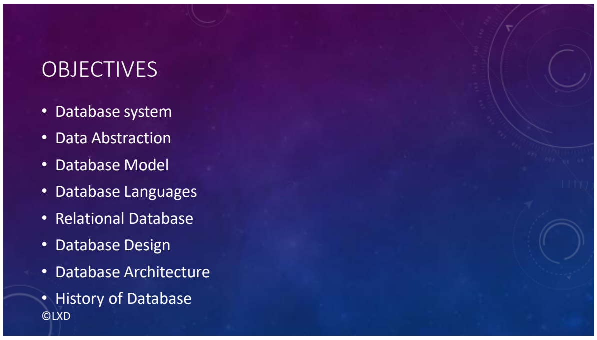
DATABASE SYSTEM PRINCIPLE - INTRODUCTION

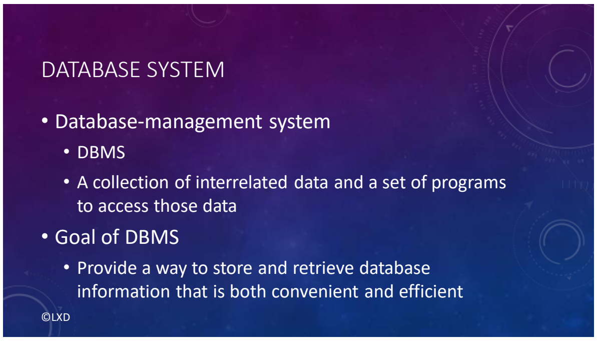
李旭东
LEEXUDONG@NANKAI.EDU.CN
NANKAI UNIVERSITY



OBJECTIVES

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

©LXD



DATABASE SYSTEM

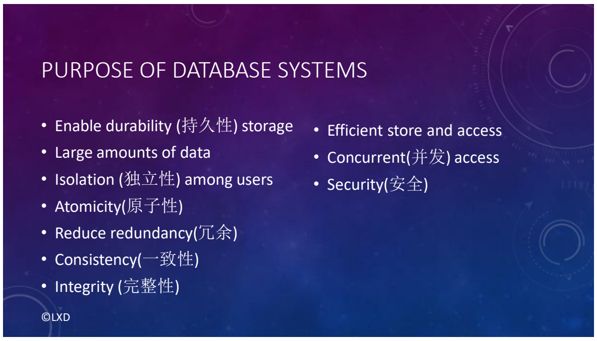
- Database-management system
  - DBMS
  - A collection of interrelated data and a set of programs to access those data
- Goal of DBMS
  - Provide a way to store and retrieve database information that is both convenient and efficient

©LXD

DATABASE-SYSTEM APPLICATIONS

- Enterprise Information
  - Sales, accounting, human resources, manufacturing, online retailers
- Banking and Finance
  - Banking, credit card transactions, finance stock
- Universities
- Airlines
- Telecommunication
- …

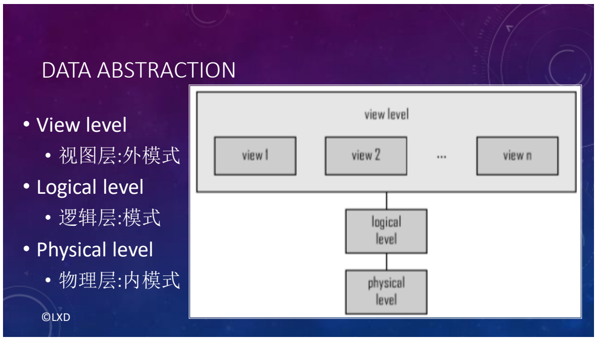
©LXD



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)

©LXD



DATA ABSTRACTION

- View level
  - 视图层:外模式
- Logical level
  - 逻辑层:模式
- Physical level
  - 物理层:内模式

©LXD

DATA ABSTRACTION: CASE

```
type instructor = record
    ID : string;
    name : string;
    dept_name : 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)
- 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

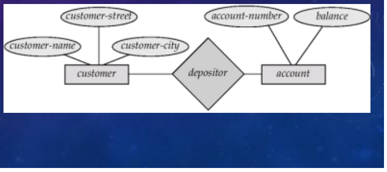
DATA (DATABASE) MODEL

- 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)
- Semistructured data model (半结构)

©LXD

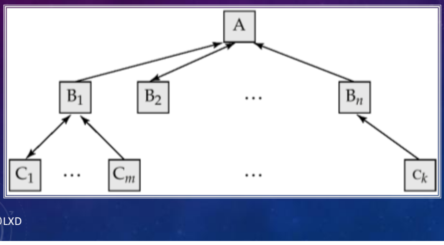
ENTITY-RELATIONSHIP MODEL

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

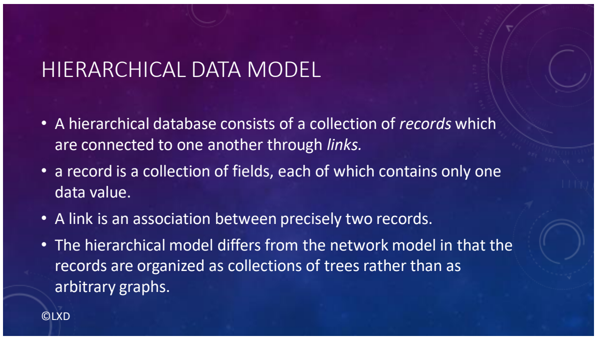


©LXD

HIERARCHICAL DATA MODEL



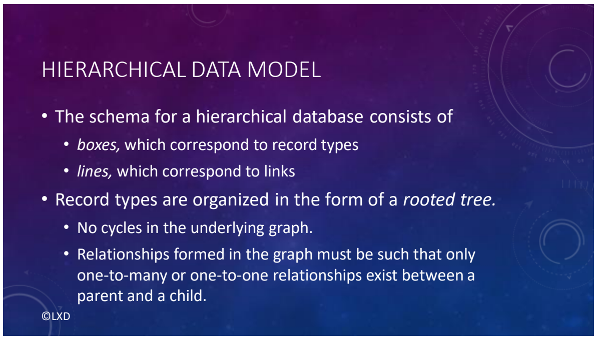
©LXD



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.

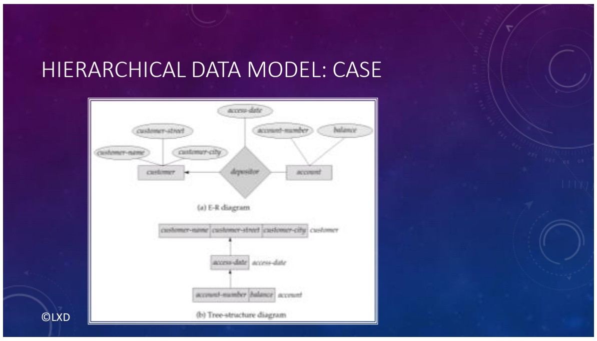
©LXD



HIERARCHICAL DATA MODEL

- 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.

©LXD

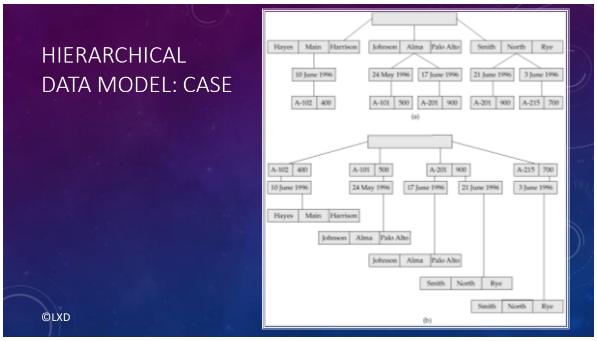


HIERARCHICAL DATA MODEL: CASE

(a) E-R diagram

(b) Tree-structure diagram

©LXD

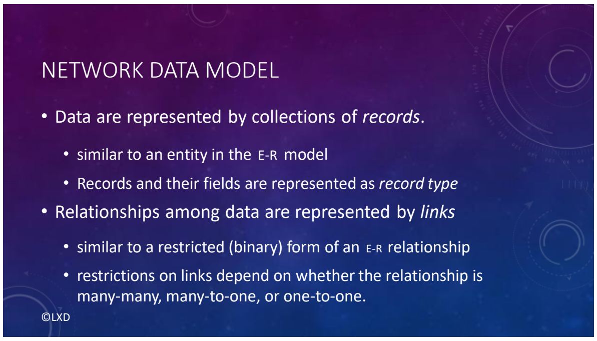


HIERARCHICAL DATA MODEL: CASE

(a)

(b)

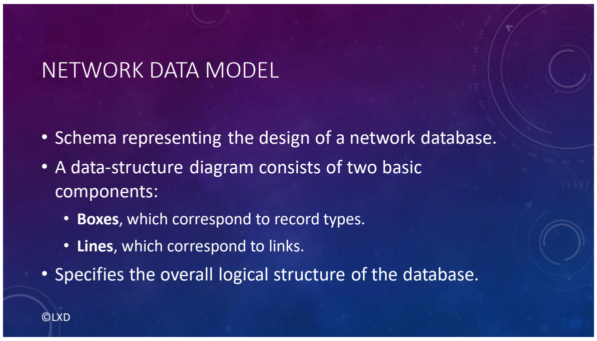
©LXD



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.

©LXD

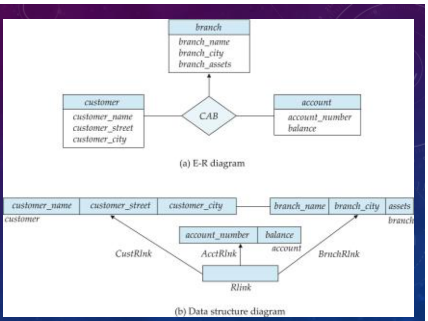


NETWORK DATA MODEL

- 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.

©LXD

NETWORK DATA MODEL : CASE

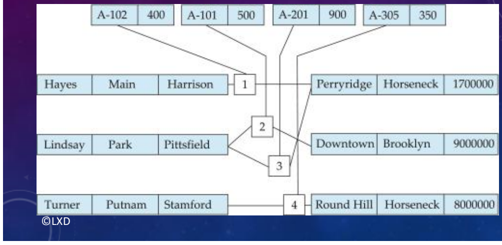


(a) E-R diagram

(b) Data structure diagram

©LXD

NETWORK DATA MODEL : CASE



©LXD

RELATIONAL DATA MODEL

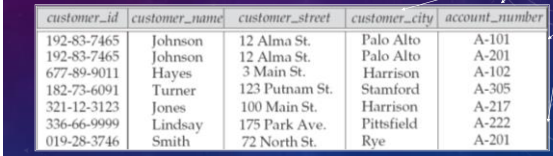
- Relational data model
  - A collection of tables
  - Table
    - A collection of records
    - Multiple columns
  - Relation

©LXD

RELATIONAL DATA MODEL: CASE

Columns

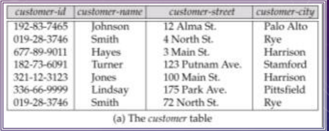
Rows



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

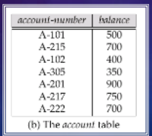
©LXD

RELATIONAL DATA MODEL: CASE



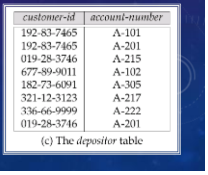
customer-id	customer-name	customer-street	customer-city
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746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

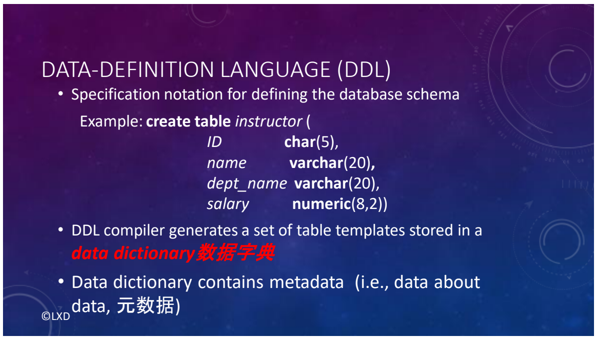
(c) The *depositor* table

©LXD

DATABASE LANGUAGES

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

©LXD



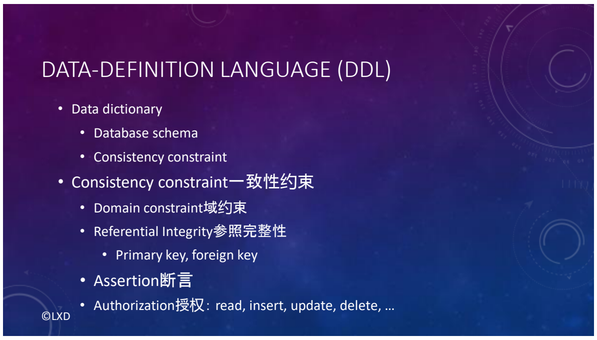
DATA-DEFINITION LANGUAGE (DDL)

- 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)

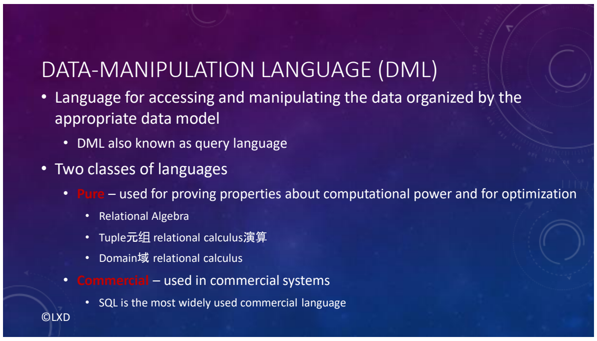
©LXD



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, …

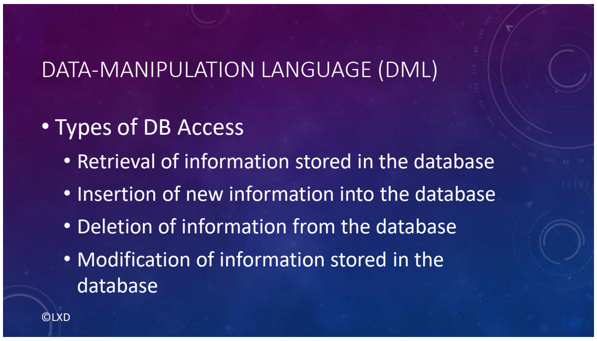
©LXD



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

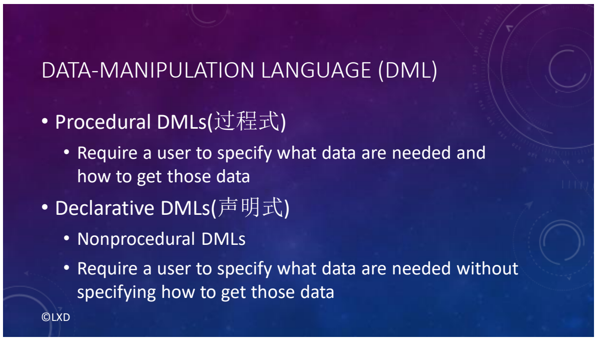
©LXD



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

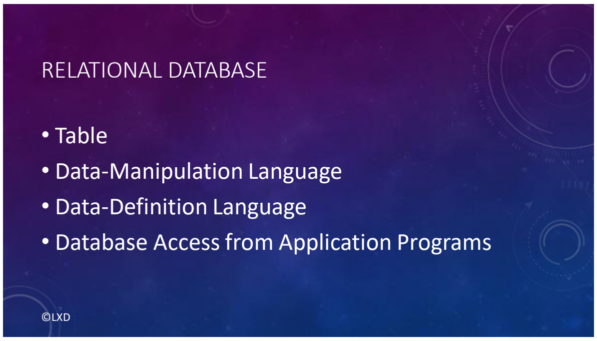
©LXD



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

©LXD

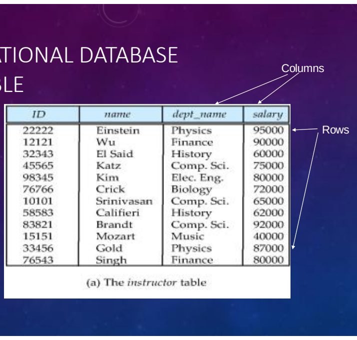


RELATIONAL DATABASE

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

©LXD

RELATIONAL DATABASE - TABLE



Columns

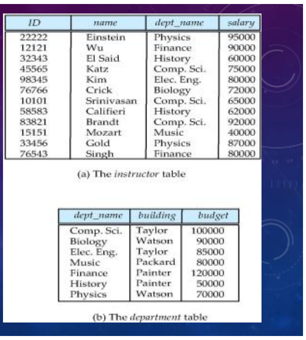
Rows

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

©LXD

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

©LXD

RELATIONAL DATABASE - DML

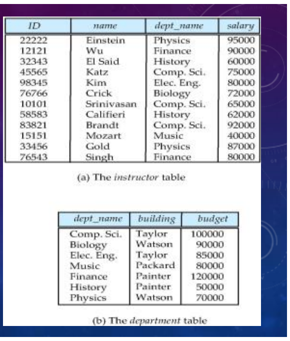
- SQL query language
  - Structured Query Language
  - Nonprocedural
- Example

  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

©LXD

RELATIONAL DATABASE - DML

Select instructor.ID,

department.dept_name

From instructor, department

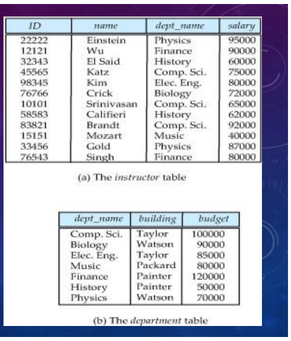
Where

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

©LXD

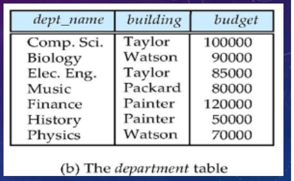
RELATIONAL DATABASE - DDL

create table department(

dept_name char(20),

building char(15),

budget numeric(12,2));



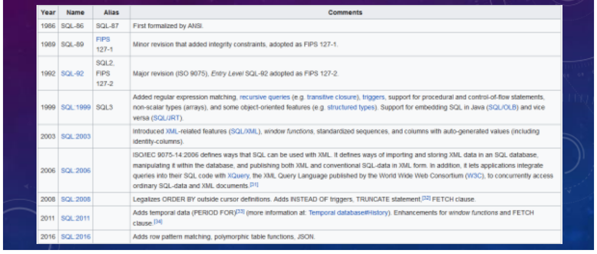
dept_name	building	budget
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

©LXD

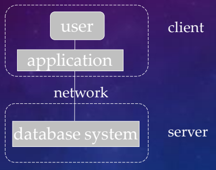
SQL STANDARD: ISO

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.[31]
2008	SQL:2008		Legalizes ORDER BY outside cursor definitions. Adds INSTEAD OF triggers, TRUNCATE statement,[32] FETCH clause.
2011	SQL:2011		Adds temporal data (PERIOD FOR)[33] (more information at: Temporal database#History). Enhancements for *window functions* and FETCH clause.[34]
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

©LXD

DATABASE DESIGN: CASE

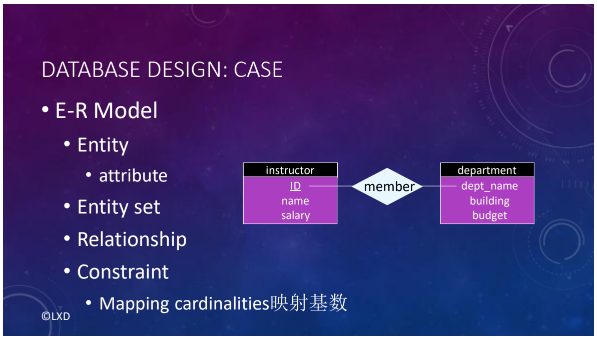
- A university organization
  - The university is organized into departments. Each department is identified by a unique name (dept name), is located in a particular building, and has a budget.
  - Each department has a list of courses it offers. Each course has associated with it a course id, title, dept name, and credits, and may also have 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.

©LXD

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).

©LXD

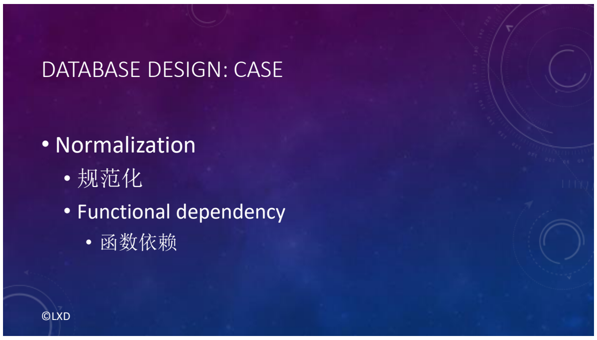


DATABASE DESIGN: CASE

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

instructor
ID
name
salary
member
department
dept_name
building
budget

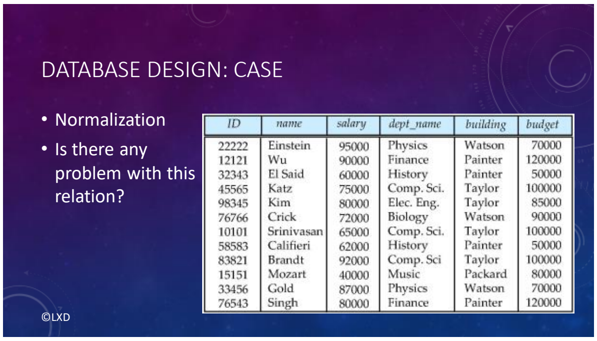
©LXD



DATABASE DESIGN: CASE

- Normalization
  - 规范化
  - Functional dependency
    - 函数依赖

©LXD

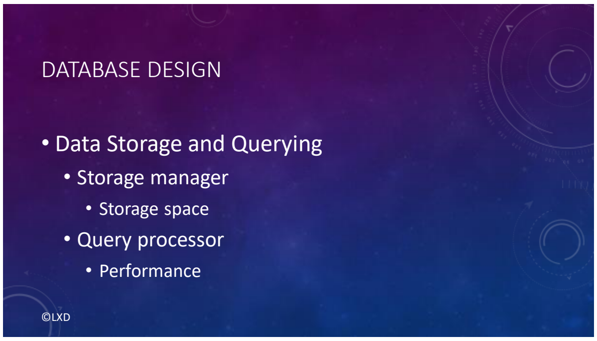


DATABASE DESIGN: CASE

- 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

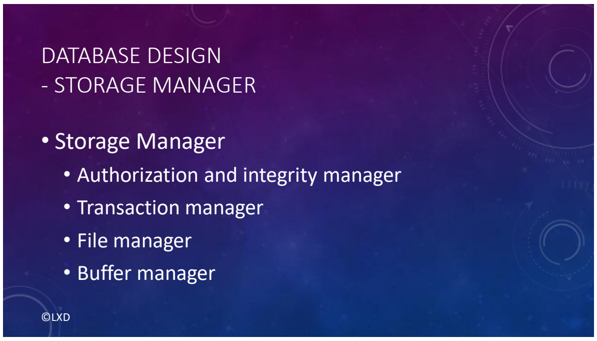
©LXD



DATABASE DESIGN

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

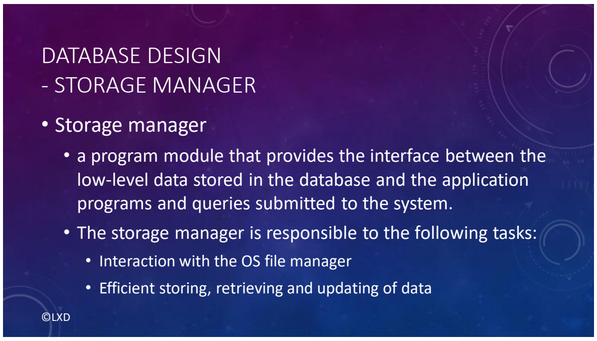
©LXD



DATABASE DESIGN - STORAGE MANAGER

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

©LXD



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

©LXD

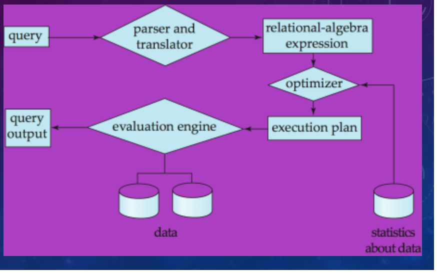
DATABASE DESIGN
- STORAGE MANAGER

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

©LXD

DATABASE DESIGN- QUERY PROCESSOR

- Parsing and translation
- Optimization
- Evaluation



©LXD

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

©LXD

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

©LXD

DATABASE DESIGN
- TRANSACTION MANAGEMENT

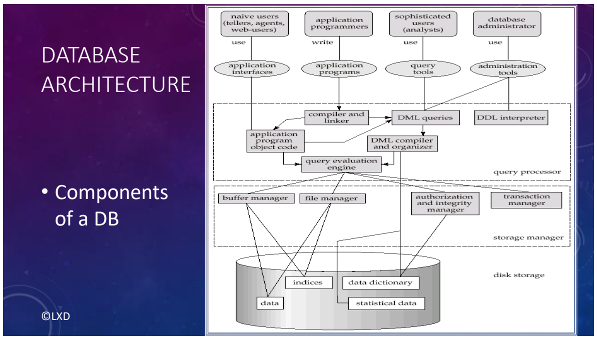
- The ACID Properties of Transactions
  - Atomicity原子性
  - Consistency一致性
  - Isolation独立性
    - Each transaction must appear to be executed as if no other transaction is executing at the same time
  - Durability持久性

©LXD

DATABASE DESIGN
- TRANSACTION MANAGEMENT

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

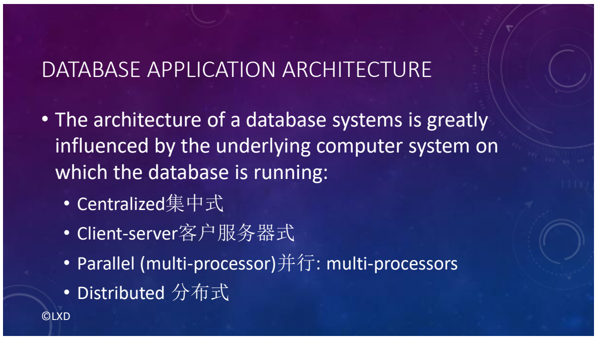
©LXD



DATABASE ARCHITECTURE

- Components of a DB

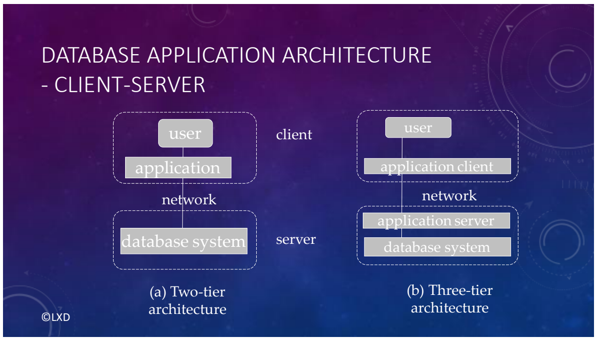
©LXD



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

©LXD

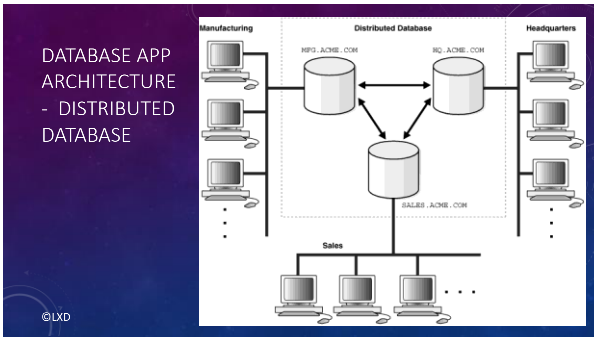


DATABASE APPLICATION ARCHITECTURE - CLIENT-SERVER

(a) Two-tier architecture

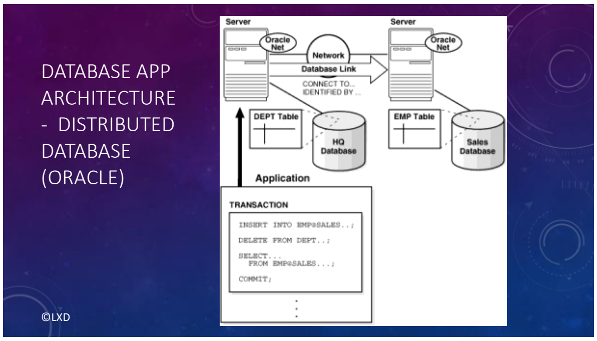
(b) Three-tier architecture

©LXD



DATABASE APP ARCHITECTURE - DISTRIBUTED DATABASE

©LXD



DATABASE APP ARCHITECTURE - DISTRIBUTED DATABASE (ORACLE)

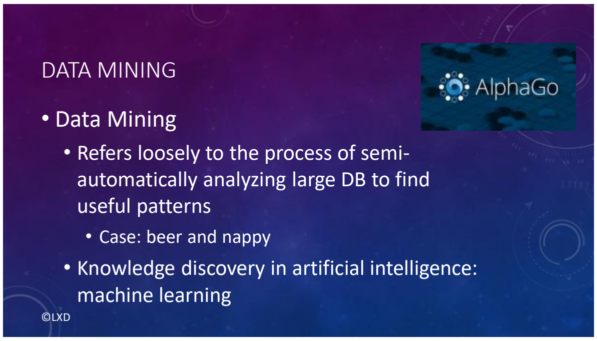
```
INSERT INTO EMP@SALES..;

DELETE FROM DEPT..;

SELECT...
  FROM EMP@SALES...;

COMMIT;
```

©LXD



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

©LXD

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

©LXD

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

©LXD

INFORMATION RETRIEVAL: TODAY

2016E = Milestone Year for 'Traditional' Live Streaming on Social Networks… NFL Live Broadcast TV of Thursday Night Football on Twitter (Fall 2016)

Hypothetical Mock-Up

Complete Sports Viewing Platform =

Live Broadcast + Analysis + Scores + Replays + Notifications + Social Media Tools

Tune-In Notifications, Game Reminders, Breaking Actions
Scoreboard Allows Fans to Follow Play-by-Play
Vertical View = Live Broadcast + Tweets Dashboard for Social Media Engagement
Horizontal View = Unencumbered, Full-Screen, TV-like Viewing Experience
Tweets Engage Fans in Real-Time Conversation
Professional Commentary and Analysis
Toggle Between Tweets from Stadium / Nearby / All
@KPCB

KPCB INTERNET TRENDS 2016 | PAGE 96

©LXD

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.

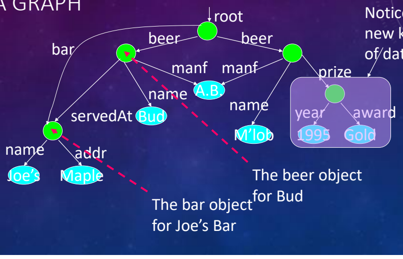
©LXD

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

©LXD

SPECIALTY DATABASES - DATA GRAPH

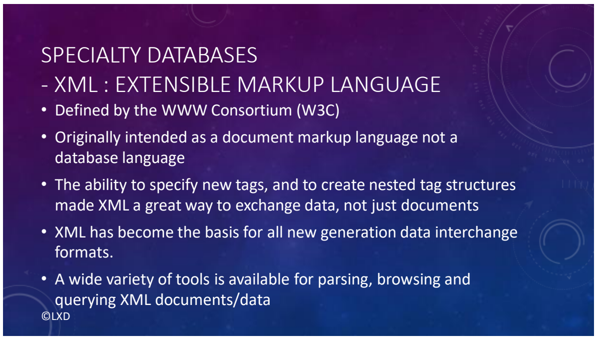


Notice a new kind of data.

The beer object for Bud

The bar object for Joe's Bar

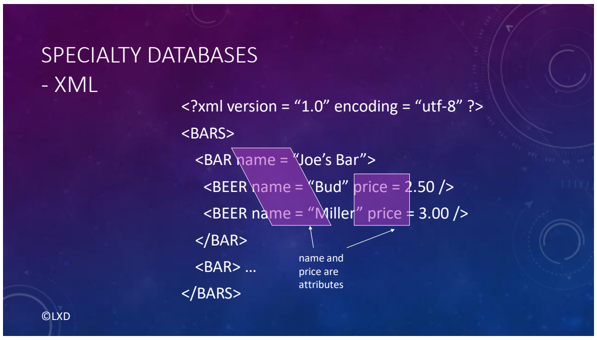
©LXD



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

©LXD

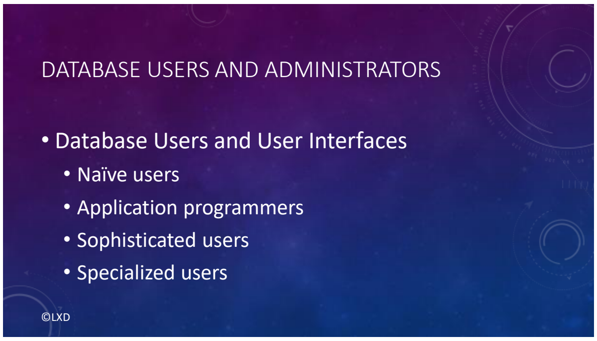


SPECIALTY DATABASES - XML

```
<?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

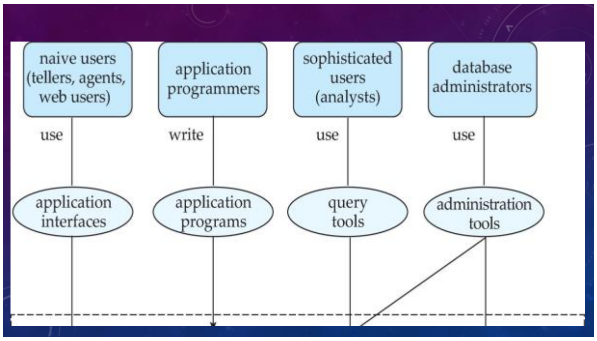
©LXD



DATABASE USERS AND ADMINISTRATORS

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

©LXD



naive users (tellers, agents, web users)

application programmers

sophisticated users (analysts)

database administrators

use

write

use

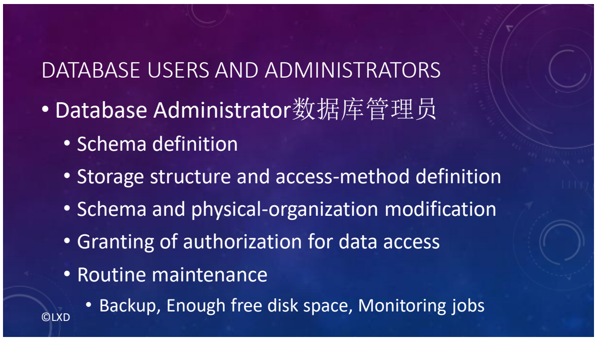
use

application interfaces

application programs

query tools

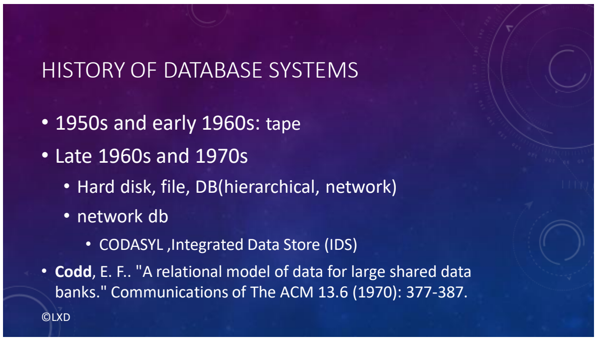
administration tools



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

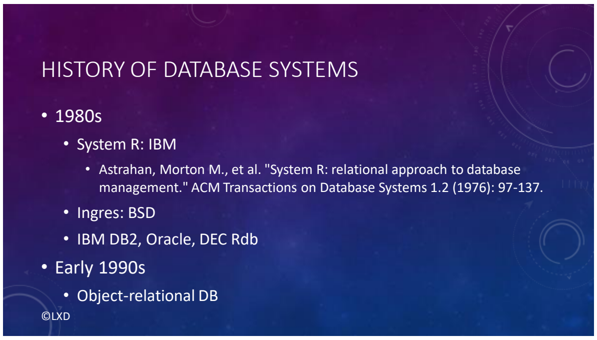
©LXD



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.

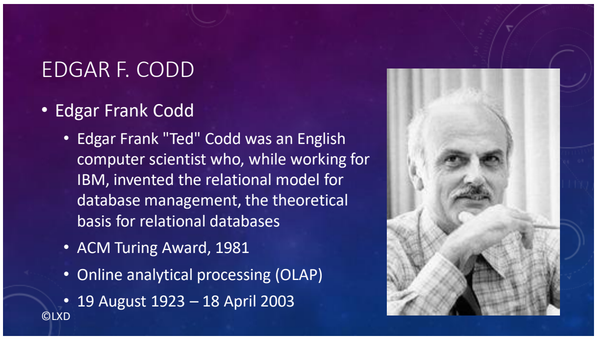
©LXD



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

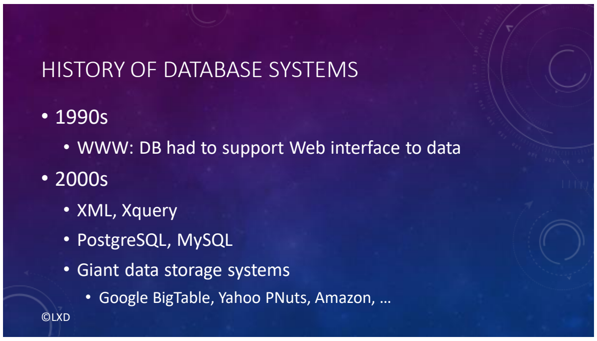
©LXD



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

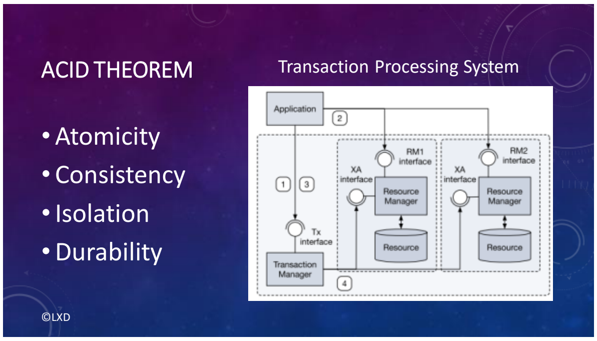
©LXD



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, …

©LXD

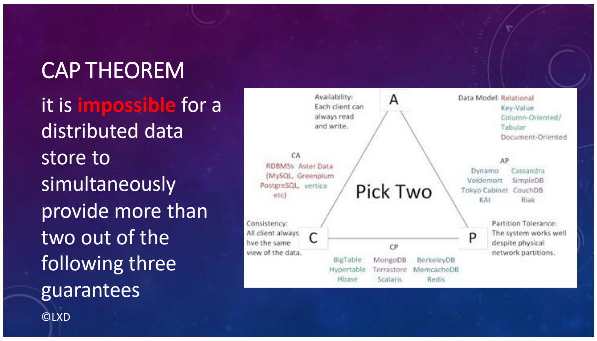


ACID THEOREM

- Atomicity
- Consistency
- Isolation
- Durability

Transaction Processing System

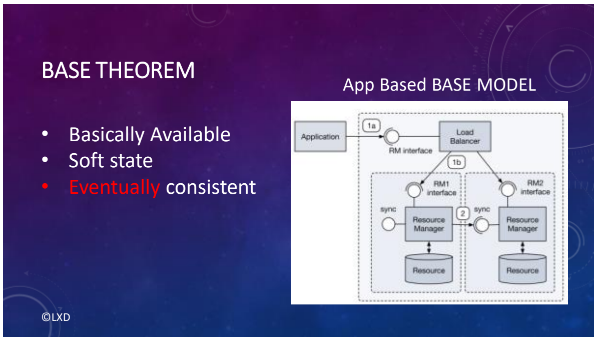
©LXD



CAP THEOREM

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

©LXD



BASE THEOREM

- Basically Available
- Soft state
- Eventually consistent

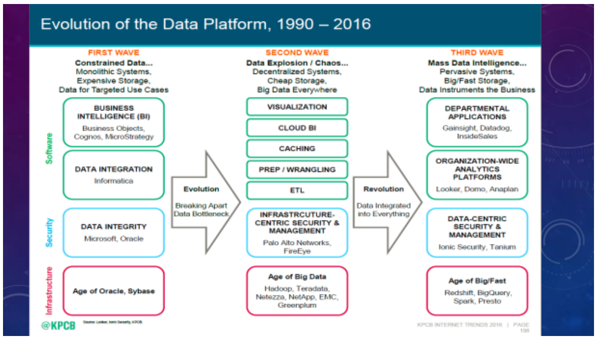
App Based BASE MODEL

©LXD

TODAY

Data as a Platform

©LXD



Evolution of the Data Platform, 1990 – 2016

FIRST WAVE
Constrained Data…
Monolithic Systems,
Expensive Storage,
Data for Targeted Use Cases

SECOND WAVE
Data Explosion / Chaos…
Decentralized Systems,
Cheap Storage,
Big Data Everywhere

THIRD WAVE
Mass Data Intelligence…
Pervasive Systems,
Big/Fast Storage,
Data Instruments the Business

	FIRST WAVE		SECOND WAVE		THIRD WAVE
Software	**BUSINESS INTELLIGENCE (BI)** Business Objects, Cognos, MicroStrategy **DATA INTEGRATION** Informatica	**Evolution** Breaking Apart Data Bottleneck	**VISUALIZATION** **CLOUD BI** **CACHING** **PREP / WRANGLING** **ETL**	**Revolution** Data Integrated into Everything	**DEPARTMENTAL APPLICATIONS** Gainsight, Datadog, InsideSales **ORGANIZATION-WIDE ANALYTICS PLATFORMS** Looker, Domo, Anaplan
Security	**DATA INTEGRITY** Microsoft, Oracle		**INFRASTRCUTURE-CENTRIC SECURITY & MANAGEMENT** Palo Alto Networks, FireEye		**DATA-CENTRIC SECURITY & MANAGEMENT** Ionic Security, Tanium
Infrastructure	**Age of Oracle, Sybase**		**Age of Big Data** Hadoop, Teradata, Netezza, NetApp, EMC, Greenplum		**Age of Big/Fast** Redshift, BigQuery, Spark, Presto

@KPCB Source: Looker, Ionic Security, KPCB

KPCB INTERNET TRENDS 2016 | PAGE 198

SUMMARY

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

©LXD

Q&A?

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

THANKS!

leexudong@nankai.edu.cn