PRINCIPLES AND TECHNOLOGIES OF DATABASE SYSTEM – INTRODUCTION

李旭东 **Li-Xudong**
LEEXUDONG@NANKAI.EDU.CN
NANKAI UNIVERSITY

OBJECTIVES

- Database system
- Data Abstraction
- Database Model
- Database Languages
- Relational Database
- Database Design
- Database Architecture
- History of Database

DATABASE SYSTEM

- Database-management system
  - DBMS
  - A collection of interrelated data and a set of programs to access those data
- Goal of DBMS
  - Provide a way to store and retrieve database information that is both convenient and efficient
  - security

DATABASE-SYSTEM APPLICATIONS

- Enterprise Information
  - Sales, accounting, human resources, CRM, ERP, manufacturing, online retailers
- Banking and Finance
  - Banking, credit card transactions, finance stock
- Universities: Students, courses, grades
- Airlines : reservations, schedule information
- Telecommunication: records of calls made, bills
- …

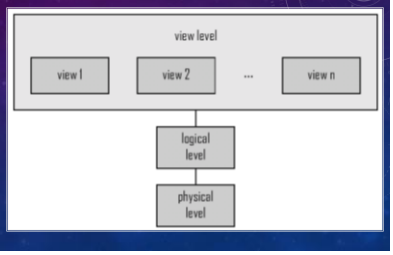
PURPOSE OF DATABASE SYSTEMS

- Enable durability (持久性) storage
- Large amounts of data
- Isolation (独立性) among users
- Atomicity(原子性)
- Reduce redundancy(冗余)
- Consistency(一致性)
- Integrity (完整性)
- Efficient store and access
- Concurrent(并发) access
- Security(安全)

DATA ABSTRACTION

ANSI/SPARC Archecture,1975

- View level
  - 视图层:外模式
  - External view: part of db
- Logical level
  - 逻辑层:模式
  - Conceptual view: what
- Physical level
  - 物理层:内模式
  - Internal view: how



EXAMPLE OF DATA ABSTRACTION: PHYSICAL VIEW, LOGICAL VIEW, EXTERNAL VIEW

```
type instructor = record
    ID : string;
    name : string;
    deptname : string;
    salary : integer;
    birth : date;
    gender : boolean;
end;

type department = record
    ...
end;

type student = record
    ...
end;
```

©LXD

DATABASE INSTANCES AND SCHEMAS

- DB Instance实例
  - The collection of information stored in the database at a particular moment
- DB Schema模式
  - The overall design of the database
  - Physical schema, logical schema
  - Subschema (view level)
- Logical data independence(逻辑数据独立性)
- Physical data independence(物理数据独立性)***

©LXD

DATA MODEL

- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints

©LXD

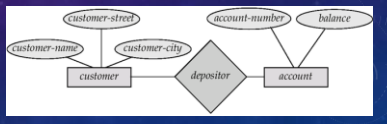
DATA (DATABASE) MODEL

- Entity-relationship model(实体关系E-R)
- Hierarchical data model(层次)
- Network data model(网络)
- Relational data model(关系)
- Object-based data model(对象)
- Semi-structured data model (半结构)
- Graph data model (图)

©LXD

ENTITY-RELATIONSHIP MODEL

- Entity-relationship model
  - 实体关系模型E-R
  - Entity
  - Relationship
  - Attribute属性



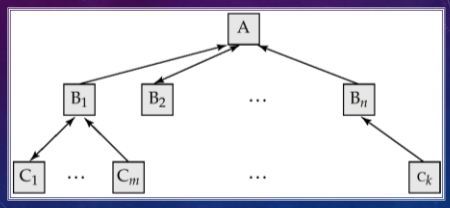
©LXD

HIERARCHICAL DATA MODEL

- A hierarchical database consists of a collection of *records* which are connected to one another through *links*.
- a record is a collection of fields, each of which contains only one data value.
- A link is an association between precisely two records.
- The hierarchical model differs from the network model in that the records are organized as collections of trees rather than as arbitrary graphs.

©LXD

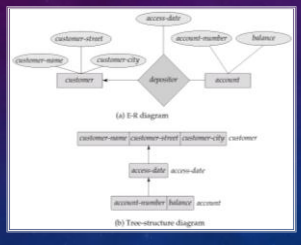
HIERARCHICAL DATA MODEL



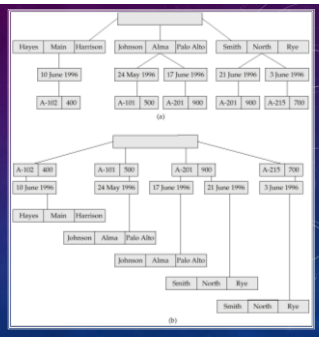
HIERARCHICAL DATA MODEL

- The schema for a hierarchical database consists of
  - *boxes*, which correspond to record types
  - *lines*, which correspond to links
- Record types are organized in the form of a *rooted tree.*
  - **No cycles** in the underlying graph.
  - Relationships formed in the graph must be such that only one-to-many or one-to-one relationships exist between a parent and a child.

HIERARCHICAL DATA MODEL: CASE



HIERARCHICAL DATA MODEL: CASE



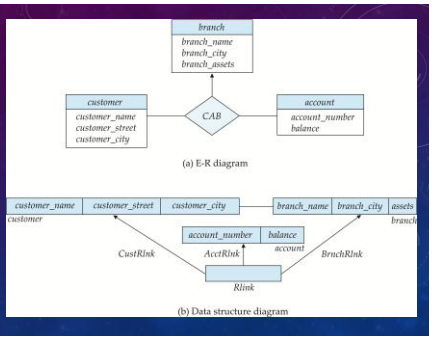
NETWORK DATA MODEL

- Data are represented by collections of *records*.
  - similar to an entity in the E-R model
  - Records and their fields are represented as *record type*
- Relationships among data are represented by *links*
  - similar to a restricted (binary) form of an E-R relationship
  - restrictions on links depend on whether the relationship is many-many, many-to-one, or one-to-one.

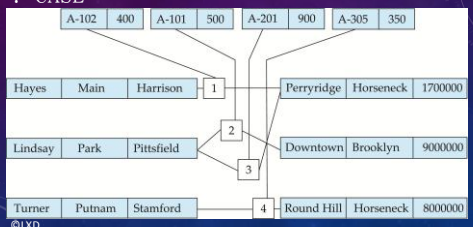
NETWORK DATA MODEL

- Schema representing the design of a network database.
- A data-structure diagram consists of two basic components:
  - **Boxes**, which correspond to record types.
  - **Lines**, which correspond to links.
- Specifies the overall logical structure of the database.

NETWORK DATA MODEL : CASE



NETWORK DATA MODEL : CASE



RELATIONAL DATA MODEL

- Relational data model
  - A collection of tables
  - Table
    - A collection of records
    - Multiple columns
  - Relation

RELATIONAL DATA MODEL: CASE

Columns

Rows

customer_id	customer_name	customer_street	customer_city	account_number
192-83-7465	Johnson	12 Alma St.	Palo Alto	A-101
192-83-7465	Johnson	12 Alma St.	Palo Alto	A-201
677-89-9011	Hayes	3 Main St.	Harrison	A-102
182-73-6091	Turner	123 Putnam St.	Stamford	A-305
321-12-3123	Jones	100 Main St.	Harrison	A-217
336-66-9999	Lindsay	175 Park Ave.	Pittsfield	A-222
019-28-3746	Smith	72 North St.	Rye	A-201

RELATIONAL DATA MODEL: CASE

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

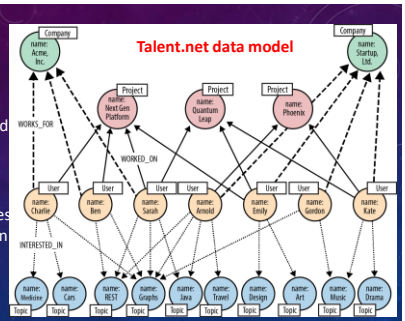
(b) The *account* table

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

GRAPH DATA MODEL

- A graph is just a collection of vertices and edges or, a set of nodes and the relationships that connect them.
- Graphs represent entities as nodes and the ways in which those entities relate to the world as relationships.



GRAPH DATA MODEL

- A labeled property graph has the following characteristics:
  - It contains nodes and relationships.
  - Nodes contain properties (key-value pairs).
  - Nodes can be labeled with one or more labels.
  - Relationships are named and directed, and always have a start and end node.
  - Relationships can also contain properties.

DATABASE LANGUAGES

- Data-Definition Language (DDL)
- Data-Manipulation Language (DML)

DATA-DEFINITION LANGUAGE (DDL)

- Specification notation for defining the database schema

  Example: **create table** *instructor* (
  ID **char**(5),
  name **varchar**(20),
  dept_name **varchar**(20),
  salary **numeric**(8,2))

- DDL compiler generates a set of table templates stored in a ***data dictionary数据字典***
- Data dictionary contains metadata (i.e., data about data, 元数据)

DATA-DEFINITION LANGUAGE (DDL)

- Data dictionary
  - Database schema
  - Consistency constraint
- Consistency constraint一致性约束
  - Domain constraint域约束
  - Referential Integrity参照完整性
    - Primary key, foreign key
  - Assertion断言
  - Authorization授权: read, insert, update, delete, …

DATA-MANIPULATION LANGUAGE (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  - DML also known as query language
- Two classes of languages
  - Pure – used for proving properties about computational power and for optimization
    - Relational Algebra
    - Tuple元组 relational calculus演算
    - Domain域 relational calculus
  - Commercial – used in commercial systems
    - SQL is the most widely used commercial language

DATA-MANIPULATION LANGUAGE (DML)

- Types of DB Access
  - Retrieval of information stored in the database
  - Insertion of new information into the database
  - Deletion of information from the database
  - Modification of information stored in the database

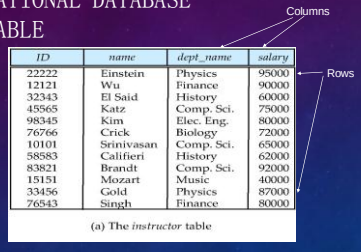
DATA-MANIPULATION LANGUAGE (DML)

- Procedural DMLs(过程式)
  - Require a user to specify what data are needed and how to get those data
- Declarative DMLs(声明式)
  - Nonprocedural DMLs
  - Require a user to specify what data are needed without specifying how to get those data

RELATIONAL DATABASE

- Table
- Data-Manipulation Language
- Data-Definition Language
- Database Access from Application Programs

RELATIONAL DATABASE – TABLE



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

RELATIONAL DATABASE – TABLE

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

RELATIONAL DATABASE – DML

- SQL query language
  - Structured Query Language
  - Nonprocedural
- Example

```
Select  instructor.name
From instructor
Where
   instructor.dept_name = 'History';
```


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

RELATIONAL DATABASE – DML

```
Select instructor.ID,
department.dept_name
From instructor, department
Where
Instructor.dept_name =
    department.dept_name
And
department.budget > 95000;
```


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

RELATIONAL DATABASE – DDL

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

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

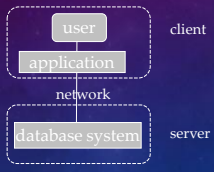
©LXD

SQL STANDARD: ISO

https://en.wikipedia.org/wiki/SQL

Year	Name	Alias	Comments
1986	SQL-86	SQL-87	First formalized by ANSI.
1989	SQL-89	FIPS 127-1	Minor revision that added integrity constraints, adopted as FIPS 127-1.
1992	SQL-92	SQL2, FIPS 127-2	Major revision (ISO 9075), *Entry Level* SQL-92 adopted as FIPS 127-2.
1999	SQL:1999	SQL3	Added regular expression matching, recursive queries (e.g. transitive closure), triggers, support for procedural and control-of-flow statements, non-scalar types (arrays), and some object-oriented features (e.g. structured types). Support for embedding SQL in Java (SQL/OLB) and vice versa (SQL/JRT).
2003	SQL:2003		Introduced XML-related features (SQL/XML), window functions, standardized sequences, and columns with auto-generated values (including identity-columns).
2006	SQL:2006		ISO/IEC 9075-14:2006 defines ways that SQL can be used with XML. It defines ways of importing and storing XML data in an SQL database, manipulating it within the database, and publishing both XML and conventional SQL-data in XML form. In addition, it lets applications integrate queries into their SQL code with XQuery, the XML Query Language published by the World Wide Web Consortium (W3C), to concurrently access ordinary SQL-data and XML documents.[21]
2008	SQL:2008		Legalizes ORDER BY outside cursor definitions. Adds INSTEAD OF triggers, TRUNCATE statement,[22] FETCH clause.
2011	SQL:2011		Adds temporal data (PERIOD FOR)[23] (more information at: Temporal database#History). Enhancements for window functions and FETCH clause.[24]
2016	SQL:2016		Adds row pattern matching, polymorphic table functions, JSON.

RELATIONAL DATABASE – DATABASE ACCESS FROM APPLICATION PROGRAMS



(a) Two-tier architecture

©LXD

RELATIONAL DATABASE – DATABASE ACCESS FROM APPLICATION PROGRAMS

- Application Programs: C, C++, Java, Php, …
- Ways to access db
  - By providing an application program interface (set of procedures) that can be used to send DML and DDL statements to the db and retrieve the results
    - ODBC (Open Database Connectivity), JDBC
  - By extending the host(宿主) language syntax to embed DML calls within the host language program
    - DML precompiler 预编译器

©LXD

DATABASE DESIGN

- Database design
  - Involves the design of the database schema
- Database design process
  - A high-level data model
    - conceptual framework: requirement
  - Choose a data model
    - logical-design, physical-design

©LXD

DATABASE DESIGN – A HIGH-LEVEL DATA MODEL

盲人摸象
The Blind Men and the Elephant

©LXD

DATABASE DESIGN: CASE

- A university organization
  - The university is organized into departments. Each department is identified by a unique name (dept name), is located in a particular building, and has a budget.
  - Each department has a list of courses it offers. Each course has associated with it a course id, title, dept name, and credits, and may also have associated prerequisites.
  - Instructors are identified by their unique ID. Each instructor has name, associated department (dept name), and salary.
  - Students are identified by their unique ID. Each student has a name, an associated major department (dept name), and tot cred (total credit hours the student earned thus far).
  - The university maintains a list of classrooms, specifying the name of the building, room number, and room capacity.

DATABASE DESIGN: CASE

- A university organization (cont.,)
  - The university maintains a list of all classes (sections) taught. Each section is identified by a course id, sec id, year, and semester, and has associated with it a semester, year, building, room number, and time slot id (the time slot when the class meets).
  - The department has a list of teaching assignments specifying, for each instructor, the sections the instructor is teaching.
  - The university has a list of all student course registrations, specifying, for each student, the courses and the associated sections that the student has taken (registered for).

DATABASE DESIGN: CASE

- E-R Model
  - Entity
    - attribute
  - Entity set
  - Relationship
  - Constraint
    - Mapping cardinalities映射基数

DATABASE DESIGN: CASE

- Normalization规范化

DATABASE DESIGN: CASE

- Normalization
- Is there any problem with this relation?

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

DATABASE DESIGN: CASE

- Normalization theory
  - Functional dependency函数依赖
    - *e.g. dept_name → building, budget*

DATABASE DESIGN

- Data Storage and Querying
  - Storage manager
    - Storage space
  - Query processor
    - Performance

DATABASE DESIGN – STORAGE MANAGER

- Storage Manager
  - Authorization and integrity manager
  - Transaction manager
  - File manager
  - Buffer manager

DATABASE DESIGN – STORAGE MANAGER

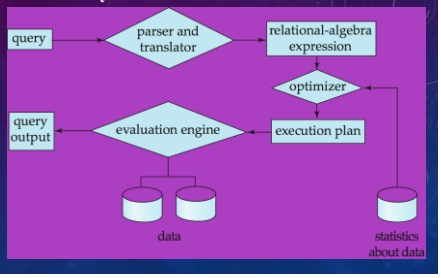
- Storage manager
  - a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
  - The storage manager is responsible to the following tasks:
    - Interaction with the OS file manager
    - Efficient storing, retrieving and updating of data

DATABASE DESIGN – STORAGE MANAGER

- Physical implementation of DB Storage
  - Data files, which store the database itself
  - Data dictionary
  - Indices(索引)

DATABASE DESIGN– QUERY PROCESSOR

- Parsing and translation
- Optimization
- Evaluation



DATABASE DESIGN – QUERY PROCESSOR

- Query Processor
  - DDL interpreter
  - DML interpreter
    - A query can be translated into any of a number of alternative evaluation plans
    - Query optimization
- Query evaluation engine查询执行引擎

DATABASE DESIGN – TRANSACTION MANAGEMENT

- Transaction事务
  - A collection of operations that performs a single logical function in a database application
  - Example: transfer funds from A to B

DATABASE DESIGN – TRANSACTION MANAGEMENT

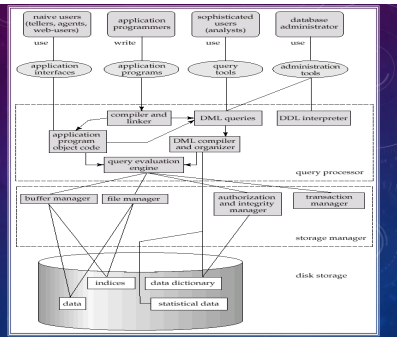
- The ACID Properties of Transactions
  - Atomicity原子性
  - Consistency一致性
  - Isolation独立性
    - Each transaction must appear to be executed as if no other transaction is executing at the same time
  - Durability持久性

DATABASE DESIGN – TRANSACTION MANAGEMENT

- Transaction Management
  - Logging记日志
  - Concurrency control
  - Deadlock resolution
  - Recovery manager
    - Failure recovery

DATABASE ARCHITECTURE

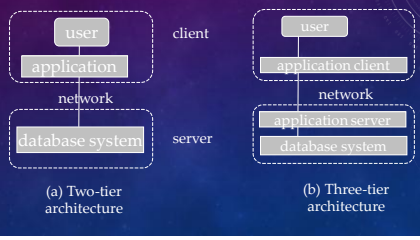
- Components of a DB



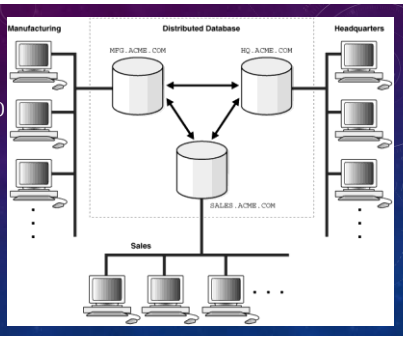
DATABASE APPLICATION ARCHITECTURE

- The architecture of a database systems is greatly influenced by the underlying computer system on which the database is running:
  - Centralized集中式
  - Client-server客户服务器式
  - Parallel (multi-processor)并行: multi-processors
  - Distributed 分布式

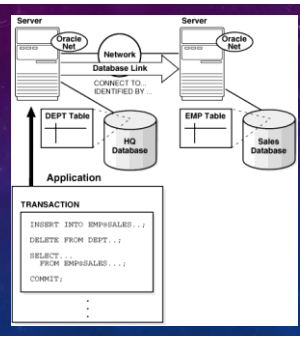
DATABASE APPLICATION ARCHITECTURE – CLIENT-SERVER



DATABASE APP ARCHITECTURE – DISTRIBUTED DATABASE



DATABASE APP ARCHITECTURE – DISTRIBUTED DATABASE (ORACLE)



DATA MINING

- Data Mining
  - Refers loosely to the process of semi-automatically analyzing large DB to find useful patterns
    - Case: beer and nappy
  - Knowledge discovery in artificial intelligence: machine learning

DATA WAREHOUSE

- Data warehouse
  - 数据仓库
  - Gather data from multiple sources under a unified schema, at a single site
  - Then, they provide the user a single uniform interface to data

INFORMATION RETRIEVAL

- Information Retrieval信息检索
  - Querying of unstructured textual data
  - Textual data is unstructured, unlike the rigidly structured data in relational databases
  - Textual data has grown explosively

INFORMATION RETRIEVAL: TODAY

MORE DATABASE MODELS …

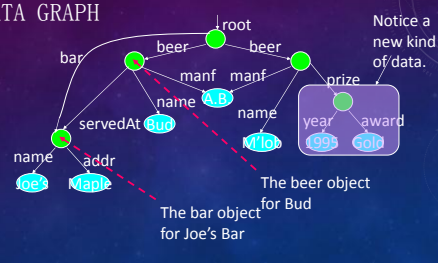
SPECIALTY DATABASES

- Object-based data models
  - Object
  - Object-relational data model
  - As extending the E-R model with notions of encapsulation, methods, and object identity.

SPECIALTY DATABASES

- Semistructured data models
  - It permit the specification of data where individual data items of the same type may have different sets of attributes.
  - Data Graph
  - XML

SPECIALTY DATABASES – DATA GRAPH



SPECIALTY DATABASES – XML : EXTENSIBLE MARKUP LANGUAGE

- Defined by the WWW Consortium (W3C)
- Originally intended as a document markup language not a database language
- The ability to specify new tags, and to create nested tag structures made XML a great way to exchange data, not just documents
- XML has become the basis for all new generation data interchange formats.
- A wide variety of tools is available for parsing, browsing and querying XML documents/data

SPECIALTY DATABASES – XML

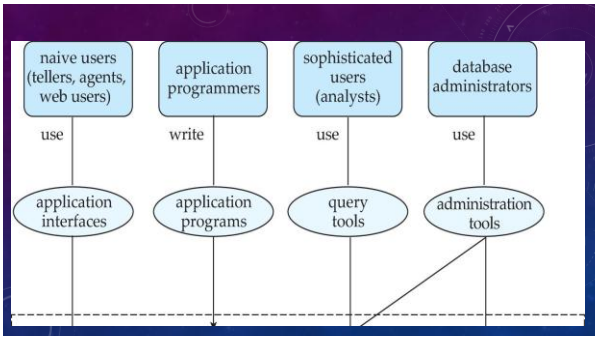
```
<?xml version = "1.0" encoding = "utf-8" ?>
<BARS>
  <BAR name = "Joe's Bar">
    <BEER name = "Bud" price = 2.50 />
    <BEER name = "Miller" price = 3.00 />
  </BAR>
  <BAR> …
</BARS>
```

name and price are attributes

DATABASE USERS AND ADMINISTRATORS

- Database Users and User Interfaces
  - Naïve users
  - Application programmers
  - Sophisticated users
  - Specialized users

©LXD



DATABASE USERS AND ADMINISTRATORS

- Database Administrator数据库管理员
  - Schema definition
  - Storage structure and access-method definition
  - Schema and physical-organization modification
  - Granting of authorization for data access
  - Routine maintenance
    - Backup, Enough free disk space, Monitoring jobs

©LXD

HISTORY OF DATABASE SYSTEMS

- 1950s and early 1960s: tape
- Late 1960s and 1970s
  - Hard disk, file, DB(hierarchical, network)
  - network db
    - CODASYL ,Integrated Data Store (IDS)
- **Codd**, E. F.. "A relational model of data for large shared data banks." Communications of The ACM 13.6 (1970): 377-387.

©LXD

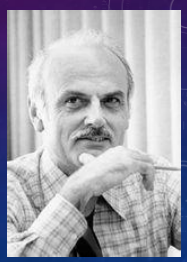
HISTORY OF DATABASE SYSTEMS

- 1980s
  - System R: IBM
    - Astrahan, Morton M., et al. "System R: relational approach to database management." ACM Transactions on Database Systems 1.2 (1976): 97-137.
  - Ingres: BSD
  - IBM DB2, Oracle, DEC Rdb
- Early 1990s
  - Object-relational DB

©LXD

EDGAR F. CODD

- Edgar Frank Codd
  - Edgar Frank "Ted" Codd was an English computer scientist who, while working for IBM, invented the relational model for database management, the theoretical basis for relational databases
  - ACM Turing Award, 1981
  - Online analytical processing (OLAP)
  - 19 August 1923 – 18 April 2003



©LXD

HISTORY OF DATABASE SYSTEMS

- 1990s
  - WWW: DB had to support Web interface to data
- 2000s
  - XML, Xquery
  - PostgreSQL, MySQL
  - Giant data storage systems
    - Google BigTable, Yahoo PNuts, Amazon, …

RANK OF DATABASES 2019.03

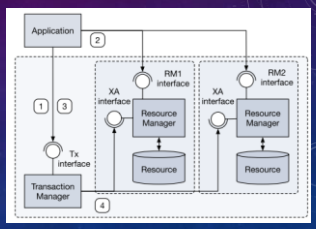
345 systems in ranking, March 2019

Rank Mar 2019	Rank Feb 2019	Rank Mar 2018	DBMS	Database Model	Score Mar 2019	Score Feb 2019	Score Mar 2018
1.	1.	1.	Oracle	Relational, Multi-model	1279.14	+15.12	-10.47
2.	2.	2.	MySQL	Relational, Multi-model	1198.25	+30.96	-30.62
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1047.85	+7.79	-56.94
4.	4.	4.	PostgreSQL	Relational, Multi-model	469.81	-3.75	+70.46
5.	5.	5.	MongoDB	Document	401.34	+6.24	+60.82
6.	6.	6.	IBM Db2	Relational, Multi-model	177.20	-2.23	-9.47
7.	↑ 9.	7.	Microsoft Access	Relational	146.20	+2.18	+14.26
8.	↓ 7.	8.	Redis	Key-value, Multi-model	146.12	-3.32	+14.90
9.	↓ 8.	9.	Elasticsearch	Search engine, Multi-model	142.79	-2.46	+14.25
10.	10.	↑ 11.	SQLite	Relational	124.87	-1.29	+10.06

ACID THEOREM

- Atomicity
- Consistency
- Isolation
- Durability

Transaction Processing System



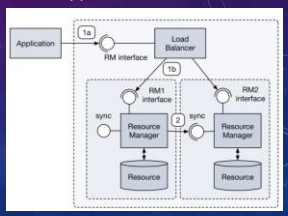
CAP THEOREM

it is **impossible** for a distributed data store to simultaneously provide more than two out of the following three guarantees

BASE THEOREM

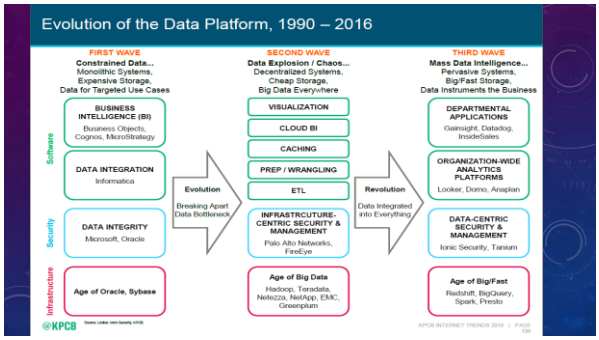
- Basically Available
- Soft state
- Eventually consistent

App Based BASE MODEL



TODAY

Data as a Platform



SUMMARY

- Database system
- Data Abstraction
- Database Model
- Database Languages
- Relational Database
- Database Design
- Database Architecture
- History of Database

©LXD

Q&A?

THANKS!

leexudong@nankai.edu.cn

赞数字文明时代之开启

李旭东 2017

文明初叶几时真，造化阴阳始幻尘。
书简成山薪火旺，零壹遁迹智能春。
有形百载多将朽，数字千年总是新。
懵懂蹒跚别旧日，一朝奋起笑前津。

©LXD