definitions of relational schema.**
- **Also specifies storage structure, access methods and consistency constraints.**
- **DDL statements are compiled, resulting in a set of tables stored in a special file: data dictionary (数据字典), which contains metadata (元数据).**
- **E.g.,**

```
CREATE TABLE account (
                account_number char(10),
                balance  integer);
```

该 SQL 语句创建了表 account

(1) Data Definition Language (cont.)

- **Data dictionary** (数据字典) contains metadata (i.e., the data about data) about:
  - Database schema
  - Data storage structure
  - Access methods and constraints
  - Statistical information
  - Authorization

(2) Data Manipulation Language (DML)

- Data Manipulation Language (DML)
  - **Retrieve data** from the database
  - **Insert / delete / update** data in the database
  - DML also known as **query language**
- Two classes of languages
  - **Procedural** – user specifies what data is required and how to get those data (C, Pascal, Java, …)
  - **Nonprocedural** – user specifies what data is required without specifying how to get those data (SQL, Prolog)

(3) SQL

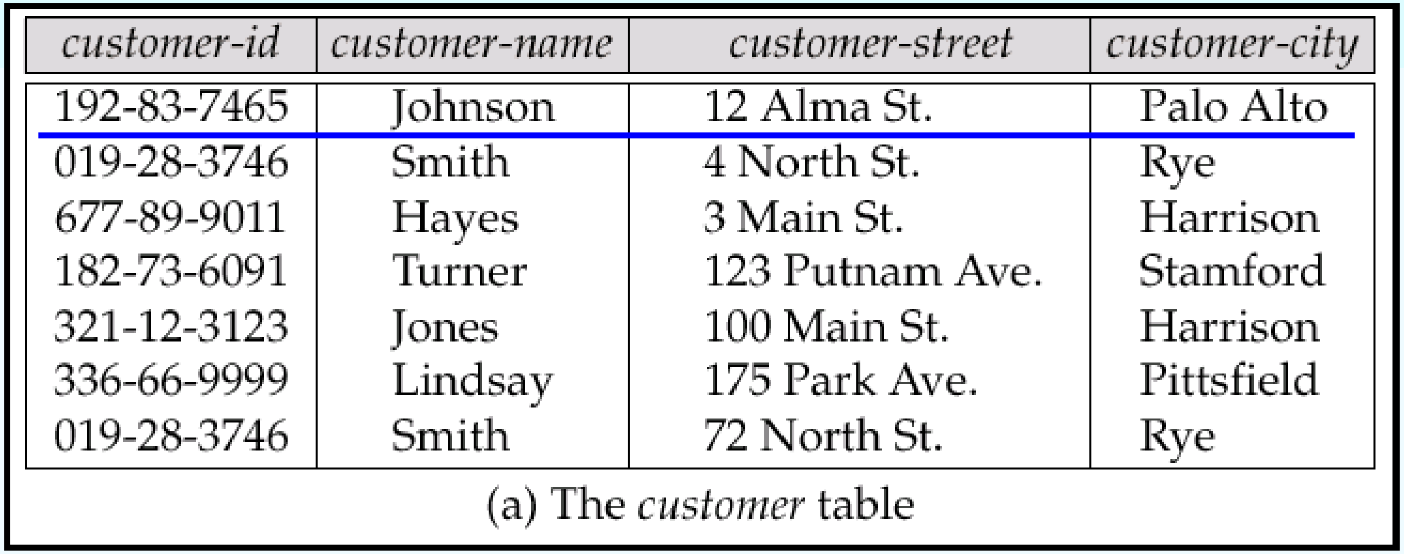
- **SQL= DDL+ DML+DCL**
- **SQL has been widely used.**
  - **SQL (Structured Query Language, 结构化查询语言), 1975. IBM. System R. Called "SEQUEL" (Structured English QUEry Language).**
  - **E.g., find the name of the customer with customer-id 192-83-7465**

```
SELECT customer-name
FROM customer
WHERE customer-id = '192-83-7465'
```

  - **E.g., find the balances of all accounts held by the customer with customer-id 192-83-7465**

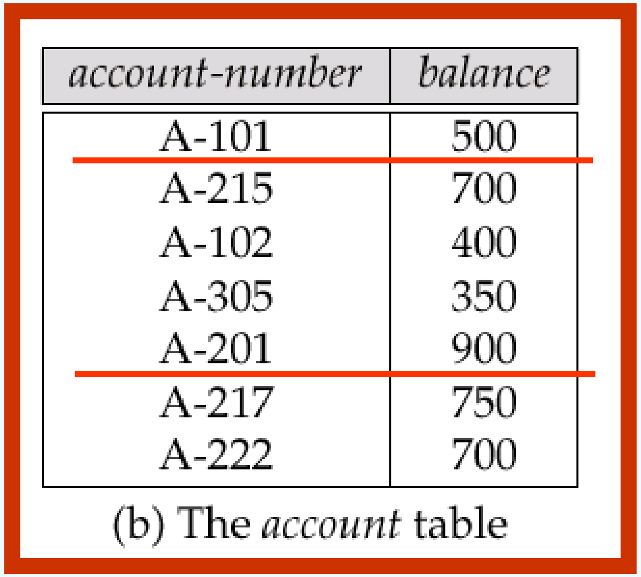
```
SELECT account.balance
FROM depositor, account
WHERE depositor.customer-id = '192-83-7465' and
          depositor.account-number = account.account-
number
```

A Sample Relational Database



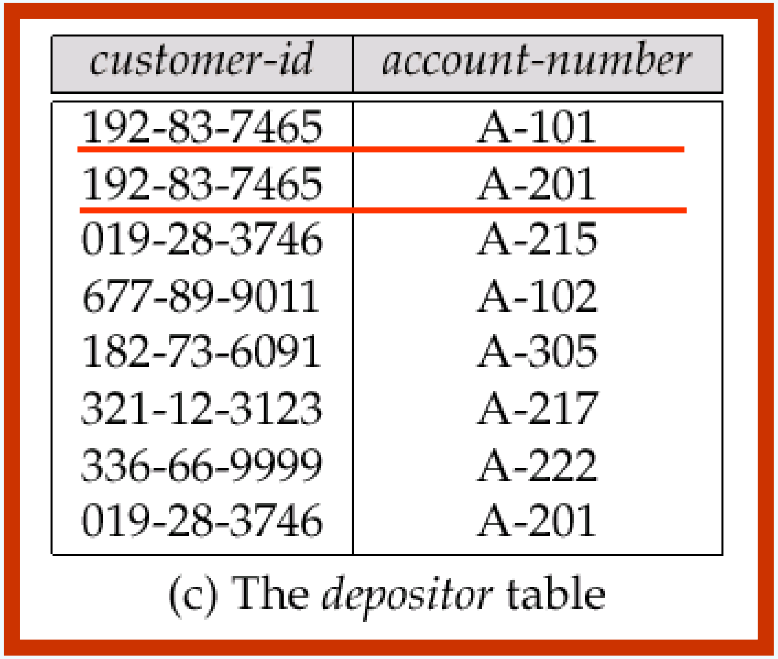
customer-id	*customer-name*	*customer-street*	*customer-city*
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table



account-number	*balance*
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table



customer-id	*account-number*
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

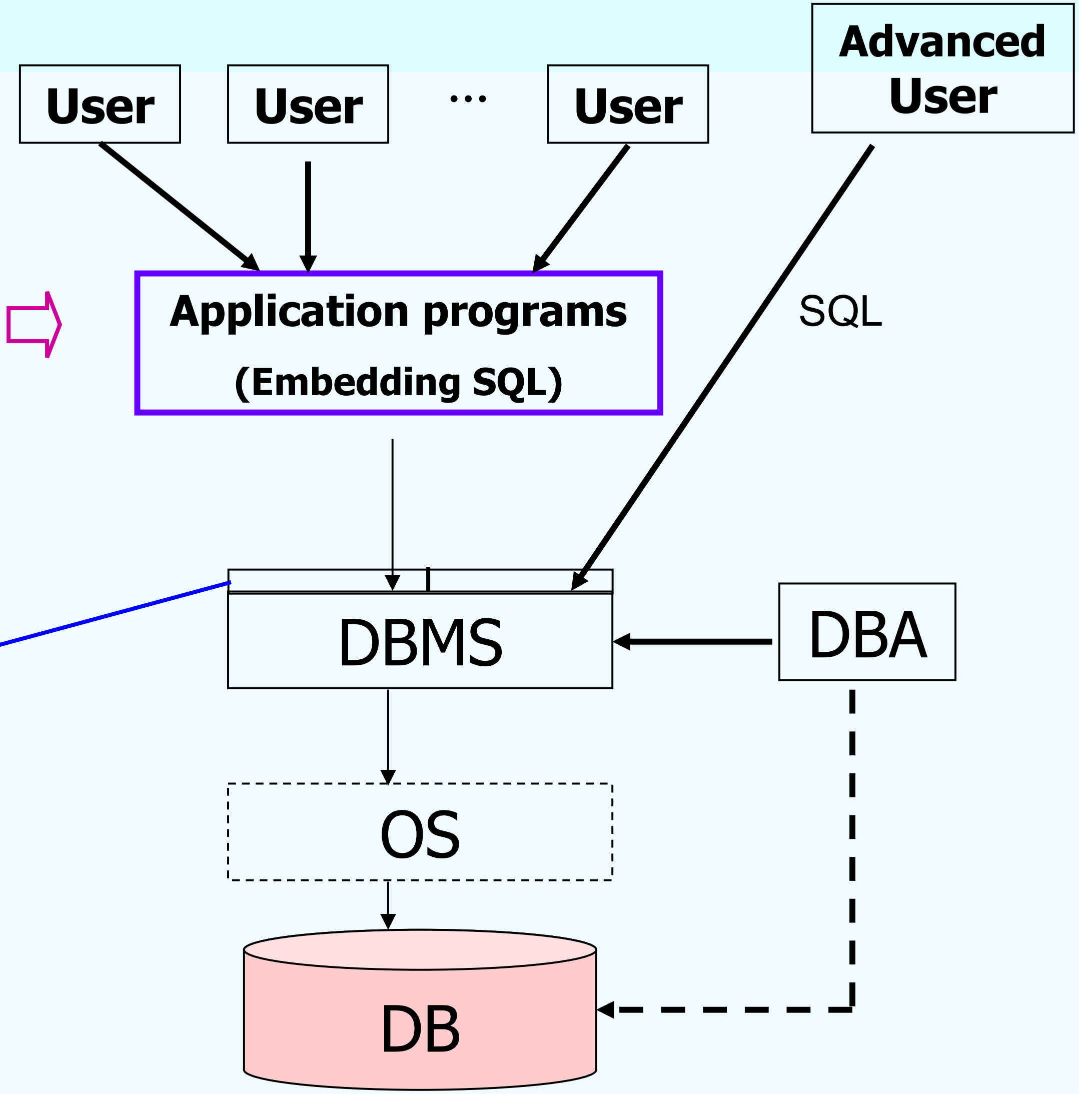
(3) SQL (cont.)

- **SQL is the most widely used query language . There are three kinds of usage:**
  - **Use it directly in the interactive environment**
    - **SQL Server: Query Analyzer (查询分析器)**
    - **Oracle: Sql*Plus, Work Sheet**
    - **MySQL: mysql command line client**
  - **Use it by host language through ODBC (Open DataBase Connectivity), JDBC**
  - **Use it by host language with embedded-SQL**

(4) How to use a database

Developed by C, C++, Fortran, Cobol, Java

Delphi, VB, PowerBuilder,



1.7 Database Users

- **Users are differentiated by the way they expect to interact with the system**
- **Naïve users - invoke one of the permanent application programs that have been written previously by a high level language. (** 普通用户 **)**
  - **E.g., people accessing database over the web, bank tellers, clerical staff**
- **Application programmers - interact with system through SQL calls.**
- **Sophisticated users - form requests in a database query language. E.g online analytical processing (OLAP), Data mining. (** 富有经验的用户 **)**
- **Specialized users - write specialized database applications that do not fit into the traditional data processing framework. E.g CAD, KDB, ES. (** 专门用户，特殊用户 **)**

专业用户

1.8 Database Administrator

- **Database administrator** (DBA): A special user having central control over database and programs accessing that data.
- DBA has the highest privilege for the database.
- DBA coordinates all the activities of the database system.
- DBA controls all users authority to the database.
- DBA has a good understanding of the enterprise's information resources and needs.

1.8 Database Administrator (cont.)

- **Database administrator's duties include:**
  - **Schema definition**
  - **Storage structure and access method definition**
  - **Schema and physical organization modification**
  - **Granting of authorization for data access**
  - **Routing maintenance**
    - **Monitoring performance and responding to changes in requirements**
    - **Security for the database (e.g., periodically backup database, recovery when failure)**

1.9 Transaction Management（事务管理）

- **Concurrent use is important. But cause problems.**
- **A transaction is a collection of operations that performs a single logical function in a database application - Jim Gray**
- **Transaction requirement: atomicity（原子性）, consistence（一致性）, isolation（隔离性）, durability（持久性）/ ACID**
- **Transaction-management component ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures. - By backup and recovery subsystem**
- **Concurrency-control manager（并发控制管理器）controls the interaction among the concurrent transactions.**

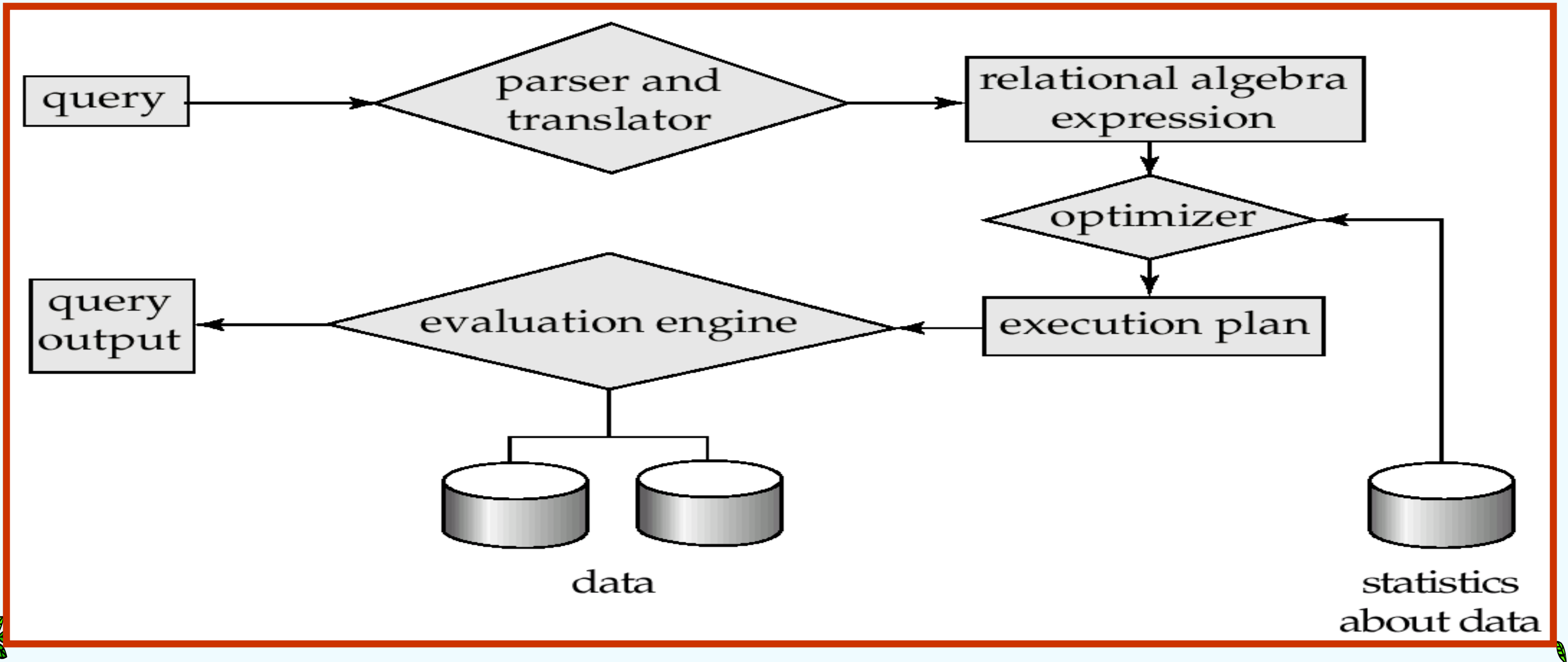
1.10 Database management system structure

1) Storage manager

- Provides interface between the low-level data stored in the database (file system) and the application programs and queries submitted to the system.
- Responsible for efficient storing, retrieving, updating data in the database.
- Includes:
  - Transaction manager
  - Authorization and integrity manger
  - File manager (interaction with the file system to process data files, data dictionary, index files)
  - Buffer manager

2) Query processor

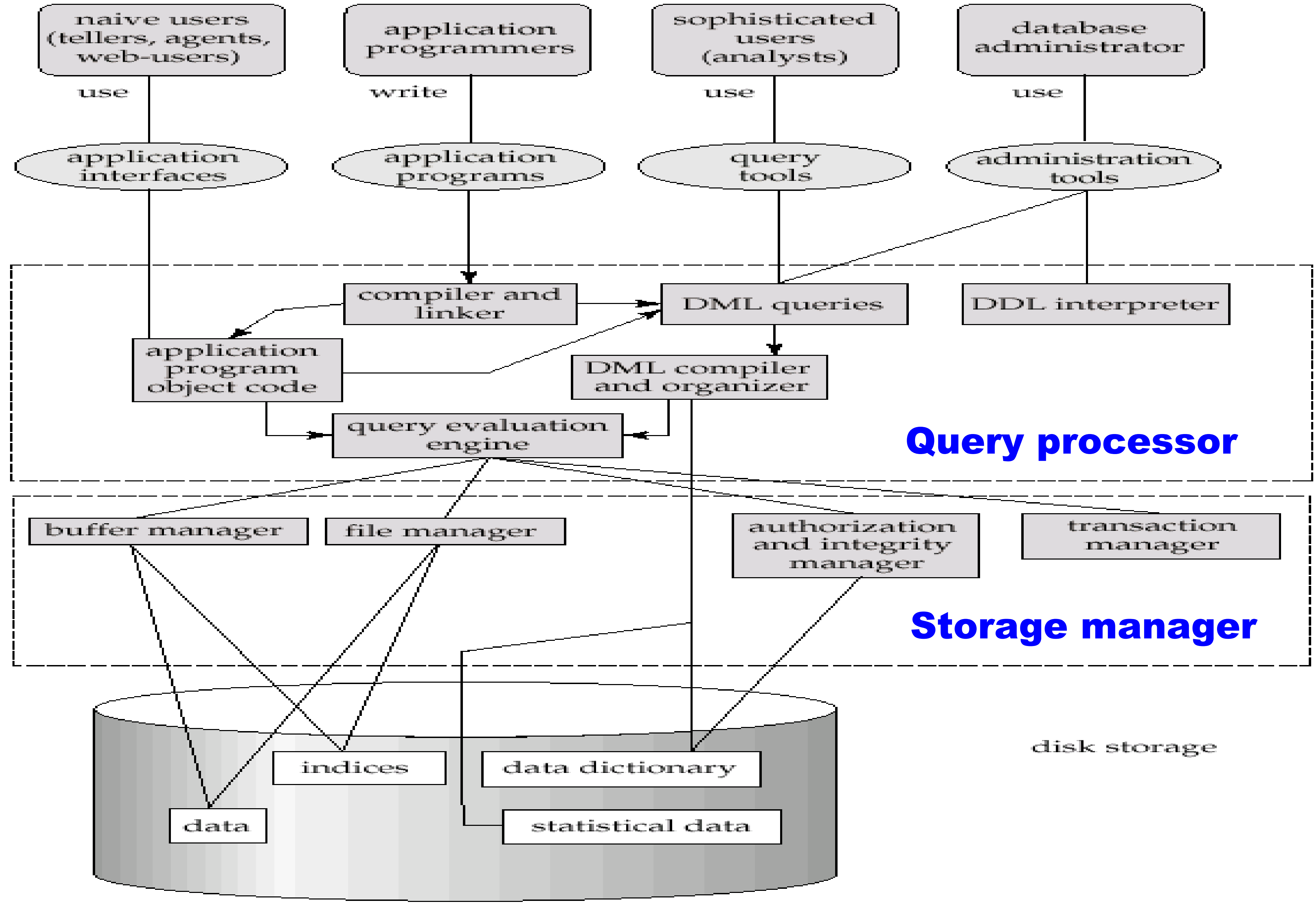
- Includes: DDL interpreter, DML compiler, query processing
  - Parsing and translation
  - Optimization
  - Evaluation



Query Processing (cont.) - Optimization

- Alternative ways of evaluating a given query
  - Equivalent expressions
  - Different algorithms for each operation
- Cost difference between a good and a bad way of evaluating a query can be enormous
- Need to estimate the cost of operations
  - Depends critically on statistical information about relations which the database must maintain
  - Need to estimate statistics for intermediate results to compute cost of complex expressions

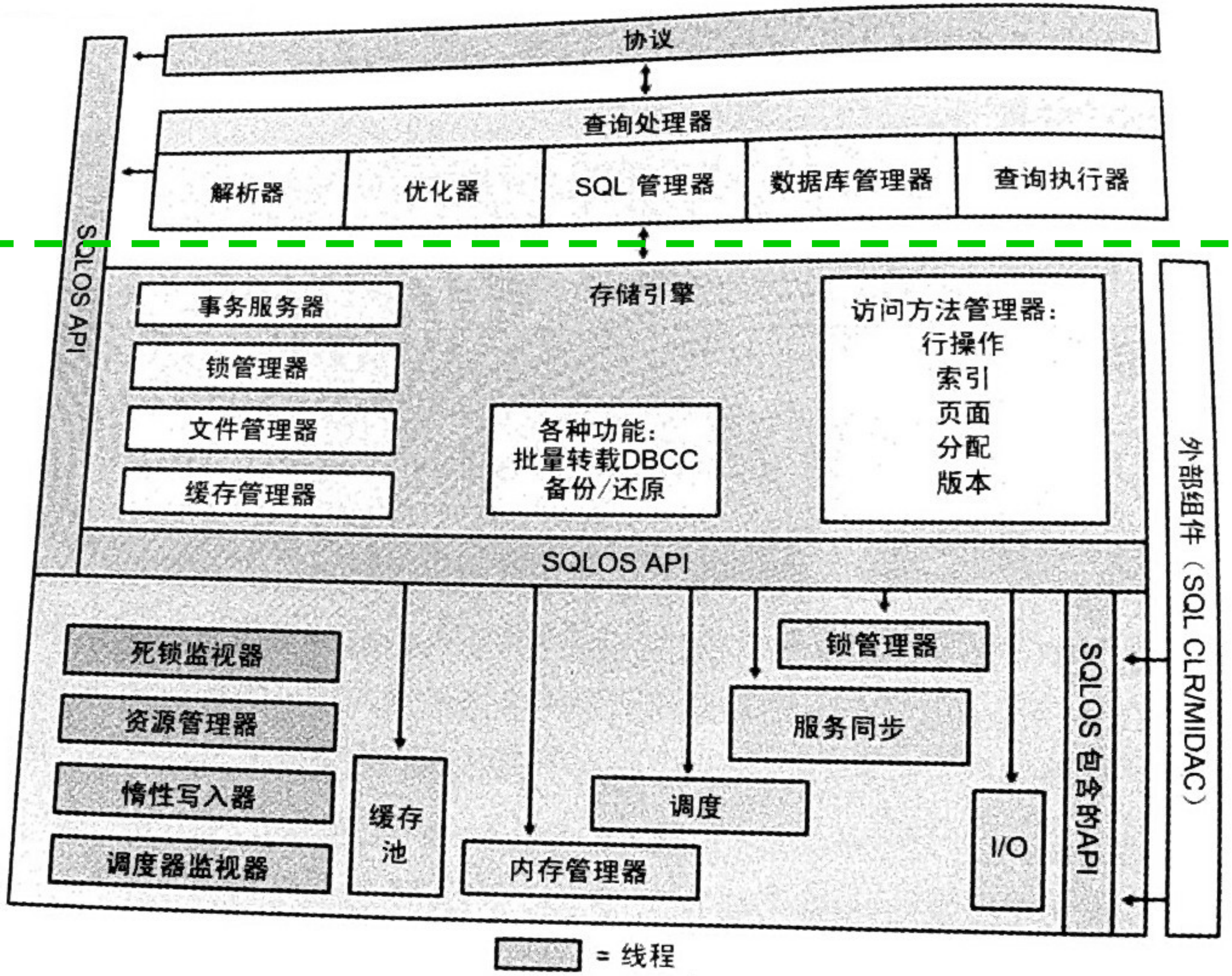
Overall System Structure



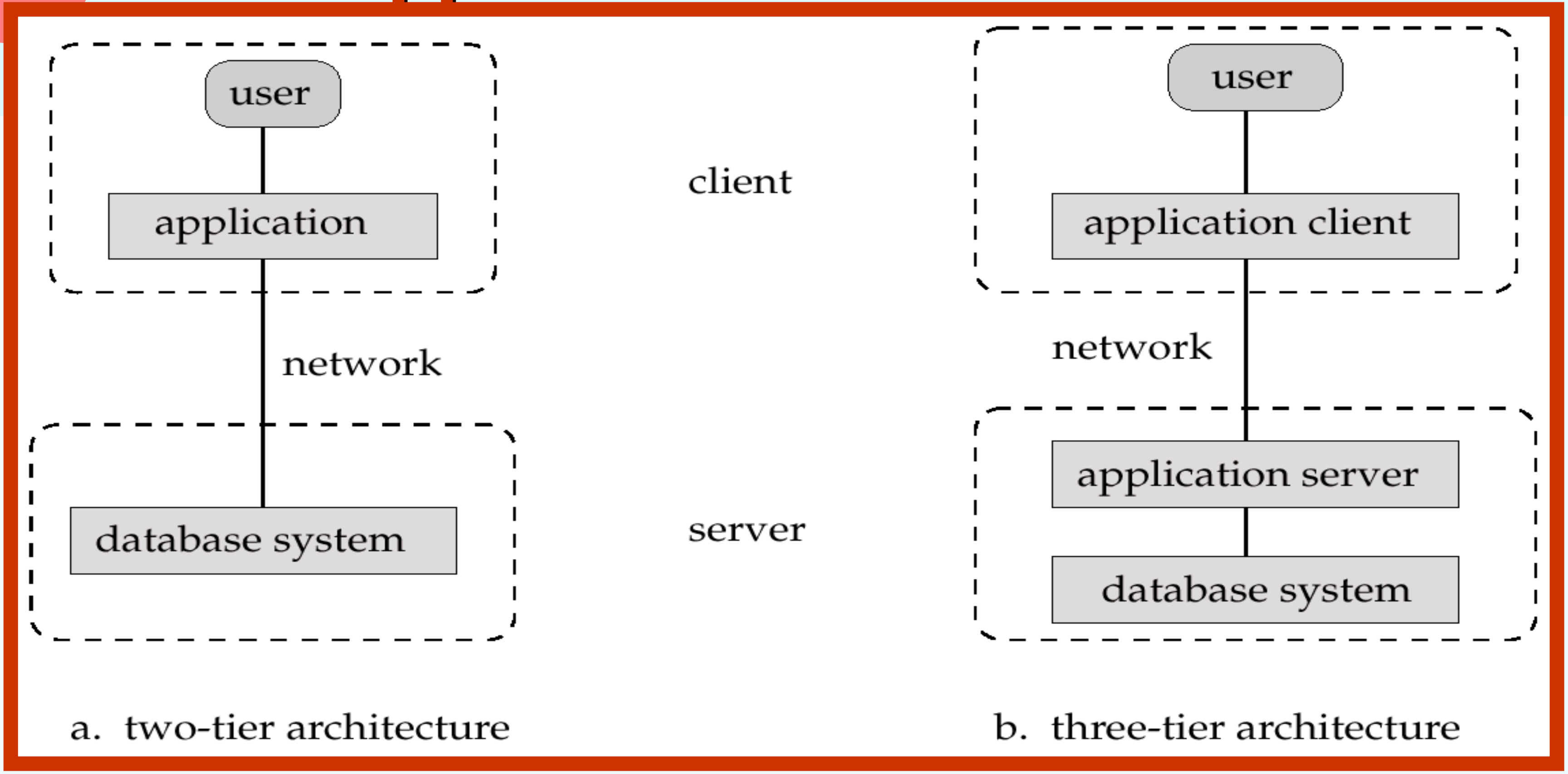
The Structure of SQL Sever

Query processor

Storage manager



Application Architectures



a. two-tier architecture

b. three-tier architecture

- Two-tier architecture: E.g., client programs using ODBC/JDBC to communicate with a database
- Three-tier architecture: E.g., web-based applications, and applications built using “**middleware**” (中间件)

Summary

- DBMS used to maintain, query large datasets
- Benefits include recovery from system crashes, concurrent access, quick application development, data integrity and security
- Levels of abstraction give data independence
- E-R model, Relational model
- DDL, DML, SQL
- DBAs hold responsible jobs and are **well-paid**!
- DBS typical architecture
- DBMS R&D is one of the broadest,

  most exciting areas in CS

The End

Exercise: 7th Edition 1.3, 1.7, 1.8, 1.12

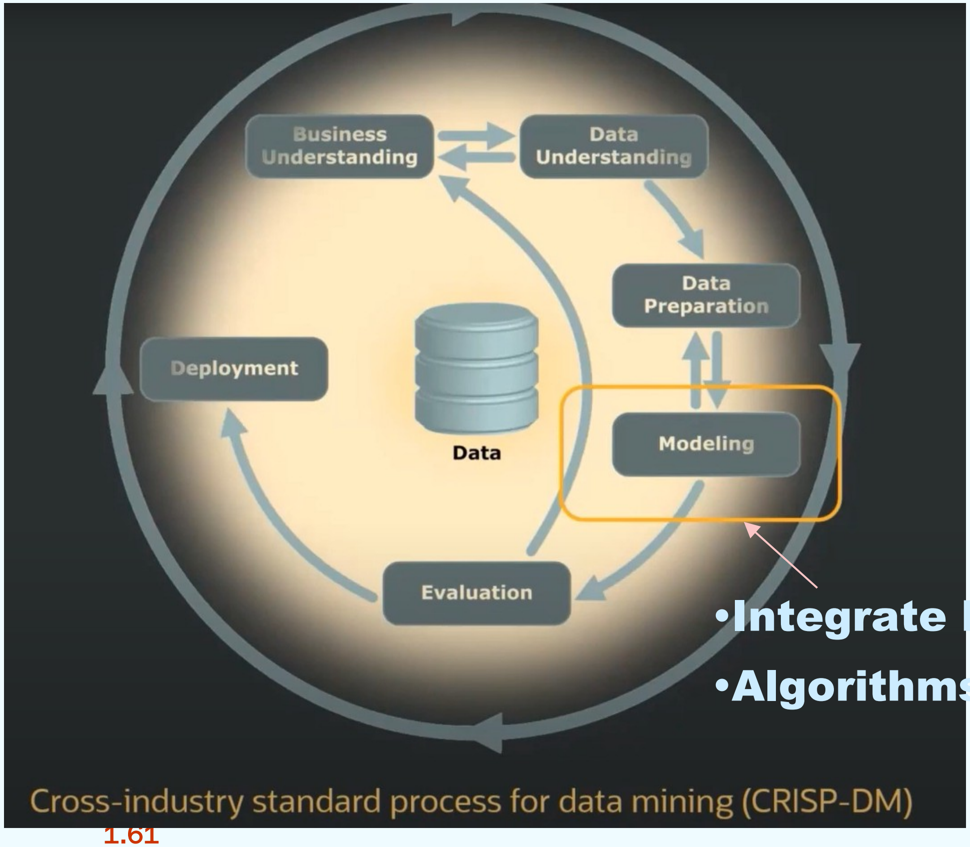
The Future

- 在 **2022** 云栖大会上阿里云智能数据库事业部负责人李飞飞表示，阿里云数据库将整体向“四化”方向发展
    - 云原生化 **(** 资源解耦、 **Serverless** 化 **)**
    - 平台化 **(** 基于云构建数据平台能力、 **OpenAPI** 标准化 **)**
    - 一体化 **(** 处理分析一体化、离在线一体化、集中分布一体化、多模处理一体化 **)**
    - 智能化 **(AI for DB** 简化运维、 **In-DB ML** 挖掘数据价值 **)**

In-DB ML

- 数据库内机器学习
  - 核心思想：将机器学习算法整合进现有的数据库管理系统
  - 优点：避免了将数据拷贝到外部机器学习工具的传输开销，直接使用机器学习算法在数据库内部分析数据

- 三个优势
  - 自动化处理：无需复杂且需多次迭代的传统机器学习建模方式，直接在数据库中自动化分析数据
  - 大规模数据训练：可以并行地，分布式地处理大规模数据，且无需数据移动
  - 可快速部署：将机器学习算法整合进数据建模过程，可快速将数据科学解决方案部署到产业上



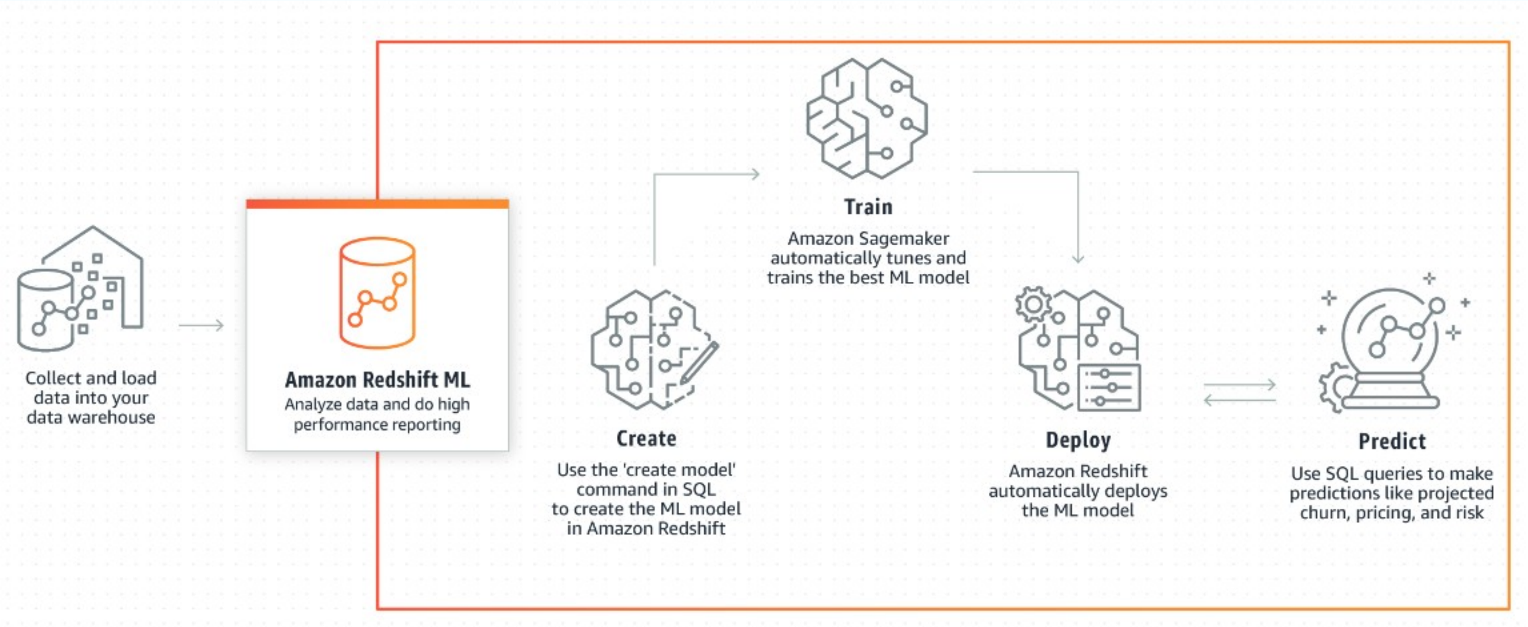
Cross-industry standard process for data mining (CRISP-DM)

•Integrate ML

•Algorithms

In-DB ML

- Amazon Redshift ML
  - 是一种基于云的稳健服务，可以在 Amazon Redshift 数据仓库中使用机器学习模型，并使用这些模型对数据进行预测
  - 易于上手： SQL 用户可以使用 SQL 命令即可创建、训练、部署机器学习模型
  - 支持多种机器学习算法： XGBoost 、多层感知、 K-Means 和线性学习器

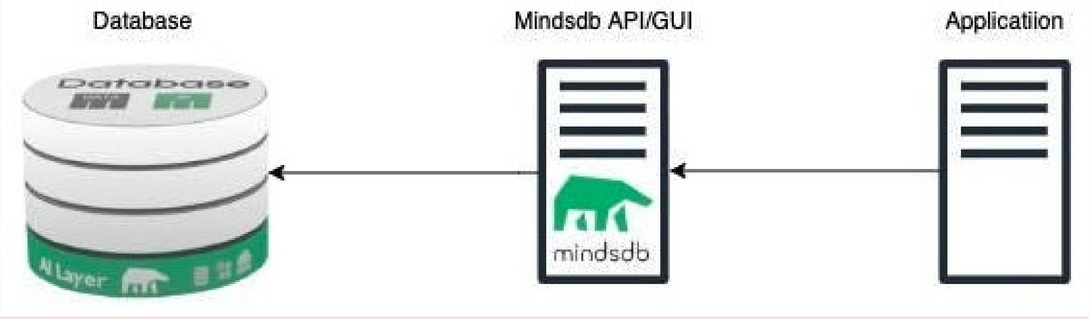

Collect and load data into your data warehouse
Amazon Redshift ML
Analyze data and do high performance reporting
Train
Amazon Sagemaker automatically tunes and trains the best ML model
Create
Use the 'create model' command in SQL to create the ML model in Amazon Redshift
Deploy
Amazon Redshift automatically deploys the ML model
Predict
Use SQL queries to make predictions like projected churn, pricing, and risk

- Amazon Redshift ML 的工作原理

In-DB ML

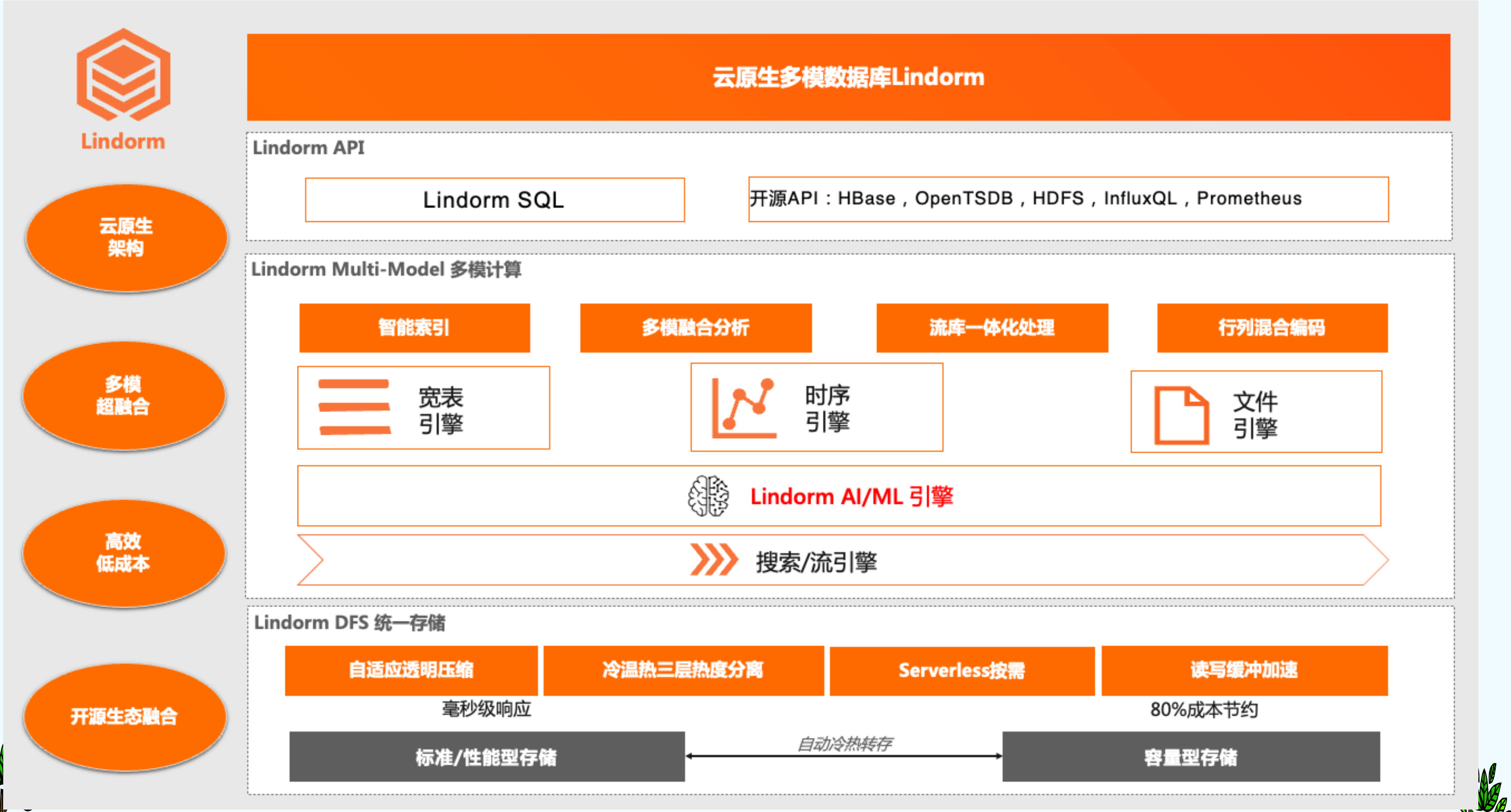
- MindsDB
  - 一个将机器学习技术整合进数据库的开源预测平台
  - 简洁易用：通过 SQL 命令可以轻松使用机器学习模型，而无需深入了解机器学习算法流程
  - 兼容多个数据库： MySQL 、 Postgres 、 Redit 、 Snowflakes 等
  - 可完成时间序列、回归和分类预测任务

- MindsDB 工作原理
  - AI Tables ：将机器学习模型作为虚拟表插入数据库中，创建预测，并可通过 SQL 语句查询 AI Tables 以获得输出



In-DB ML

- Lindorm ML 是阿里云首个云原生多模数据库内机器学习系统，实现了数据存储分析一体化
- 提供端到端的机器学习模型训练、管理和推理能力，降低了机器学习应用落地门槛



Lindorm

云原生架构

多模超融合

高效低成本

开源生态融合

云原生多模数据库Lindorm

Lindorm API

Lindorm SQL

开源API：HBase，OpenTSDB，HDFS，InfluxQL，Prometheus

Lindorm Multi-Model 多模计算

智能索引

多模融合分析

流库一体化处理

行列混合编码

宽表引擎

时序引擎

文件引擎

Lindorm AI/ML 引擎

搜索/流引擎

Lindorm DFS 统一存储

自适应透明压缩

冷温热三层热度分离

Serverless按需

读写缓冲加速

毫秒级响应

80%成本节约

标准/性能型存储

自动冷热转存

容量型存储