

系统架构设计师

第6章 数据库设计基础

授课：王建平

目录

1

数据库基本概念

2

关系型数据库概念

3

关系代数

4

SQL语句

5

规范化理论

6

数据库设计过程

7

并发控制

8

分布式数据库

9

数据库安全、数据仓库与商业智能

目录

- 1 数据库基本概念
- 2 关系型数据库概念
- 3 关系代数
- 4 SQL语句
- 5 规范化理论
- 6 数据库设计过程
- 7 并发控制
- 8 分布式数据库
- 9 数据库安全、数据仓库与商业智能

数据库基本概念

◆基本概念

◆数据(Data)是数据库中存储的基本对象，是描述事物的符号记录。数据的种类：文本、图形、图像、音频、视频等。

◆数据库(Database, DB)数据库是统一管理的、长期储存在计算机内的，有组织的相关数据的集合。

◆数据库的基本特征：（★）

- ✓ 数据按一定的数据模型组织、描述和储存
- ✓ 数据间联系密切、冗余度较小
- ✓ 数据独立性较高
- ✓ 易扩展
- ✓ 可为各种用户共享

◆关系型数据库和文件系统的区别（★★）

	设计难度。	数据冗余程度。	数据架构。	应用扩展性。
关系型数据库。	针对特定应用系统设计，难度较大。	遵守数据库范式，数据冗余较小。	以数据库为中心组织，管理数据。	数据库独立于应用系统，数据库系统接口标准化，易于在不同应用之间共享数据。
文件系统。	针对特定应用系统设计，难度较小。	可能在多个文件中复制相同的数据属性，数据冗余较大。	以应用为中心管理数据。	符合特定应用系统要求的文件数据很难在不同的应用系统之间共享。

数据模型

◆数据模型是数据特征的抽象，它从抽象层次上描述了系统的静态特征、动态行为和约束条件，为数据库系统的信息表示与操作提供一个抽象的框架。

数据模型的三要素：（★）

- ✓ 数据结构：对象类型的集合，是对系统静态特性的描述。
- ✓ 数据操作：对数据库中各种对象的实例(值)允许执行的操作集合，包括操作及操作规则。如检索、插入、删除和修改，操作规则有优先级等。
- ✓ 数据的约束条件：一组完整性规则的集合。对具体的应用数据必须遵循特定的语义约束条件。

◆常见的基本数据模型：（★）

- ✓ 层次模型：用树形结构表示数据与数据间的联系。
- ✓ 网状模型：采用网络结构表示数据间联系的数据模型。
- ✓ 关系模型：二维表。
- ✓ 面向对象数据模型(第三代数据库系统)

数据库管理系统

◆数据库管理系统(DBMS)

DBMS 是数据库系统的核心软件，是由一组相互关联的数据集合和一组用以访问这些数据的软件组成。它是一种解决如何科学地组织和储存数据如何高效地获取和维护数据的系统软件。

DBMS 功能主要包括数据定义，数据库操作，数据库运行管理，数据组织、存储和管理，数据库的建立和维护。（★）

(1)数据定义

DBMS 提供数据定义语言 (DDL)，可以对数据库的结构进行描述，包括外模式、模式和内模式的定义；数据库的完整性定义；安全保密定义，如口令、级别和存取权限等。这些定义存储在数据字典中，是 DBMS 运行的基本依据。

(2)数据库操作

DBMS 向用户提供数据操纵语言 (DML)，实现对数据库中数据的基本操作，如检索、插入、修改和删除。

(3)数据库运行管理。数据库在运行期间，多用户环境下的并发控制、安全性检查和存取控制、完整性检查和执行、运行日志的组织管理、事务管理和自动恢复等都是 DBMS 的重要组成部分。这些功能可以保证数据库系统的正常运行。

(4) 数据组织、存储和管理。

(5) 数据库的建立和维护。

数据库的建立和维护，包括数据库的初始建立、数据的转换、数据库的转储和恢复、数据库的重组和重构、性能监测和分析等。

数据库管理系统

◆DBMS 的特点

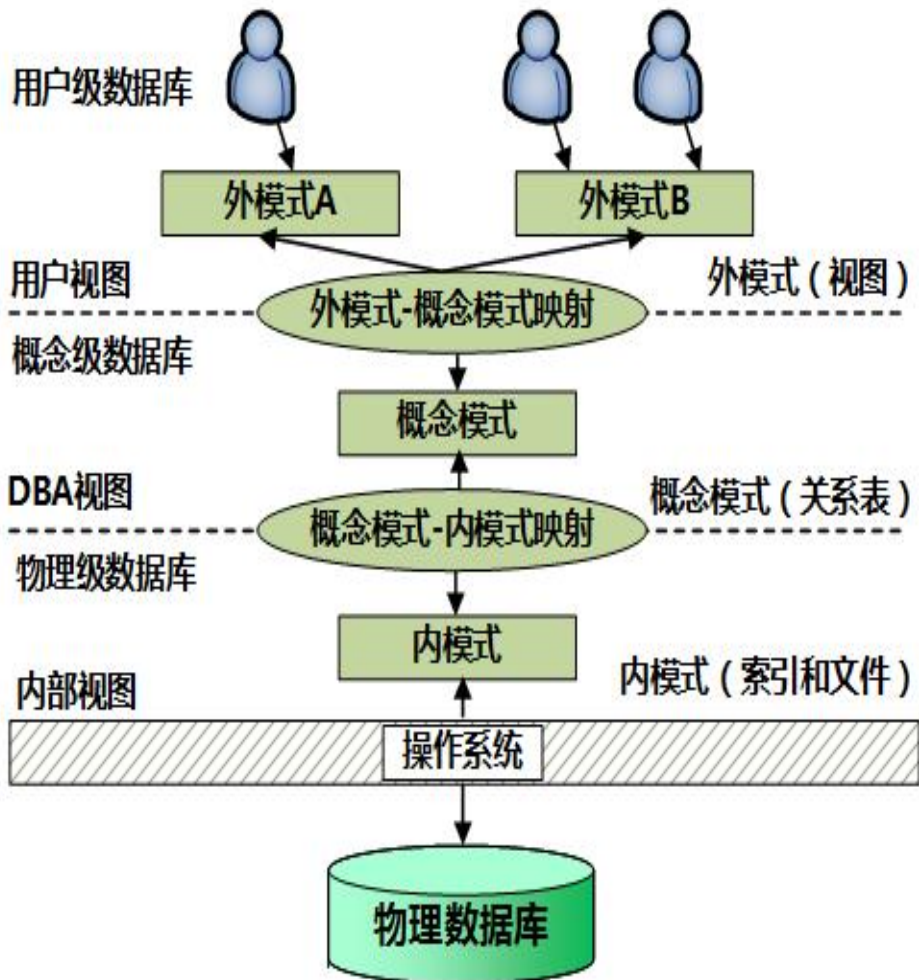
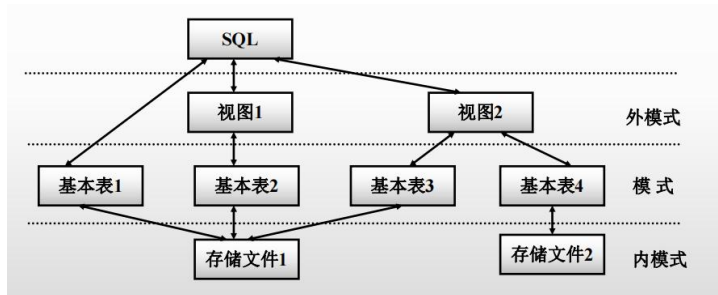
(1) 数据结构化且统一管理。(2) 有较高的数据独立性。(3) 数据控制功能。

DBMS 提供了数据控制功能，以适应共享数据的环境。数据控制功能包括对数据库中数据的安全性、完整性、并发和恢复的控制。

- ✓ 数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改或破坏。这样，用户只能按规定对数据进行处理，例如，划分了不同的权限，有的用户只有读数据的权限，有的用户有修改数据的权限，用户只能在规定的权限范围内操纵数据库。
- ✓ 数据的完整性是指数据库正确性和相容性，是防止合法用户使用数据库时向数据库加入不符合语义的数据。保证数据库中数据是正确的，避免非法的更新。
- ✓ 并发控制是指在多用户共享的系统中，许多用户可能同时对同一数据进行操作。DBMS的并发控制子系统负责协调并发事务的执行，保证数据库的完整性不受破坏，避免用户得到不正确的数据。
- ✓ 故障恢复:数据库中的常见故障是事务内部故障、系统故障、介质故障及计算机病毒等。故障恢复主要是指恢复数据库本身，即在故障导致数据库状态不一致时，将数据库恢复到某个正确状态或一致状态。恢复的原理非常简单，就是要建立冗余 (redundancy) 数据。换句话说，确定数据库是否可恢复的方法就是其包含的每一条信息是否都可以利用冗余的存储在别处的信息重构。

三级模式两级映像

◆数据库三级模式两级映像 (★★★)



概念模式	外模式	内模式
概念模式是数据库中 全体数据的逻辑结构 和特征的描述，是所有用户的公共数据视图。一个数据库只有一个概念模式。	外模式（子模式、用户模式）用以描述 用户看到 或使用的哪部分数据的逻辑结构，用户根据外模式用数据操作语句或应用程序去操作数据库中的数据。	内模式定义的是存储记录的类型、存储域的表示以及存储记录的物理顺序，指引元、索引和存储路径等数据的存储组织。一个数据库只有一个内模式。

逻辑独立性

当模式改变时（例如增加新的关系，新的属性，改变属性的数据类型等），由数据库管理员对各个外模式/模式的映像做相应的改变，可以使外模式保持不变。应用程序是依据数据的外模式编写的，从而应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性。

物理独立性

当数据库的存储结构改变了，由数据库管理员对模式/内模式映像做响应的改变，可以使模式保持不变，从而应用程序也不必改变，保证了数据与程序的物理独立性，简称数据的物理独立性。

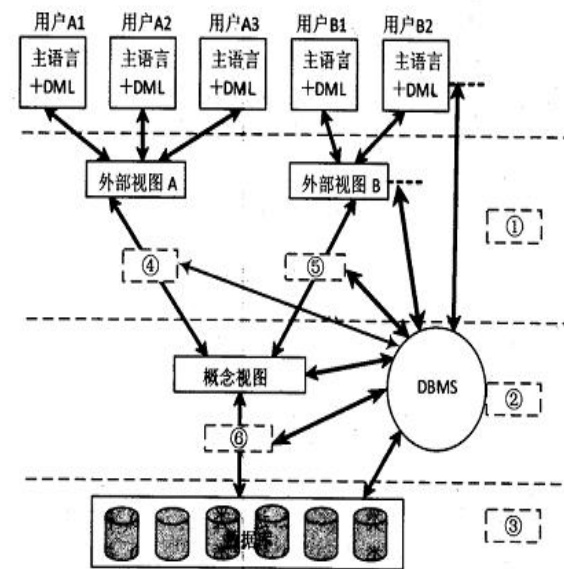
典型真题

数据库的产品很多，尽管它们支持的数据模型不同，使用不同的数据库语言，而且数据的存储结构也各不相同，但体系结构基本上都具有相同的特征，采用“三级模式和两级映像”，如下图所示，图中①，②，③分别代表数据库系统中（40），图中④，⑤，⑥分别代表数据库系统中（41）。

- (40) A.物理层、逻辑层、视图层
B.逻辑层、物理层、视图层
C.视图层、物理层、逻辑层
D.视图层、逻辑层、物理层.

- (41) A.外模式/内模式映射、外模式/内模式映射、概念模式/内模式映射
B.外模式/概念模式映射、外模式/概念模式映射、概念模式/内模式映射.
C.概念模式/内模式映射、概念模式/内模式映射、外模式/内模式映射
D.外模式/内模式映射、外模式/内模式映射、概念模式/外模式映射

【答案】D B



典型真题

在采用三级模式结构的数据库系统中，如果对数据库中的表Emp创建聚簇索引，那么改变的是数据库的（ ）。

- A. 模式 B. 内模式 C. 外模式 D. 用户模式

试题分析

数据库采用三级模式的结构：

- （1）概念模式也称模式，是数据库中全部数据的整体逻辑结构的描述。
 - （2）外模式也称用户模式或子模式，是用户与数据库系统的接口，是用户用到的那部分数据的描述。
 - （3）内模式也称存储模式，是数据库在物理存储方面的描述，定义所有内部记录类型、索引和文件的组织方式，以及数据控制方面的细节。
- 聚簇索引也叫簇类索引，是一种对磁盘上实际数据重新组织以按指定的一个或多个列的值排序，它改变的是数据库的内模式。

参考答案：B

典型真题

数据的物理独立性和逻辑独立性分别是通过修改()来完成的。

- A.外模式与内模式之间的映像、模式与内模式之间的映像
- B.外模式与内模式之间的映像、外模式与模式之间的映像
- C.外模式与模式之间的映像、模式与内模式之间的映像
- D.模式与内模式之间的映像、外模式与模式之间的映像.

参考答案：D

NOSQL数据库

◆常见的NoSQL数据库按存储方式，可分为文档存储、键值存储、列存储和图存储。（★★）

分 类	典型产品	应 用 场 景	优 点	缺 点
文档存储	MongoDB、CouchDB	Web 应用，存储面向文档和半结构化数据	结构灵活，可以根据 value 构建索引	缺乏统一的查询语法；无事务处理能力
键值存储	Memcached、Redis	内容缓存，如会话、配置文件、参数等	扩展性好，灵活性强，大量操作时性能高	数据无结构化，通常被当成字符串或者二进制数据，通过键查询值
列存储	Bigtable、HBase、Cassandra	分布式数据存储和管理	可扩展性强，查找速度快，复杂性低	功能局限：不支持事务的强一致性
图存储	Neo4j、OrientDB	社交网络、推荐系统、专注于构建系统图谱	支持复杂的图形算法	复杂性高，只能支持一定的数据规模

NOSQL数据库

◆NOSQL框架 (★★)

NOSQL整体框架分为4层，由下至上分为数据持久层 (Data Persistence)、数据分布层、数据逻辑模型层 (Data Logical Model) 和接口层 (Interface)，层次之间相辅相成，协调工作。

(1) 数据持久层定义了数据的存储形式，主要包括基于内存、硬盘、内存和硬盘接口、订制可插拔4种形式。基于内存形式的数据存取速度最快，但可能会造成数据丢失；基于硬盘的数据存储可能保存很久，但存取速度慢于基于内存形式的数据；内存和硬盘相结合的形式，结合了前两种形式的优点，既保证了速度，又保证了数据不丢失；订制可插拔则保证了数据存取具有较高的灵活性。

(2) 数据分布层定义了数据是如何分布的，相对于关系型数据库，NOSQL可选的机制比较多，主要有3种形式：一是CAP支持，可用于水平扩展；二是多数据中心支持，可以保证在横跨多数据中心时也能够平稳运行；三是动态部署支持，可以在运行着的集群中动态地添加或删除结点。

(3) 数据逻辑层表述了数据的逻辑表现形式。

(4) 接口层为上层应用提供了方便的数据调用接口，提供的选择远多于关系型数据库。接口层提供了5种选择：Rest、Thrift、Map/Reduce、Get/Put、特定语言API，使得应用程序和数据库的交互更加方便。

NOSQL分层架构并不代表每个产品在每一层只有一种选择。相反，这种分层设计提供了很大的灵活性和兼容性，每种数据库在不同层面可以支持多种特性。

◆NOSQL数据库在以下这几种情况比较适用： (★★)

数据模型比较简单；需要灵活性更强的IT系统；对数据库性能要求较高；不需要高度的数据一致性；

目录

- 1 数据库基本概念
- 2 关系型数据库概念
- 3 关系代数
- 4 SQL语句
- 5 规范化理论
- 6 数据库设计过程
- 7 并发控制
- 8 分布式数据库
- 9 数据库安全、数据仓库与商业智能

关系表类型

◆关系表类型（★★）

关系表类型	要点
基本关系（基本表或基表）	实际存在的表，实际存储数据的逻辑表示。
查询表	查询结果对应的表。
视图表	由基表或其他视图表导出的表，本身不独立存储，数据库只存放它的定义，常称为虚表。

◆视图：是一个虚拟表，其内容由查询定义（仅保存在SQL查询语句）同真实的表一样，视图也有一系列的行列数据，但是视图并没有真正存储这些数据，而是通过查询原始表，动态生成所需要的数据。

（★★）

◆优点：（★★★★）

- ① 简化用户操作
- ② 视图可以使得用户以多种角度看待同一数据
- ③ 对重构数据库提供一定程度的逻辑独立性
- ④ 对机密数据提供了安全保护

◆物化视图：不是传统意义的虚拟表，是实体化的视图，本身存数据，对原始数据更新时，物化视图也会更新。（★★）

（频繁用到的一些表）

典型真题

在数据库的安全机制中，通过建立()使用户只能看到部分数据，从而保护了其它数据的安全性。

- A.索引
- B.视图
- C.触发器
- D.存储过程

参考答案：B

解析：只能看到部分数据，就是给用户看的视图。

典型真题

在数据库系统中，视图是一个（ ）。

- A. 真实存在的表，并保存了待查询的数据
- B. 真实存在的表，只有部分数据来源于基本表
- C. 虚拟表，查询时只能从一个基本表中导出
- D. 虚拟表，查询时可以从一个或者多个基本表或视图中导出

试题分析

视图是从一个或几个基本表(或视图)导出的虚拟表，其内容由查询定义。同真实的表一样，视图包含一系列带有名称的列和行数据。但是，视图并不在数据库中以存储的数据值集形式存在。行和列数据来自定义视图的查询所引用的表，并且在引用视图时动态生成。

参考答案：D

关系型数据库基本概念

◆关系型数据库表基本概念（★★★）

概念	名词解释
关系	可以理解为一张二维表，每个关系都具有一个关系名，就是通常说的表名。
元组	可以理解为二维表中的一行，在数据库中经常被称为记录。
属性	可以理解为二维表中的一列，在数据库中经常被称为字段。
域	属性的取值范围，也就是数据库中某一列的取值限制。
关键字	一组可以唯一标识元组的属性，数据库中常称为主键，由一个或多个列组成。
关系模式	指对关系的描述。其格式为：关系名（属性1，属性2，……，属性N），在数据库中成为表结构。

关系型数据库基本概念

◆关系型数据库表基本概念 (★★★)

The diagram illustrates the basic concepts of a relational database table. It features a table with five columns and four rows. Annotations include: '属性' (Attribute) pointing to the column headers; '域' (Domain) pointing to the '平均成绩' (Average Grade) column; '行 (记录)' (Row (Record)) pointing to the data rows; and '列' (Column) pointing to the column headers. A bracket on the left labeled '元组' (Tuple) groups the rows.

课程编号	课程名称	学时	学期	平均成绩
J001	数据库基础	36	1	78
J003	C语言	60	1	81
Z004	操作系统	44	2	67
X001	编译原理	40	2	66

完整性约束

◆数据库表完整性约束 (★★)

☆实体完整性。实体完整性是指实体的主属性不能取空值。(唯一非空)

主键

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

☆参照完整性。在关系数据库中主要是值得外键参照的完整性。若A关系中的某个或者某些属性参照B或其他几个关系中的属性，那么在关系A中该属性要么为空，要么必须出现B或者其他的关系的对应属性中。

D#	Dname	Dean
01	机电	李三
02	能源	李四
03	计算机	李五
04	自动控制	李六

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402
98050206	孙六	男	21	05	980502
98060207	吴七	女	20		980602

完整性约束

☆用户定义完整性。用户定义完整性反映的某一个具体应用所对应的数据必须满足一定的约束条件。如（社保上下限、男女）年龄（1-50）等。

Student					
S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402
98040204	詹姆斯·卡梅隆	男	20	04	980402
98040206	王六	01	21	04	980402
98040207	王七	女	200	04	980402

◆触发器：在用户关系表上的一类由事件驱动的特殊过程，可以进行更为复杂的检查和操作，具有更精细和更强大的数据控制能力。（只有增删改能激活触发器，select无法激活触发器，激活后由数据库服务器自动执行）触发器中不包括事务控制语句。（★★★）

◆存储过程：将具有特定功能的一段SQL语句（多于一条）在数据库服务器上预先定义并编译，以供应用程序调用，该段SQL程序可被定义为存储过程。（★★★）

存储过程里可以包含流程控制。被编译后存在数据库中。

典型真题

将具有特定功能的一段SQL语句（多于一条）在数据库服务器上预先定义并编译，以供应用程序调用，该段SQL程序可被定义为（）。

A.触发器 B.视图 C.触发器 D.存储过程

参考答案：D

目录

- 1 数据库基本概念
- 2 关系型数据库概念
- 3 关系代数
- 4 SQL语句
- 5 规范化理论
- 6 数据库设计过程
- 7 并发控制
- 8 分布式数据库
- 9 数据库安全、数据仓库与商业智能

关系代数 (★★★)

运算符		含义	名词解释
集合运算符	\cup	并	关系R与S的并是由属于R或属于S的元组构成的集合。
	$-$	差	关系R与S的差是由属于R但不属于S的元组构成的集合
	\cap	交	关系R与S的交是由属于R同时又属于S的元组构成的集合。
	\times	笛卡尔积	两个元组分别为n目和m目的关系R和S的笛卡尔积是一个(n+m)列的元组的集合。元组的前n列是关系R的一个元组，后m列是关系S的一个元组。

关系代数 (☆☆☆)

R

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1

S

A	B	C
a1	b2	c2
a1	b3	c2
a2	b2	c1

$R \cup S$

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1
a1	b3	c2

$R - S$

A	B	C
a1	b1	c1

$R \cap S$

A	B	C
a1	b2	c2
a2	b2	c1

$R \times S$

$R \times S$

R.A	R.B	R.C	S.A	S.B	S.C
a1	b1	c1	a1	b2	c2
a1	b1	c1	a1	b3	c2
a1	b1	c1	a2	b2	c1
a1	b2	c2	a1	b2	c2
a1	b2	c2	a1	b3	c2
a1	b2	c2	a2	b2	c1
a2	b2	c1	a1	b2	c2
a2	b2	c1	a1	b3	c2
a2	b2	c1	a2	b2	c1

关系代数 (★★★★)

设关系R、S分别是下图的 (a) 和 (b) , 则 $R \div S$ 的结果为 () 。

1、找相同部分为BC

2、R里的其它元素就为象集

a1的象集为 (b1、c2) (b2, c3) (b2, c1)

a2的象集为 (b3, c7) , (b2, c3)

a3的象集为 (b4, c6)

a4的象集为 (b6, c6)

3、只有a1包含了s和r里公共的部分, 所以

$R \div S = (a1)$

R		
A	B	C
a ₁	b ₁	c ₂
a ₂	b ₃	c ₇
a ₃	b ₄	c ₆
a ₁	b ₂	c ₃
a ₄	b ₆	c ₆
a ₂	b ₂	c ₃
a ₁	b ₂	c ₁

(a)

S		
B	C	D
b ₁	c ₂	d ₁
b ₂	c ₁	d ₁
b ₂	c ₃	d ₂

(b)

$R \div S$
A
a ₁

(c)

关系代数 (★★★)

运算符		含义	名词解释
专门的关系运算符	σ	选择	取得关系R中符合条件的行
	π	投影	取得关系R中符合条件的列
	$=$ \bowtie	连接	<ul style="list-style-type: none">● 等值连接：关系R、S,取两者笛卡尔积中属性值相等的元组。● 自然连接：一种特殊的等值连接，它要求比较的属性列必须是相同的属性组，并且把结果中重复属性去掉。注意与笛卡尔积的区别。

关系代数 (★★★)

例如：学生表 (Student)

Sno	Sname	Ssex	Sage	Sdept
202015001	李阳	女	18	计算机
202015113	王明	男	18	信息安全
202015254	刘立	男	19	计算机
202015117	ZHANG	男	29	哲学

关系代数 (★★★)

例如：查询年龄小于20岁的学生 $\sigma_{\text{Sage} < 20}(\text{Student})$

Sno	Sname	Ssex	Sage	Sdept
202015001	李阳	女	18	计算机
202015113	王明	男	18	信息安全
202015254	刘立	男	19	计算机

例如：查询学生的姓名和所在系 $\pi_{\text{Sname}, \text{Sdept}}(\text{Student})$

Sname	Sdept
李阳	计算机
王明	信息安全
刘立	计算机
ZHANG	哲学

关系代数 (★★★)

R

A	B	C
a1	b1	5
a1	b2	6
a2	b3	8
a2	b4	12

R.B=S.B

A	R.B	C	S.B	E
a1	b1	5	b1	3
a1	b2	6	b2	7
a2	b3	8	b3	10
a2	b3	8	b3	2

S

B	E
b1	3
b2	7
b3	10
b3	2
b5	2

R \bowtie **S**

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2

关系代数 (★★★)

- (1) 左外连接：R和S进行自然连接时，只把R中舍弃的元组放到新关系中。
- (2) 右外连接：R和S进行自然连接时，只把S中舍弃的元组放到新关系中。
- (3) 完全外连接：R和S进行自然连接时，只把R和S中舍弃的元组都放到新关系中。

R关系		
A1	A2	A3
a	b	c
b	a	d
c	d	d
d	f	g

S关系		
A1	A2	A4
a	z	a
b	a	h
c	d	d
d	s	c

自然连接

A1	A2	A3	A4
b	a	d	h
c	d	d	d

左外连接

A1	A2	A3	A4
a	b	c	null
b	a	d	h
c	d	d	d
d	f	g	null

右外连接

A1	A2	A3	A4
a	z	null	a
b	a	d	h
c	d	d	d
d	s	null	c

完全外连接

A1	A2	A3	A4
a	b	c	null
b	a	d	h
c	d	d	d
d	f	g	null
a	z	null	a
d	s	null	c

典型真题

试题分析

本题结果是

R 与 S 的左外联接

A1	A2	A3	A4
1	2	3	null
2	1	4	8
3	4	4	4
4	6	7	null

R 与 S 的右外联接

A1	A2	A3	A4
1	9	null	1
2	1	4	8
3	4	4	4
4	8	null	3

从运算的结果可以看出 R 与 S 的左外联接、右外联接的元组个数分别为 4 、 4 。

参考答案：B、C

典型真题

给定关系 $R(A, B, C, D)$ 和关系 $S(C, D, E)$ ，对其进行自然连接运算 $R \bowtie S$ 后的属性列为（ ）个；与 $\sigma_{R.B > S.E}(R \bowtie S)$ 等价的关系代数表达式为（ ）。

A. 4 B. 5 C. 6 D. 7

A. $\sigma_{2 > 7}(R \times S)$

B. $\pi_{1,2,3,4,7}(\sigma_{2' > 7' \wedge 3=5 \wedge 4=6}(R \times S))$

C. $\Sigma_{2' > 7'}(R \times S)$

D. $\pi_{1,2,3,4,7}(\sigma_{2 > 7 \wedge 3=5 \wedge 4=6}(R \times S))$

试题分析

自然连接运算去掉重复的属性列，结果为5列：A，B，C，D，E。

参考答案：B、D

目录

- 1 数据库基本概念
- 2 关系型数据库概念
- 3 关系代数
- 4 SQL语句
- 5 规范化理论
- 6 数据库设计过程
- 7 并发控制
- 8 分布式数据库
- 9 数据库安全、数据仓库与商业智能

SQL语句 (★)

SQL 语言核心 9 个动词

SQL 功 能	动 词
数 据 查 询	SELECT
数 据 定 义	CREATE, DROP, ALTER
数 据 操 纵	INSERT, UPDATE DELETE
数 据 控 制	GRANT, REVOKE

SQL语言中，Grant为授权语句，其语法规则为：

Grant <权限> on 表名 to 用户 with grant option。

权限字段包括：select/insert/update/delete/all等，就是英文意思增删改查等权限。

并且允许授权给他人，加上with grant option。

创建表 create table;

指定主键 primary key();

指定外键 foreign key();

修改表 alter table;

删除表 drop table;

索引 index; 视图 view;

SQL语句 (★)

```
CREATE TABLE S (Sno CHAR(5)NOT NULL UNIQUE,  
Sname CHAR(30)UNIQUE,  
Status CHAR(8),  
City CHAR(20)  
PRIMARY KEY(Sno));  
ALTER TABLE S ADD Zap CHAR(6);  
DROP TABLE Student  
CREATE UNIQUE INDEX S-SNO ON S(Sno);  
CREATE VIEW CS-STUDENT
```

SQL语句 (★★)

◆数据库查询 select...from...where;

分组查询group by,分组时要注意select后的列名要适应分组, having为分组查询附加条件:

[例1] 求各个课程号及相应的选课人数。

```
SELECT Cno, COUNT(Sno) AS 选课人数
FROM SC
GROUP BY Cno;
```

Cno	选课人数
1	22
2	34
3	44
4	33
5	48

[例2] 求选修人数大于40人的课程编号及人数

```
SELECT Cno, COUNT(Sno) AS 选课人数
FROM SC
GROUP BY Cno
HAVING COUNT(Sno) > 40;
```

Cno	选课人数
3	44
5	48

◆字符串匹配like,%匹配多个字符串, _匹配任意一个字符串: select *from t1where sname like'a_'

◆数据库插入insert into...values(): insert into t1 values('a',66)

◆数据库删除delete from...where:delete t1 where sno=4

◆数据库修改update...set...where:

update t1 set sname='aa'where sno=3排序order by,默认为升序, 降序要加关键DESC:

select*from t1 order by sno desc

◆SELECT[ALL|DISTINCT]<目标列表表达式>[,<目标列表表达式>]...FROM<表名或视图名>[,<表名或视图名>][WHERE<条件表达式>]

[GROUP BY<列名1>[HAVING<条件表达式>]][ORDER BY<列名2>[ASC|DESC]...]

典型真题

若要将部门表Demp中name列的修改权限赋予用户Ming，并允许Ming将该权限授予他人，实现的SQL语句如下：

GRANT () ON TABLE Demp TO Ming ()

A、SELECT (name)

B、UPDATE(name).

C、INSERT(name)

D、ALL PRIVILEGES(name)

A、FOR ALL

B、CASCADE

C、WITH GRANT OPTION.

D、WITH CHECK OPTION

答案：B C

查询优化问题 (★)

◆典型的启发式规则：

- 1.选择运算应尽可能先做。在优化策略中这是最重要、最基本的一条
- 2.把投影运算和选择运算同时进行。避免重复扫描关系。
- 3.把投影同其前或其后的双目运算结合起来
- 4.把某些选择同在它前面要执行的笛卡尔积结合起来成为一个连接运算
- 5.找出公共子表达式：如果重复出现的子表达式的结果不是很大的关系并且从外存中读入这个关系比计算该子表达式的时间少得多，则先计算一次公共子表达式并把结果写入中间文件是合算

典型真题

在关系R(A1, A2, A3)和S(A1, A3, A4)上进行关系运算的4个等价的表达式E1、E2、E3和E4如下所示：
如果严格按照表达式运算顺序，则查询效率最高的是()。

$$E_1 = \Pi_{A_1, A_4} (\sigma_{A_2 < '2022' \wedge A_4 = '95'} (R \bowtie S))$$

$$E_2 = \Pi_{A_1, A_4} (\sigma_{A_2 < '2022'} (R) \bowtie \sigma_{A_4 = '95'} (S))$$

$$E_3 = \Pi_{A_1, A_4} (\sigma_{A_2 < '2022' \wedge R.A_3 = S.A_3 \wedge A_4 = '95'} (R \times S))$$

$$E_4 = \Pi_{A_1, A_4} (\sigma_{R.A_3 = S.A_3} (\sigma_{A_2 < '2022'} (R) \times \sigma_{A_4 = '95'} (S)))$$

A.E1 B.E2. C.E3 D.E4

典型真题

参考答案：B

此题是关系代数表达式查询问题，相同结果下，自然连接的效率优于笛卡尔积。

备选答案中，E1和E2为自然连接，故优先选择A、B选项。E3和E4缺少了 $R.A1=S.A1$ 这个等值条件。

优化SQL语句，减少比较次数是提高查询效率的有效方法。

A选项和B选项相比，B选项将可以对子表做的操作先做了，再做连接，最后投影，这是效率最高的一种方法。

目录

- 1 数据库基本概念
- 2 关系型数据库概念
- 3 关系代数
- 4 SQL语句
- 5 规范化理论
- 6 数据库设计过程
- 7 并发控制
- 8 分布式数据库
- 9 数据库安全、数据仓库与商业智能

规范化理论 (★★★)

◆ 例如有两个关系模式：

学生（学号、姓名、性别、身份证号、教师编号）

教师（教师编号、姓名、工资）

超键	在关系中能唯一标识元组的属性集称为关系模式的超键。一个属性可以为作为一个超键,多个属性组合在一起也可以作为一个超键。例如学生关系模式中的（学号）、（身份证号）、（学号，姓名）、（身份证号、性别）等。
候选键	不含有多余属性的超键称为候选键，是超键的最小子集。例如，（学号）、（身份证号）
主键	用户挑选出来做元组标识的一个候选键称为主键。例如，我们通常选择（学号）作为主键。
外键	如果关系模式R中的某些属性集不是R的主键，而是关系模式S的主键，则这个属性集对模式R而言是外键。例如，（教师编号）是学生关系模式的外键。
主属性	包含在任何一个候选键中的属性称为主属性，否则称为非主属性。

规范化理论 (★★★)

例如：关系模式Student (Sno, Sdept, Mname, Cno, Grade)

名称	定义
完全函数依赖	$(Sno, Cno) \rightarrow Grade$ 是完全函数依赖。
部分函数依赖	$(Sno, Cno) \rightarrow Sdept$ 是部分函数依赖。
传递依赖	$Sno \rightarrow Sdept, Sdept \rightarrow Mname$, 则称Sno传递依赖于Mname。

Armstrong公理 (★★★★)

◆Armstrong公理：从已知的一些函数依赖，可以推导出另外一些函数依赖，这就需要一系列推理规则，这些规则常被称作“Armstrong 公理”。

设关系式 $R(U, F)$ ， U 是关系模式 R 的属性集， F 是 U 上一组函数依赖，则有以下三条推理规则：

A1自反律：若 $Y \subseteq X \subseteq U$ ，则 $X \rightarrow Y$ 为 F 所蕴含；

A2增广律：若 $X \rightarrow Y$ 为 F 所蕴含，且 $Z \subseteq U$ ，则 $XZ \rightarrow YZ$ 为 F 所蕴含；

A3传递律：若 $X \rightarrow Y$ ， $Y \rightarrow Z$ 为 F 所蕴含，则 $X \rightarrow Z$ 为 F 所蕴含。

根据上面三条推理规则，又可推出下面三条推理规则：

合并规则：若 $X \rightarrow Y$ ， $X \rightarrow Z$ ，则 $X \rightarrow YZ$ 为 F 所蕴含；

伪传递规则：若 $X \rightarrow Y$ ， $WY \rightarrow Z$ ，则 $XW \rightarrow Z$ 为 F 所蕴含；

分解规则：若 $X \rightarrow Y$ ， $Z \subseteq Y$ ，则 $X \rightarrow Z$ 为 F 所蕴含。

典型真题

给定关系模式 $R\langle U, F \rangle$; 其中 U 为属性集, F 是 U 上的一组函数依赖, 那么 Armstrong 公理系统的增广律是指 ()。

- A. 若 $X \rightarrow Y$, $X \rightarrow Z$, 则 $X \rightarrow YZ$ 为 F 所蕴涵
- B. 若 $X \rightarrow Y$, $WY \rightarrow Z$, 则 $XW \rightarrow Z$ 为 F 所蕴涵
- C. 若 $X \rightarrow Y$, $Y \rightarrow Z$ 为 F 所蕴涵, 则 $X \rightarrow Z$ 为 F 所蕴涵
- D. 若 $X \rightarrow Y$, 为 F 所蕴涵, 且 $Z \subseteq U$, 则入 $XZ \rightarrow YZ$ 为 F 所蕴涵

典型真题

设 U 是关系模式 R 的属性集， F 是 R 上成立的只涉及 U 中属性的函数依赖集。函数依赖的推理规则有以下三条：

- ① A1自反律：若 $Y \subseteq X \subseteq U$ ，则 $X \rightarrow Y$ 为 F 所蕴含；
- ② A2增广律：若 $X \rightarrow Y$ 为 F 所蕴含，且 $Z \subseteq U$ ，则 $XZ \rightarrow YZ$ 为 F 所蕴含；
- ③ A3传递律：若 $X \rightarrow Y$ ， $Y \rightarrow Z$ 为 F 所蕴含，则 $X \rightarrow Z$ 为 F 所蕴含。

根据上面三条推理规则，又可推出下面三条推理规则：

- ④ 合并规则：若 $X \rightarrow Y$ ， $X \rightarrow Z$ ，则 $X \rightarrow YZ$ 为 F 所蕴含；
- ⑤ 伪传递规则：若 $X \rightarrow Y$ ， $WY \rightarrow Z$ ，则 $XW \rightarrow Z$ 为 F 所蕴含；
- ⑥ 分解规则：若 $X \rightarrow Y$ ， $Z \subseteq Y$ ，则 $X \rightarrow Z$ 为 F 所蕴含。

参考答案：D

规范化理论 (★★★)

◆设有一个关系模式R(SNAME, CNAME, TNAME, TADDRESS), 其属性分别表示学生姓名、课程名、任课教师姓名和任课教师地址。

◆不规范化带来的四大问题:

设有一个关系模式R(SNAME, CNAME, TNAME, TADDRESS), 其属性分别表示学生姓名、选修的课程名、任课教师姓名和任课教师地址。仔细分析一下, 就会发现这个模式存在下列存储异常的问题:

(1) 数据冗余: 数据被重复存储, 如某门课程有100个学生选修, 那么在R的关系中就要出现100个元组, 这门课程的任课教师姓名和地址也随之重复出现100次。

(2) 修改异常: 修改导致数据不一致, 如由于上述冗余问题, 当需要修改这个教师的地址时, 就要修改100个元组中的地址值, 否则就会出现地址值不一致的现象。

(3) 插入异常: 插入时异常, 如不知道听课学生名单, 这个教师的任课情况和家庭地址就无法进入数据库; 否则就要在学生姓名处插入空值。

(4) 删除异常: 删除了不该删除的数据, 如当只有一条记录时, 要删除这个学生选课信息, 会将课程名、教师名和教师地址都给删除了。

规范化理论 (★★★)



关系数据库设计的方法之一就是设计满足适当范式的模式，通常可以通过判断分解后的模式达到几范式来评价模式的规范化程度。

范式：符合某一种级别的关系模式的集合。

规范化理论 (★★★)

例：考虑含有属性A,B,C,D,E,F的关系，设关系有FD: $AB \rightarrow C$, $BC \rightarrow AD$, $D \rightarrow E$, $CF \rightarrow B$ ，则{A,B}的闭包{A,B}⁺是什么？

$\{A,B\}^+ = \{A,B,C,D,E\}$

1) 1NF

若关系模式R的每一个分量都是不可再分的数据项，则关系模式R属于第一范式。记为 $R \in 1NF$ 。

例如，供应者和它所提供的零件信息，关系模式FIRST和函数依赖集F如下：

FIRST(Sno, Sname, Status, City, Pno, Qty),

$F = \{Sno \rightarrow Sname, Sno \rightarrow Status, Status \rightarrow City, (Sno, Pno) \rightarrow Qty\}$

码：(Sno, Pno)

Sno	Sname	Status	City	Pno	Qty
S1	精益	20	天津	P1	200
S1	精益	20	天津	P2	300
S1	精益	20	天津	P3	480
S2	盛锡	10	北京	P2	168
S2	盛锡	10	北京	P3	500
S3	东方红	30	北京	P1	300
S3	东方红	30	北京	P2	280
S4	泰达	40	上海	P2	460

存在冗余度大、引起修改操作不一致、插入异常、删除异常

规范化理论 (★★★)

2) 2NF

若关系模式 $R \in 1NF$, 且每一个非主属性完全依赖于码, 则关系模式 $R \in 2NF$ 。换句话说, 当 1NF 消除了非主属性对码的部分函数依赖, 则称为 2NF。

例如: FIRST 关系中的码是 Sno、Pno, 而 $Sno \rightarrow Status$, 因此非主属性 Status 部分函数依赖于码, 故非 2NF 的。

将 FIRST 关系分解为:

$FIRST_1(Sno, Sname, Status, City) \in 2NF$

$FIRST_2(Sno, Pno, Qty) \in 2NF$

因为分解后的关系模式 $FIRST_1$ 的码为 Sno, 非主属性 Sname、Status、City 完全依赖于码 Sno, 所以属于 2NF;

关系模式 $FIRST_2$ 的码为 Sno、Pno, 非主属性 Qty 完全依赖于码, 所以也属于 2NF。

规范化理论 (★★★)

3) 3NF

若关系模式 $R(U, F)$ 中不存在这样的码 X , 属性组 Y 及非主属性 $Z (Z \notin Y)$ 使得 $X \rightarrow Y, (Y \text{ 不能推出 } X) Y \rightarrow Z$ 成立, 则关系模式 $R \in 3NF$ 。

即当2NF消除了非主属性对码的传递函数依赖, 则称为3NF。

$FIRST_1(Sno, Sname, Status, City) \in 2NF$, 中有 $Sno \rightarrow Status, Status \rightarrow City$,

所以 $FIRST_1 \notin 3NF$, 因为存在着非主属性 $City$ 传递依赖于码 Sno 。若此时将 $FIRST_1$ 继续分解为:

$FIRST_{11}(Sno, Sname, Status) \in 3NF$

$FIRST_{12}(Status, City) \in 3NF$

通过上述分解, 数据库模式 $FIRST$ 转换为

$FIRST_{11}(Sno, Sname, Status)$

$FIRST_{12}(Status, City)$

$FIRST_2(Sno, Pno, Qty)$

由于这3个子模式都达到了3NF, 因此称分解后的数据库模式达到了3NF。

可以证明, 3NF的模式必是2NF的模式。产生冗余和异常的两个重要原因是部分依赖和传递依赖。

规范化理论 (★★★)

4)BCNF(Boyce Codd Normal Form,巴克斯范式)

关系模式 $R \in 1NF$,若 $X \rightarrow Y$ 且 $Y \not\subseteq X$ 时, X 必含有码, 则关系模式 $R \in BCNF$ 。也就是说, 当3NF消除了主属性对码的部分函数依赖和传递函数依赖, 则称为BCNF。

结论: 一个满足BCNF的关系模式, 应有如下性质:

- (1)所有非主属性对每一个码都是完全函数依赖;
- (2)所有非主属性对每一个不包含它的码, 也是完全函数依赖;
- (3)没有任何属性完全函数依赖于非码的任何一组属性。

例如, 设 $R(Pno, Pname, Mname)$ 的属性分别表示零件号、零件名和厂商名, 如果约定, 每种零件号只有一个零件名, 但不同的零件号可以有相同的零件名; 每种零件可以有多个厂商生产, 但每家厂商生产的零件应有不同的零件名。这样我们可以得到如下一组函数依赖:

$Pno \rightarrow Pname, (Pname, Mname) \rightarrow Pno$

由于该关系模式 R 中的候选码为 $(Pname, Mname)$ 或 $(Pno, Mname)$, 因而关系模式 R 的属性都是主属性, 不存在非主属性对码的传递依赖, 所以 R 是3NF的。但是, 主属性 $Pname$ 传递依赖于码 $(Pname, Mname)$, 因此 R 不是BCNF的。

当一种零件由多个生产厂家生产时, 零件名与零件号间的联系将多次重复, 带来冗余和操作异常现象。若将 R 分解成:

$R1(Pno, Pname)$ 和 $R2(Pno, Mname)$

就可以解决上述问题, 并且分解后的关系模式 $R1$ 、 $R2$ 都属于BCNF。

规范化理论 (★★★)

5) 4NF

关系模式 $R \in 1NF$, 若对于 R 的每个非平凡多值依赖 $X \twoheadrightarrow Y$ 且 $Y \not\subseteq X$ 时, X 必含有码, 则关系模式 $R(U, F) \in 4NF$ 。

4NF 是限制关系模式的属性间不允许有非平凡且非函数依赖的多值依赖。

注意: 如果只考虑函数依赖, 关系模式最高的规范化程度是 BCNF;
如果考虑多值依赖, 关系模式最高的规范化程度是 4NF。

规范化理论-4NF (★★★)

◆4NF

[例]教师授课关系：学校中某一门课程由多个教师讲授，他们使用相同的一套参考书。每个教员可以讲授多门课程，每种参考书可以供多门课程使用。

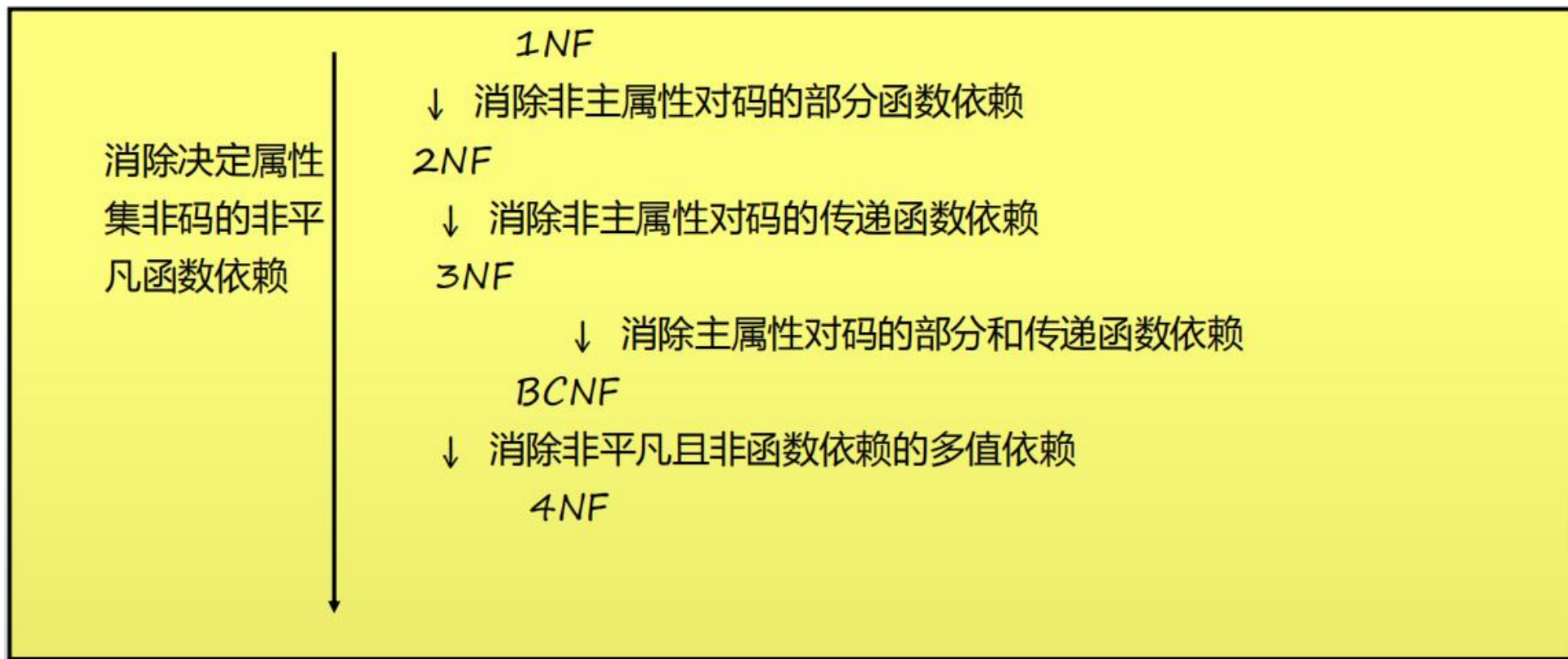
课 程 C	教 员 T	参 考 书 B
物理	{ 李 勇 王 军 }	{ 普通物理学 光学原理 物理习题集 }
数学	{ 李 勇 张 平 }	{ 数学分析 微分方程 高等代数 }
计算数学	{ 张 平 周 峰 }	{ 数学分析 ... }
⋮	⋮	⋮

课程C	教员T	参考书B
物 理	李 勇	普通物理学
物 理	李 勇	光学原理
物 理	李 勇	物理习题集
物 理	王 军	普通物理学
物 理	王 军	光学原理
物 理	王 军	物理习题集
数 学	李 勇	数学分析
数 学	李 勇	微分方程
数 学	李 勇	高等代数
数 学	张 平	数学分析
数 学	张 平	微分方程
数 学	张 平	高等代数
...

例将Teaching (C, T, B) 分解到4NF。

分析：该模式有MVD: $C \twoheadrightarrow T$, $C \twoheadrightarrow B$, 而键为{C,T,B}。则不满足4NF。
利用上述分解方法的: {C,T}, {C, B}

规范化理论 (★★★)



这几种范式之间 $1NF \supseteq 2NF \supseteq 3NF \supseteq BCNF \supseteq 4NF \supseteq 5NF$ 成立。

典型真题

例：考虑含有属性A,B,C,D,E,F的关系，设关系有FD: $AB \rightarrow C$, $BC \rightarrow AD$, $D \rightarrow E$, $CF \rightarrow B$ ，则{A,B}的闭包{A,B}⁺是什么？

$\{A,B\}^+ = \{A,B,C,D,E\}$

给定关系模式R (U,F) , $U=\{A_1,A_2,A_3,A_4\}$, $F=\{A_1 \rightarrow A_2A_3, A_2A_3 \rightarrow A_4\}$, 那么在关系R中 (43) 。以下说法错误的是 (44) 。

- A.有1个候选关键字A1.
- B.有1个候选关键字A2A3
- C.有2个候选关键字A2和A3
- D.有2个候选关键字A1和A2A3

- A.根据F中 " $A_1 \rightarrow A_2A_3$ " , 可以得出 " $A_1 \rightarrow A_2$ "
- B.根据F中 " $A_1 \rightarrow A_2A_3$ " , 可以得出 " $A_1 \rightarrow A_3$ "
- C.根据F中 " $A_2A_3 \rightarrow A_4$ " , 可以得出 " $A_2 \rightarrow A_4, A_3 \rightarrow A_4$ " .
- D.根据F中 " $A_1 \rightarrow A_2A_3, A_2A_3 \rightarrow A_4$ " , 可以得出 " $A_1 \rightarrow A_4$ "

答案：A C

典型真题

给定关系模式 $R \langle U, F \rangle$ ，其中，属性集 $U = \{A, B, C, D, E\}$ 。函数依赖集 $F = \{AC \rightarrow B, B \rightarrow DE\}$ 。关系

R () 且分别有 ()。

A. 只有一个候选关键字AC

B. 只有一个候选关键字AB

C. 有两个候选关键字AC、BC

D. 有两个候选关键字AC、AB

A. 一个非主属性和四个主属性

B. 两个非主属性和三个主属性

C. 三个非主属性和两个主属性

D. 四个非主属性和一个主属性

试题分析

由“ $AC \rightarrow B, B \rightarrow DE$ ”，可以判断AC是唯一候选键（候选关键字），通过AC可以获得其他全部属性。

因此AC分别是主属性，B、D、E为非主属性。

参考答案：A、C

典型真题

销售公司数据库中的关系零件为 $P(Pno, Pname, Sname, City, Qty)$ ， Pno 表示零件号， $Pname$ 表示零件名称， $Sname$ 表示供应商， $City$ 表示所在地， Qty 表示库存量。其函数依赖集 $F=\{Pno \rightarrow Pname, (Pno, Sname) \rightarrow Qty, Sname \rightarrow City\}$ 。关系 P 为（ ），存在冗余度大、修改操作不一致、插入异常和删除异常的问题。若将 P 分解为（ ），则可以解决这一问题。

A. 1NF B. 2NF C. 3NF D. 4NF

A. $P1(Pname, Qty)$ 、 $P2(Pno, Sname, City)$

B. $P1(Pname, Pname)$ 、 $P2(Sname, City, Qty)$

C. $P1(Pno, Pname)$ 、 $P2(Pno, Sname, Qty)$ 、 $P3(Sname, City)$

D. $P1(Pno, Pname)$ 、 $P2(Pno, Qty)$ 、 $P3(Sname, City)$ 、 $P4(City, Qty)$

典型真题

试题分析

原零件关系P存在非主属性对码的部分函数依赖： $(Pno, Sname) \rightarrow Qty$, $Sname \rightarrow City$, 因此P属于1NF。分解后的关系模式P1P2和P3消除了非主属性对码的部分函数依赖，同时不存在传递依赖，故达到3NF。

参考答案：A、C

无损和保持函数依赖 (★★)

★保持函数依赖分解

设数据库模式 $p=\{R_1, R_2, \dots, R_k\}$ 是关系模式 R 的一个分解, F 是 R 上的函数依赖集, p 中每个模式 R_i 上的 FD 集是 F_i 。如果 $\{F_1, F_2, \dots, F_k\}$ 与 F 是等价的(即相互逻辑蕴涵), 那么称分解 p 保持 FD。

★保持函数依赖分解

例1: 有关系模式 $R(A, B, C)$, $F=\{A \rightarrow B, B \rightarrow C\}$, 将其拆分为: $R_1(A, B), R_2(B, C)$, 是否保持函数依赖。

例2: 有关系模式 $R(A, B, C)$, $F=\{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$, 将其拆分为: $R_1(A, B), R_2(B, C)$, 是否保持函数依赖。

例3: 有关系模式 $R(A, B, C)$, $F=\{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$, 将其拆分为: $R_1(A, B), R_2(A, C)$, 是否保持函数依赖。

例4: 有关系模式 $R(A, B, C, D, E)$, $F=\{A \rightarrow B, D \rightarrow E\}$, 将其拆分为: $R_1(A, B, C), R_2(D, E)$, 是否保持函数依赖。

无损和保持函数依赖 (★★)

★无损分解

什么是有损，什么又是无损？有损：不能还原。 无损：可以还原。

无损连接分解：指将一个关系模式分解成若干个关系模式后，通过自然联接和投影等运算仍能还原到原来的关系模式。

第一种方法是追踪表格法，适用于任何关系。

第二种方式是公式法，适用于两个模式得关系。

公式：如果R的分解为 $p=\{R_1, R_2\}$, F为R所满足的函数依赖集合，分解p具有无损连接性的充分必要条件是：

$R_1 \cap R_2 \rightarrow (R_1 - R_2)$

或 $R_1 \cap R_2 \rightarrow (R_2 - R_1)$

其中， $R_1 \cap R_2$ 表示模式的交，为R1与R2中公共属性组成， $R_1 - R_2$ 或 $R_2 - R_1$ 表示模式的差集， $R_1 - R_2$ 表示R1中去除R1和R2的公共属性所组成。当模式R分解成两个关系模式R1和R2时，如果R1与R2的公共属性能函数决定R1中或R2中的其它属性，这样的分解就具有无损连接性。

典型真题

设关系模式 $R(U, F)$ ，其中： $U = \{A, B, C, D, E\}$ ， $F = \{A \rightarrow B, DE \rightarrow B, CB \rightarrow E, E \rightarrow A, B \rightarrow D\}$ 。（ ）为关系模式 R 的候选关键字。分解（ ）是无损连接，并保持函数依赖的。

A. AB B. DE C. DB D. CE

A. $\rho = \{R_1(AC), R_2(ED), R_3(B)\}$

B. $\rho = \{R_1(AC), R_2(E), R_3(DB)\}$

C. $\rho = \{R_1(AC), R_2(ED), R_3(AB)\}$

D. $\rho = \{R_1(ABC), R_2(ED), R_3(ACE)\}$

典型真题

试题分析

第54题选项D中， $E \rightarrow A$ ， $A \rightarrow B$ ， $B \rightarrow D$ ， $CB \rightarrow E$ 包括了全部的属性。

第55题判断无损连接有多种方法，下面介绍一种构造二维表进行判断的方法（以正确选项D为例）：

① 构造一个初始的二维表，若“属性”属于“模式”中的属性，则填 a_j ，否则填 b_{ij} 。

模式 \ 属性	A	B	C	D	E
ABC	a1	a2	a3	b14	b15
ED	b21	b22	b23	a4	a5
ACE	a1	b32	a3	b34	a5

典型真题

② 逐一考察关系式中的函数依赖，将表中的 b_{ij} 修改成 a_j 。如 $CB \rightarrow E$ ，则上表中第一行第六列的 b_{15} ，修改成 a_5 。修改结果如下表所示。

模式 \ 属性	A	B	C	D	E
ABC	a1	a2	a3	a4	a5
ED	a1	a2	b23	a4	a5
ACE	a1	a2	a3	b34	a5

③ 修改后的表格若任一行存在 a_1, a_2, a_3, a_4, a_5 的情况，该分解就属于无损连接，如上表格中第一行存在 a_1, a_2, a_3, a_4, a_5 ，因此属于无损连接。

参考答案：D、D

典型真题

给定关系模式 $R \langle U, F \rangle$, $U = \{A, B, C, D, E\}$, $F = \{B \rightarrow A, D \rightarrow A, A \rightarrow E, AC \rightarrow B\}$, 则 R 的候选关键字为 (), 分解 $\rho = \{R_1(ABCE), R_2(CD)\}$ ()。

A. CD B. ABD C. ACD D. ADE

- A. 具有无损连接性, 且保持函数依赖
- B. 不具有无损连接性, 但保持函数依赖
- C. 具有无损连接性, 但不保持函数依赖
- D. 不具有无损连接性, 也不保持函数依赖

典型真题

试题分析

CD能推出题中关系式的所有属性，因此 R 的候选关键字为CD。

判断是否为无损连接，首先进行 $R1 \cap R2 = C$ ，由于C不能推出R1或者R2中的任何属性值，因此该分解为有损分解。

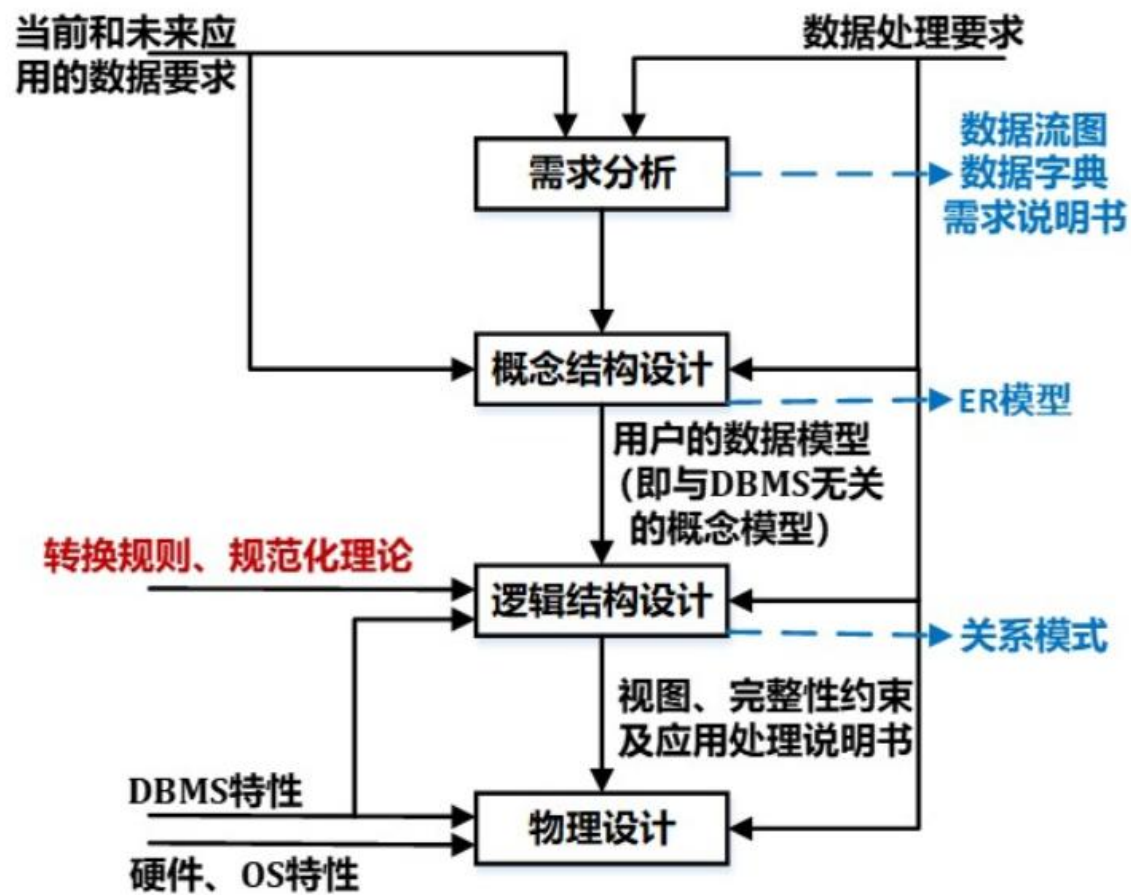
原关系式F中有 $D \rightarrow A$ 而分解的R1（ABCE）中没有D，所以该分解不保持函数依赖。

参考答案：A 、 D

目录

- 1 数据库基本概念
- 2 关系型数据库概念
- 3 关系代数
- 4 SQL语句
- 5 规范化理论
- 6 数据库设计过程
- 7 并发控制
- 8 分布式数据库
- 9 数据库安全、数据仓库与商业智能

数据库设计过程 (★★★)



典型真题

在数据库的（ ）阶段进行关系规范化。

A.需求分析 B.概念设计 C.逻辑设计 D.物理设计

答案：C

在数据库设计的需求分析、概念结构设计、逻辑结构设计和物理结构设计的四个阶段中，基本E-R图是（ ）。

- A.需求分析阶段形成的文档，并作为概念结构设计阶段的设计依据
- B.逻辑结构设计阶段形成的文档，并作为概念结构设计阶段的设计依据
- C.概念结构设计阶段形成的文档，并作为逻辑结构设计阶段的设计依据
- D.概念结构设计阶段形成的文档，并作为物理设计阶段的设计依据

答案：C

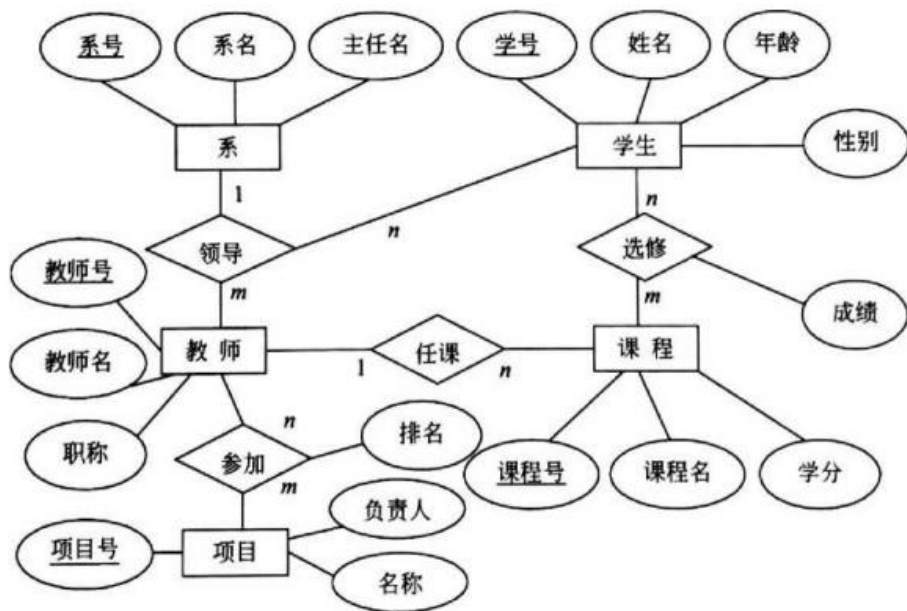
数据库设计过程 (★★★)

阶段	概念	可交付成果	
需求分析	目标是通过调查研究，了解用户的数据和处理要求，并按照一定格式整理成需求规格说明书。	数据流图及数据字典的内容：数据项、数据流、数据存储、数据加工（处理过程）从而确定系统边界。	
概念结构设计	在需求分析阶段产生的需求说明书的基础上，按照特定的方法将它们抽象为一个不依赖于任何DBMS的数据模型，即概念模型	1、抽象数据★ 2、设计局部E-R图（局部视图---识别实体及其标识---确定实体间的联系---分配实体及联系的属性） 3、合并取消冲突，设计初步E-R模式 4、修改重构消除冗余生成基本E-R图	E-R模型（实体关系模型）
逻辑结构设计	将概念模型转化为某个特定的DBMS上的逻辑模型（层次模型、网状模型、关系模型）	1、转换为数据模型 2、关系规范化 3、模式优化 4、设计用户子模式★	关系模式
物理设计	①设计存储记录结构，包括记录的组成、数据项的类型和长度，以及逻辑记录到 存储记录的映射。 ②确定数据存储安排。 ③设计访问方法，为存储在物理设备上的数据提供存储和检索的能力。 ④进行完整性和安全性的分析与设计。 ⑤数据库程序设计。		

数据库设计过程-概念结构设计 (★★★)

◆概念结构设计：其任务是在需求分析阶段产生的需求说明书的基础上，按照特定的方法将它们抽象为一个不依赖于任何DBMS的数据模型，即概念模型。

下图为学校教学管理系统E-R图。



实体

属性

联系：1: 1、1: n或m: n。

数据库设计过程-概念结构设计 (★★★)

◆集成的方法:

- 多个局部E-R图一次性集成
- 逐步集成, 用累加的方式一次性集成两个局部的E-R图。

冲突类型:

1、属性冲突

- 属性域冲突: 不同学校编码方式不同
- 属性值冲突: 重量采用千克、磅

2、结构冲突

- 同一对象在不同应用中的抽象不同:
职工在某一应用中是实体, 在另一应用中是属性。
- 同一实体在不同ER图中属性个数和排列次序不同

3、命名冲突

- 同名异义
- 异名同义

典型真题

某企业开发信息管理系统平台进行 E-R 图设计，人力部门定义的是员工实体具有属性：员工号、姓名、性别、出生日期、联系方式和部门，培训部门定义的培训师实体具有属性：培训师号，姓名和职称，其中职称={初级培训师，中级培训师，高级培训师}，这种情况属于()，在合并E-R图时，解决这一冲突的方法是()。

A.属性冲突 B.结构冲突 C.命名冲突 D.实体冲突

- A.员工实体和培训师实体均保持不变
- B.保留员工实体、删除培训师实体
- C.员工实体中加入职称属性，删除培训师实体
- D.将培训师实体所有属性并入员工实体，删除培训师实体

答案：B C

解析：结合第二问，可知道第二问C的做法是最合适的，反推可知这个问题是结构冲突。

数据库设计过程-逻辑结构设计 (★★★)

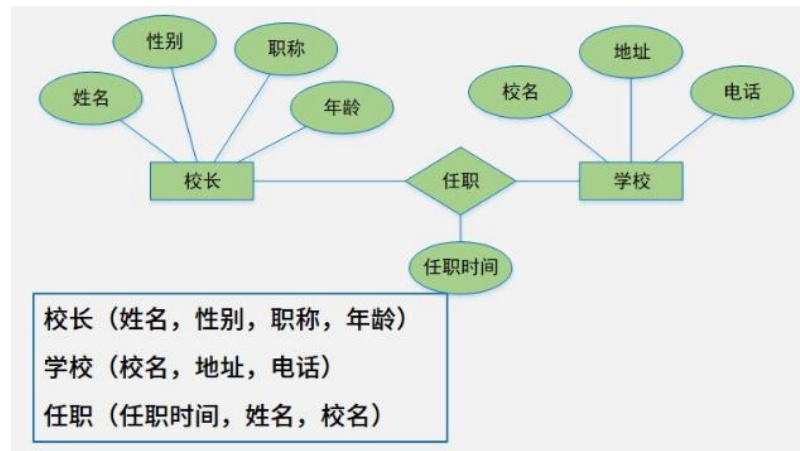
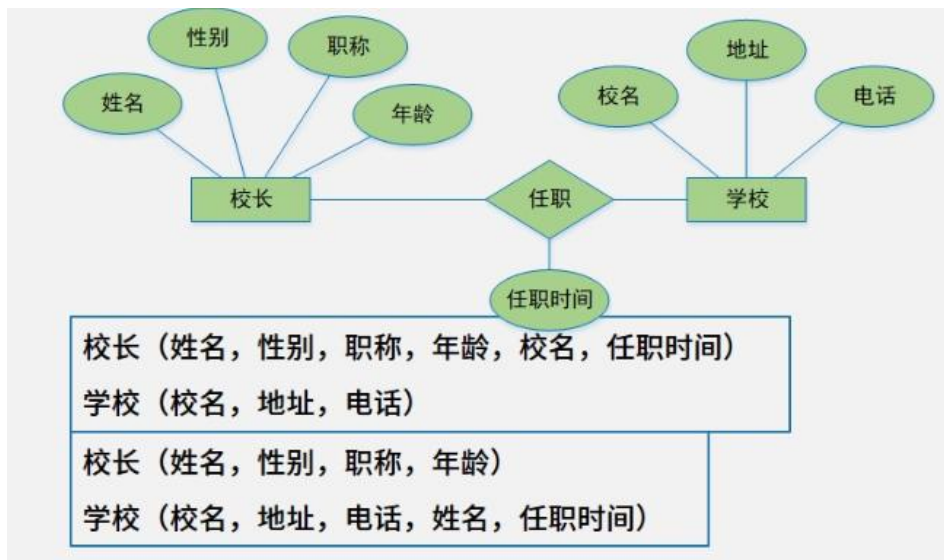
◆逻辑结构设计-E-R图转换为关系模式

一个实体型必须转换为一个关系模式。

联系转关系模式有以下几种方式：

一、联系的属性，1:1的联系：

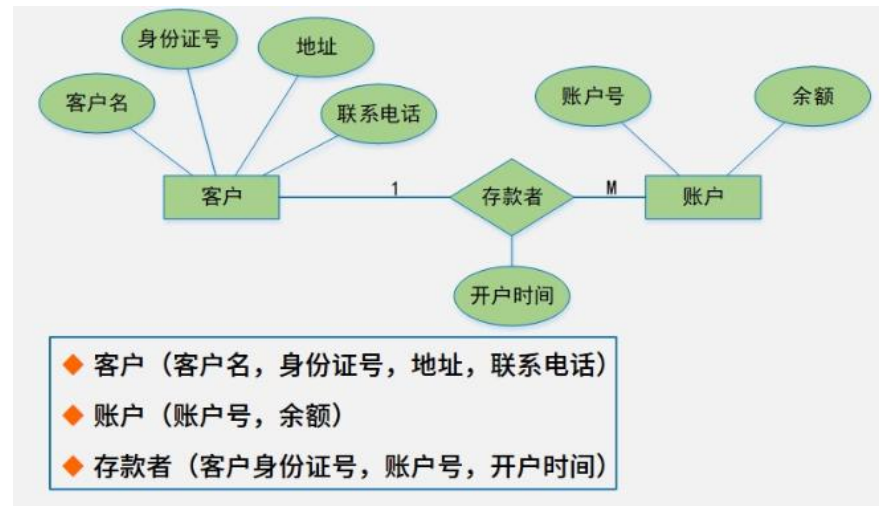
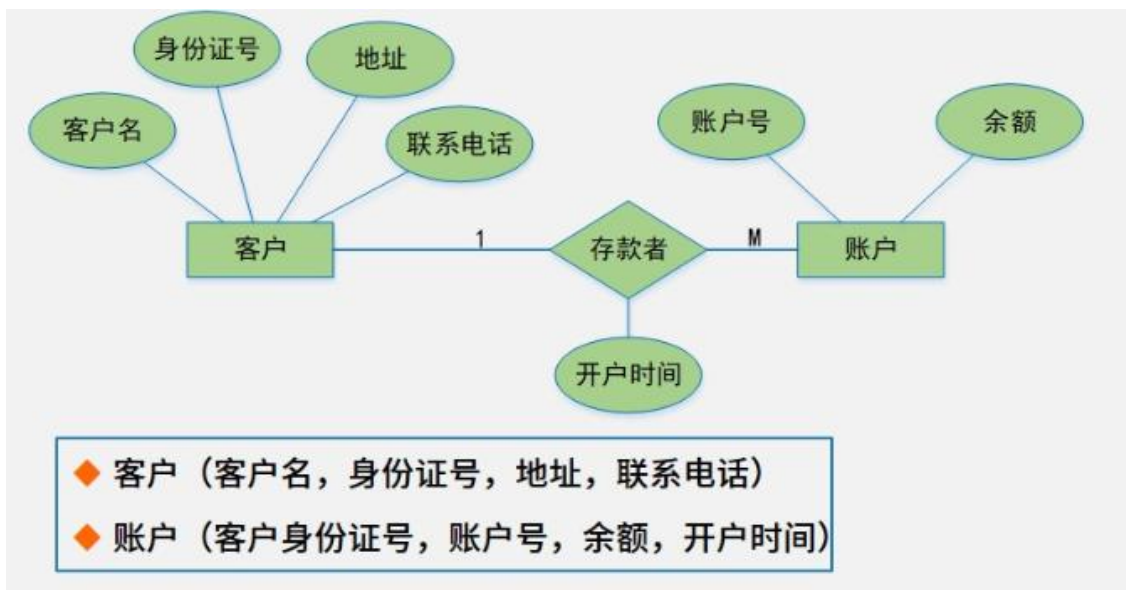
- 1、一个1:1联系可以转换为一个独立的关系模式。
- 2、或者也可以与任意一端对应的关系模式合并。



数据库设计过程 (★★★)

二、联系的属性，1:n的联系：

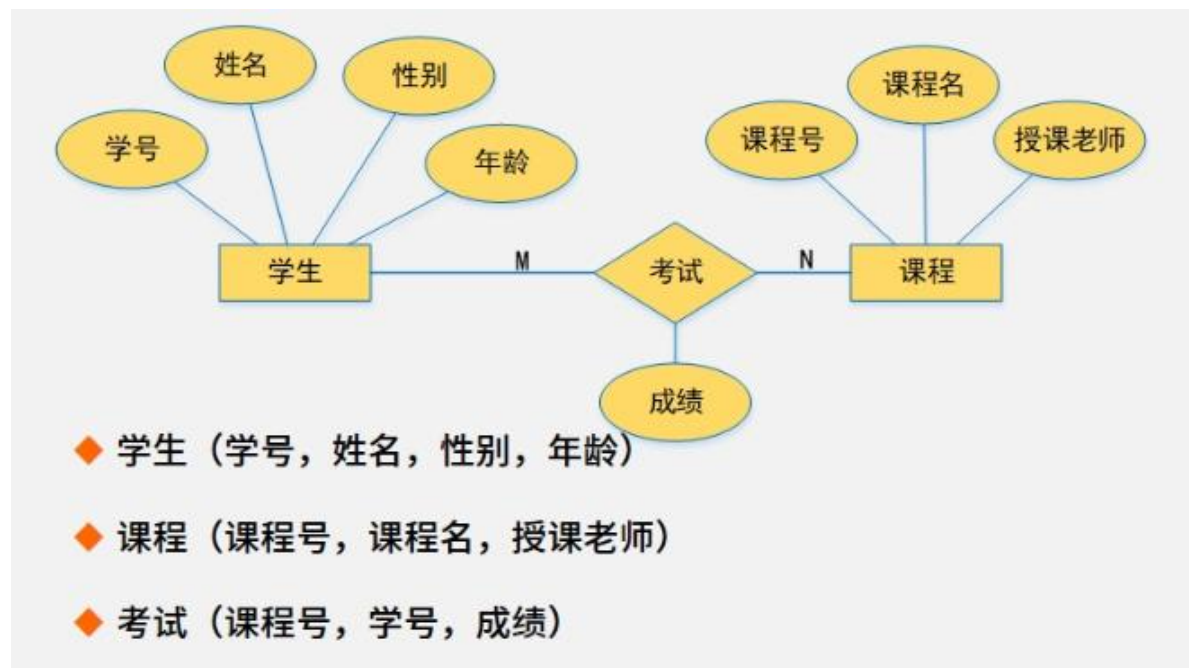
- 1、一个1:n的联系可以转换为一个独立的关系模式。
- 2、与n端对应的关系模式合并。



数据库设计过程 (★★★)

三、联系的属性，n:m的联系：

一个n:m的联系可以转换为一个独立的关系模式。

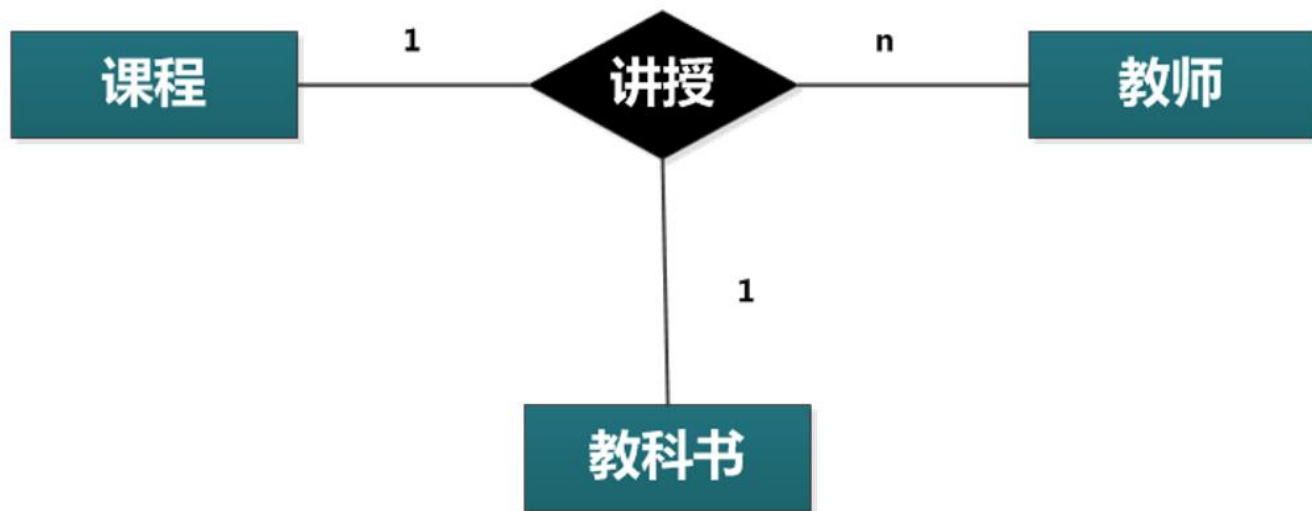


数据库设计过程 (★★★)

四、联系的属性，多元的联系：

一个多元联系可以转换为一个独立的关系模式。

例如：讲授（职工号，课程号，书号...），其中课程号、职工号、书号为关系模式的组合（联合）主键。



数据库设计过程

1、用户视图的确定

- 根据数据流图及用户信息建立视图模式，提高数据的安全性和独立性。
- 根据数据流图确定处理过程使用的视图。
- 根据用户类别确定不同用户使用的视图。

2、反规范化 (★★★)

反规范化是加速读操作性能(数据检索)的方法，通过有选择地在数据结构标准化后添加特定的冗余数据来增加查询效率。

常见的反规范化操作有：冗余列、派生列、表重组和表分割，其中表分割又分为水平分割和垂直分割。

反规范化会带来冗余数据不一致问题，常采用数据同步的方法来解决。

方法有：应用程序同步、批量处理同步和触发器同步等。

技术手段	说明
增加派生性冗余列	已有单价和数量列，增加“总价”列
增加冗余列	已有学号列，增加“姓名”列
重新组表	把拆分的表重新组表
分割表	把用户表做水平分割，长沙的用户存在长沙，上海的用户存在上海

数据库设计过程 (★)

1、物理结构设计

物理结构设计就是为逻辑数据模型选取一个最适合应用环境的物理结构（包括存储结构和存取方法）。

物理结构设计的主要工作步骤：确定数据分布、存储结构、访问方式

2、数据库实施

运用DBMS提供的数据库语言(如SQL)及宿主语言，根据逻辑设计和物理设计的结果，在计算机上建立起实际的数据库结构，数据加载(装入)，进行试运行和评价的过程，叫数据库的实施(实现)。包括以下三个步骤：建立数据库结构、数据加载(组织数据入库)、试运行和评价。

3、数据库运行维护

数据库维护工作的主要内容：

- 数据库性能的监测和改善
- 数据库备份及故障恢复
- 数据库重组和重构

数据库备份与故障恢复 (★★)

数据库备份及故障恢复

数据备份是指为防止系统出现操作失误或系统故障导致数据丢失，而将全部或部分数据集合从应用主机的硬盘或阵列复制到其它的存储介质的过程。

备份可分为：

- ✓ 冷备份（静态备份）是在数据库正常关闭的情况下进行的，所以备份过程中不允许对数据库进行任何存取、修改活动。
- ✓ 热备份（动态备份）热备份是指在数据库正常运行的状态下，将数据库中的数据文件备份出来。

按备份的内容分类

- ✓ 完全备份：备份全部文件，并不依赖文件的存档属性来确定备份那些文件。
- ✓ 差分备份：备份自上一次完全备份以来变化过的文件。
- ✓ 增量备份：备份上一次备份后（无论是哪种备份），所有发生变化的文件

数据库备份与故障恢复 (★)

故障关系	故障原因	解决办法
事务本身的可预期故障	本身逻辑	在程序中预先设置Rollback语句
事务本身的不可预期故障	算术溢出、违反存储保护	由DBMS的恢复子系统通过日志，撤销事务对数据库的修改，回退到事务初始状态
系统故障	系统停止运转	通过使用检查点法，系统重启时自动完成。
介质故障	外存被破坏	一般使用日志重新做业务。
撤销事务（UNDO）：故障发生时未完成的事务，放入Undo撤销。 重做事务（REDO）：故障发生前已提交的事务，放入Redo重做。		

典型真题

例.假设系统中有正在运行的事务，若要转储全部数据库，则应采用()方式。

- A.静态全局转储
- B.动态增量转储
- C.静态增量转储
- D.动态全局转储

答案：D

数据库设计过程

一、应用系统中，用户不能直接访问后台的数据库，需要高级程序语言来完成与用户之间的交互，因此数据库管理系统需要提供程序级别的接口来访问数据。

常见的应用程序与数据库的交互方式：

- ✓ 库函数，如Oracle的OCI(Oracle调用接口)。
- ✓ 嵌入式 SQL，将SQL语句直接写入某种高级程序语言。
- ✓ 通用数据接口标准，如ODBC、JDBC、DAO、RDO、ADO等。
- ✓ 对象关系映射 (ORM)，是一种为了解决面向对象与关系数据库不匹配的技术。就是将Java中的对象与数据库中的表关联对应起来。

二、框架

(1) Hibernate：全自动的框架，强大、复杂、笨重、学习成本较高。

(2) Mybatis：半自动的框架。

(3) JPA (Java Persistence API): JPA JDK 5.0 注解或XML描述对象- 关系表的映射关系，是Java 自带的框架。

目录

- 1 数据库基本概念
- 2 关系型数据库概念
- 3 关系代数
- 4 SQL语句
- 5 规范化理论
- 6 数据库设计过程
- 7 并发控制
- 8 分布式数据库
- 9 数据库安全、数据仓库与商业智能

并发控制-事务属性

◆数据库系统运行的基本工作单位是事务，事务相当于操作系统中的进程，是用户定义的一个数据库操作序列，这些操作序列要么全做要么全不做，是一个不可分割的工作单位。（★★）

01

原子性 (Atomicity)

操作

操作序列要么全做要么全不做。

02

一致性 (Consistency)

数据库从一个一致性状态变到另一个一致性状态。

03

隔离性 (Isolation)

执行

不能被其他事务干扰。

04

持续性 (永久性) (Durability)

变化

一旦提交，改变就是永久性的。

典型真题

事务的()是指“当多个事务并发执行时，任一事务的更新操作直到其成功提交的整个过程，对其他事务都是不可见的”。

- A.原子性
- B.一致性
- C.隔离性
- D.持久性

参考答案：C

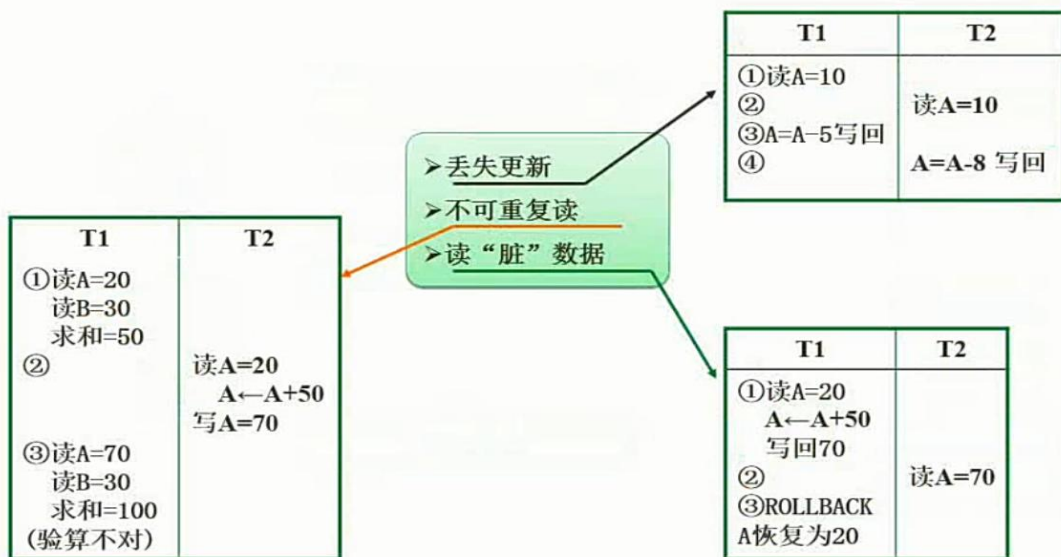
并发控制-事务管理

事务通常以 BEGIN TRANSACTION（事务开始）语句开始，以 COMMIT 或 ROLLBACK 语句结束。COMMIT 称为“事务提交语句”，表示事务执行成功的结束。ROLLBACK 称为“事务回退语句”，表示事务执行不成功的结束。（★★★）

◆丢失更新：事务1对数据A进行了修改并写回，事务2也对A进行了修改并写回，此时事务2写回的数据会覆盖事务1写回的数据，就丢失了事务1对A的更新。即对数据A的更新会被覆盖。

◆不可重复读：事务2读A,而后事务1对数据A进行了修改并写回，此时若事务2再读A,发现数据不对。即一个事务重复读A两次，会发现数据A有误。

◆读脏数据：事务1对数据A进行了修改后，事务2读数据A,而后事务1回滚，数据A恢复了原来的值，那么事务2对数据A做的事是无效的，读到了脏数据。



并发控制-事务管理-封锁技术

◆处理并发控制的主要方法是采用封锁技术。（★★★）

它有两种类型：排他型封锁（X 封锁）和共享型封锁（S 封锁），分别介绍如下：

（1）排他型封锁（简称 X 封锁）。如果事务 T 对数据 A（可以是数据项、记录、数据集，乃至整个数据库）实现了 X 封锁，那么只允许事务 T 读取和修改数据 A，其他事务要等事务 T 解除 X 封锁以后，才能对数据 A 实现任何类型的封锁。可见 X 封锁只允许一个事务独锁某个数据，具有排他性。

（2）共享型封锁（简称 S 封锁）。X 封锁只允许一个事务独锁和使用数据，要求太严。需要适当放宽，例如可以允许并发读，但不允许修改，这就产生了 S 封锁概念。S 封锁的含义是：如果事务 T 对数据 A 实现了 S 封锁，那么允许事务 T 读取数据 A，但不能修改数据 A，在所有 S 封锁解除之前绝不允许任何事务对数据 A 实现 X 封锁。

独占X，共享S

并发控制-事务管理-封锁技术

防止丢失更新加锁

T1	T2
①对A加写锁 ② ③读A=10 ④A=A-5写回 ⑤释放对A的写锁 ⑥ ⑦ ⑧	对A加写锁 等待 等待 等待 读A=5 A=A-8写回 释放对A的写锁

防止读”脏“数据

T1	T2
①对A加写锁 ②读A=20 ③A=A+50 ④写回70 ⑤ ⑥ROLLBACK ⑦A恢复为20, 释放写锁 ⑧ ⑨	对A加读锁 等待 等待 读A=20 释放对A的读锁

并发控制-事务管理-封锁技术

防止不可重复读加锁

T1	T2
<p>①对A与B加S锁(读锁) 读A=20 读B=30 求和=50 ② ③读A=20读B=30求和=50 释放对A和B的读锁</p>	<p>对A加X锁(写锁) 注：由于A已加了读锁，所以 等待等待 等待 等待 等待 读A=20 A=A+50 写A=70 释放对A的写锁</p>

并发控制-事务管理-封锁协议

级别	内容	内容		优点	缺点
一级封锁协议	事务在修改数据之前，必须对该数据加X锁，直到事务结束才释放	只读数据可以不加锁		防止“丢失修改”	不加锁的事务，可能读脏或可能“不可重复读”
二级封锁协议		对其它事务在读数据前必须先加S锁	读完后即可释放S锁	防止“丢失修改” 防止“读脏数据”	对加S锁的事务，可能“不可重复读”
三级封锁协议			直到事务结束时才释放S锁	防止“丢失修改” 防止“读脏数据” 防止“不可重复读”	

并发控制-事务管理-封锁协议

◆两段锁封锁协议：运用封锁方法时，对数据对象加锁时需要约定一些规则，如何时申请加锁？持锁时间？何时释放封锁理论上证明使用两段锁协议产生的是可串行化调度，事务必须分两个阶段对数据进行加锁和解锁。

但是不能保证产生死锁。

- I. 第一阶段是：获得封锁，也称扩展阶段：事务可以申请获得任何数据项上的任何类型的锁，但是不能释放任何锁
- II. 第二阶段：释放封锁，也称收缩阶段：事务可以释放任何数据项上的任何类型的锁，但是不能再申请任何锁

封锁对象的大小-----封锁粒度

两段锁协议可串行化，可能发生死锁。

【例】事务T_i遵守两段锁协议，其封锁序列是：

Slock A Slock B Xlock C Unlock B Unlock A Unlock C ;

|← 扩展阶段 →| |← 收缩阶段 →|

事务T_j不遵守两段锁协议，其封锁序列是：

Slock A Unlock A Slock B Xlock C Unlock C Unlock B ;

典型真题

若事务T1对数据D1加了共享锁，事务T2、T3分别对数据D2、D3加了排它锁，则事务T1对数据（ ）；事务T2对数据（ ）。

A.D2、D3加排它锁都成功

B.D2、D3加共享锁都成功

C.D2加共享锁成功，D3加排它锁失败

D.D2、D3加排它锁和共享锁都失败。

A.D1、D3加共享锁都失败

B.D1、D3加共享锁都成功

C.D1加共享锁成功，D3如排它锁失败。

D.D1加排它锁成功，D3加共享锁失败

典型真题

试题分析：

共享锁（S锁）：又称读锁，若事物T对数据对象A加上S锁，其他事物只能再对A加S锁，而不能加X锁，而不能加X锁，直到T释放A上的S锁

排他锁（X锁）：又称写锁。若事物T对数据对象A加上X锁，其他事物不能再对A加任何锁，直到T释放A上的锁。

参考答案：D C

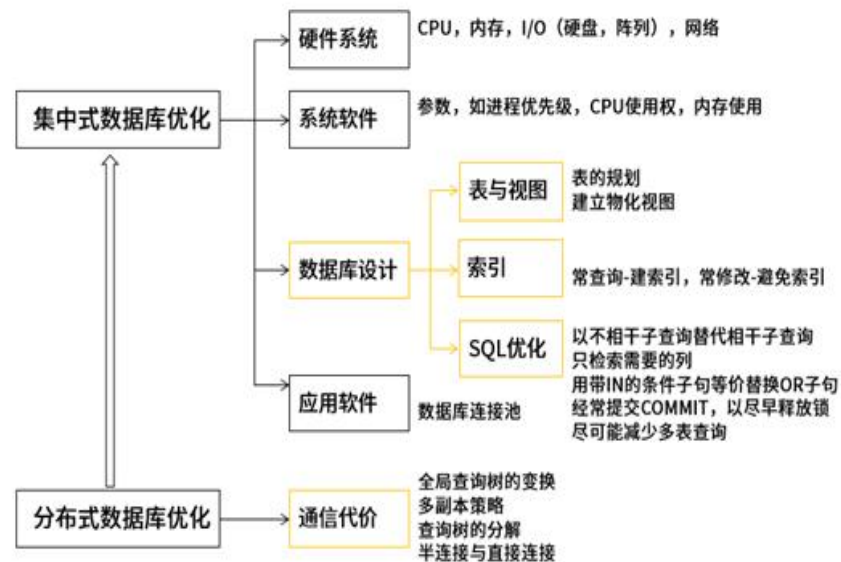
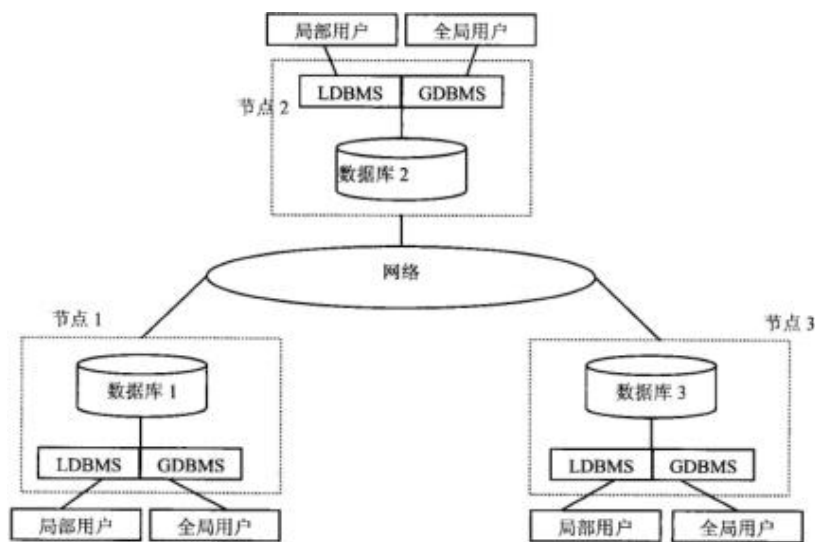
目录

- 1 数据库基本概念
- 2 关系型数据库概念
- 3 关系代数
- 4 SQL语句
- 5 规范化理论
- 6 数据库设计过程
- 7 并发控制
- 8 分布式数据库
- 9 数据库安全、数据仓库与商业智能

分布式数据库

◆分布式数据库系统：物理上分散逻辑相关的数据库系统，系统的数据分布存放在计算机网络的
不同场地的计算机，每一个场地都有自治处理（独立处理）能力并可完成局部应用，而每一个场
地也参与全局应用，程序通过网络通信子系统执行全局应用。（★★）

- 物理分布性
- 逻辑相关性
- 场地自治性
- 场地透明性



分布式数据库管理系统

◆分布式数据库管理系统组成：

LDBMS（局部数据库管理系统）、GDBMS（全局数据库管理系统）、全局数据字典、通信管理（CM）

◆分布式数据库管理系统-结构

- 全局控制集中的DDBMS
- 全局控制分散的DDBMS
- 全局控制部分分散的DDBMS

典型真题

某银行信息管理系统采用分布式数据库系统，以便对本地储户的存储业务能够在本地正常进行而不依赖于其他场地的数据库，这种情况称为分布式数据库的()。

A.分布性 B.共享性 C.可用性 D.自治性.

参考答案：D

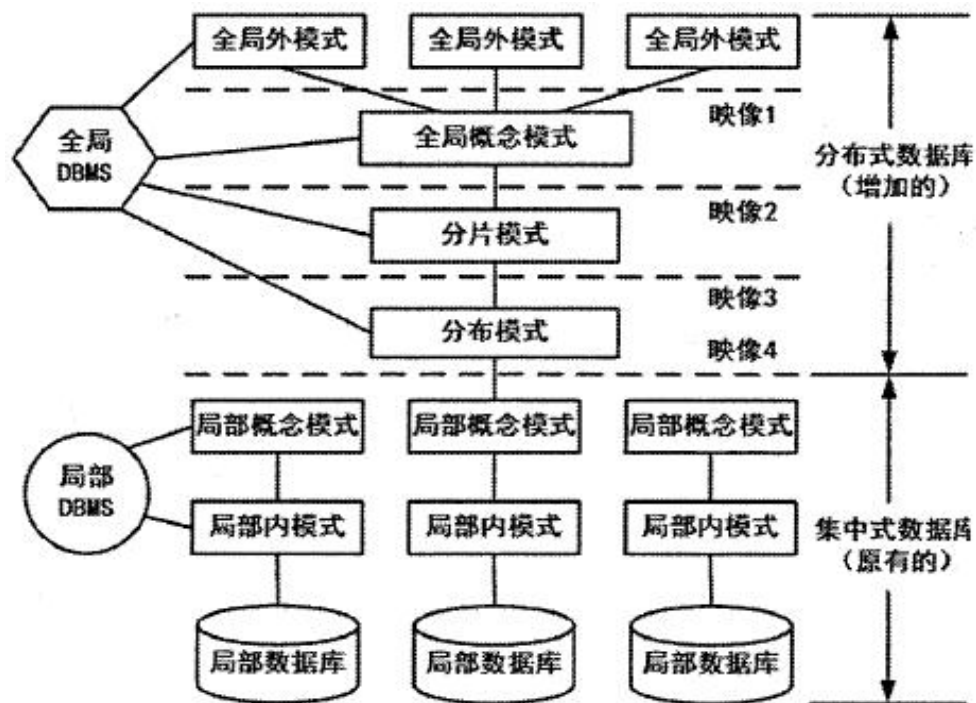
某高校信息系统采用分布式数据库系统，该系统中“当某一场地故障时，系统可以使用其它场地上的副本而不至于使整个系统瘫痪”和“数据在不同场地上的存储”分别称为分布式数据库的()。

A.共享性和分布性 B.自治性和分布性 C.自治性和可用性 D.可用性和分布性.

参考答案：D

分布式数据库管理系统

◆分布式数据库系统的模式结构有六个层次，可以概括和说明任何分布式数据库系统的概念和结构。（★★★）



数据分片 (★★★)

数据存放的单位不是关系而是分段，一个片断是全局概念模式的某个逻辑部分

4种形式：

- 水平分片：把全局关系的所有元组划分为不想交的子集
- 垂直分片：把全局关系的所有属性划分为若干子集，每个属性至少映射到一个垂直分片中，并且包含主键
- 导出分片：水平分片的条件不是本关系的属性条件，而是其他关系的属性条件
- 混合分片：以上三者的混合

3条原则：完备性、可重构、不相交

4种数据分配策略 (★★★)

- 集中式：所有的数据分段都安排在同一个场地上。
- 分割式：所有数据只有一份，它被分割成若干逻辑分段，每个逻辑分段被指派到一个特定的场地上。
- 全复制式：数据在每个场地重复存储，也就是每个场地上都有一个完整的数据副本。
- 混合式：介于分割式和全复制之间的分配方式。

典型真题

分布式数据库系统除了包含集中式数据库系统的模式结构之外，还增加了几个模式级别，其中（ ）定义分布式数据库中数据的整体逻辑结构，使得数据使用方便，如同没有分布一样。

- A. 分片模式 B. 全局外模式 C. 分布模式 D. 全局概念模式.

参考答案：D

解析：本题考查分布式数据库的基本概念。

分布式数据库在各结点上独立，在全局上统一。因此需要定义全局的逻辑结构，称之为全局概念模式，全局外模式是全局概念模式的子集，分片模式和分布模式分别描述数据在逻辑上的分片方式和在物理上各结点的分布形式。

分布式数据库管理系统

◆分布式数据库系统透明性 (★★★)



✪ **分片透明**: 是指用户不必关心数据是如何分片的, 它们对数据的操作在全局关系上进行, 即如何分片对用户是透明的。

✪ **复制透明**: 用户不用关心数据库在网络中各个节点的复制情况, 被复制的数据的更新都由系统自动完成。

✪ **位置透明**: 是指用户不必知道所操作的数据放在何处, 即数据分配到哪个或哪些站点存储对用户是透明的。

✪ **局部映像透明性(逻辑透明)**: 是最低层次的透明性, 该透明性提供数据到局部数据库的映像, 即用户不必关心局部 DBMS 支持哪种数据模型、使用哪种数据操纵语言, 数据模型和操纵语言的转换是由系统完成的。因此, 局部映像透明性对异构型和同构异质的分布式数据库系统是非常重要的。

分布式数据库管理系统

两阶段提交协议 (★★)

2PC协议把数据提交分成两个阶段

- 表决阶段：协调者发布“准备提交”指令，所有参与者进行表决，具有一票否决权.目的是形成一个共同的决定。
- 执行阶段：根据协调者的指令，参与者提交或撤销事务，并给协调者发送确认信息。目的是实现这个协调者的决定。

规则：只要有一个参与者撤销事务，协调者就必须做出全局撤销的决定。

只有所有参与者都同意提交事务，协调者才能做出全局提交的决定。

三阶段提交协议

3PC协议把数据提交分成两个阶段

- 第一阶段：协调者向所有参与者发布“准备提交”指令，所有参与者进行表决，只有所有参与者都投票“建议提交”，才会进入第二阶段。
- 第二阶段：协调者向所有参与者发“全局预提交”，只有所有参与者都提交“准备就绪”才能进入第三阶段
- 第三阶段：协调者向所有的参与者发布“全局提交”报文

目录

- 1 数据库基本概念
- 2 关系型数据库概念
- 3 关系代数
- 4 SQL语句
- 5 规范化理论
- 6 数据库设计过程
- 7 并发控制
- 8 分布式数据库
- 9 数据库安全、数据仓库与商业智能

数据库安全

措施（★）	说明
用户标识和鉴定	最外层的安全保护措施，可以使用用户帐户、口令及随机数检验等方式
存取控制	对用户进行授权，包括操作类型(如查找、插入、删除、修改等动作)和数据对象(主要是数据范围)的权限。(Grant和Revoke)
密码存储和传输	对远程终端信息用密码传输
视图的保护	对视图进行授权
审计	使用一个专用文件或数据库，自动将用户对数据库的所有操作记录下来

数据仓库

◆数据仓库是一个面向主题的、集成的、非易失的、且随时间变化的数据集合，用于支持管理决策。

(★)

- 面向主题：按照一定的主题域进行组织的。
- 集成的：数据仓库中的数据是在对原有分散的数据库数据抽取、清理的基础上经过系统加工、汇总和整理得到的，必须消除源数据中的不一致性，以保证数据仓库内的信息是关于整个企业的一致全局信息。
- 相对稳定的：数据仓库的数据主要供企业决策分析之用，所涉及的数据操作主要是数据查询，一旦某个数据进入数据仓库以后，一般情况下将被长期保留，也就是数据仓库中一般有大量的查询操作，但修改和删除操作很少，通常只需要定期的加载、刷新。
- 反映历史变化：数据仓库中的数据通常包含历史信息，系统记录了企业从过去某一时点(如开始应用数据仓库的时点)到目前的各个阶段的信息，通过这些信息，可以对企业的发展历程和未来趋势做出定量分析和预测。

数据仓库

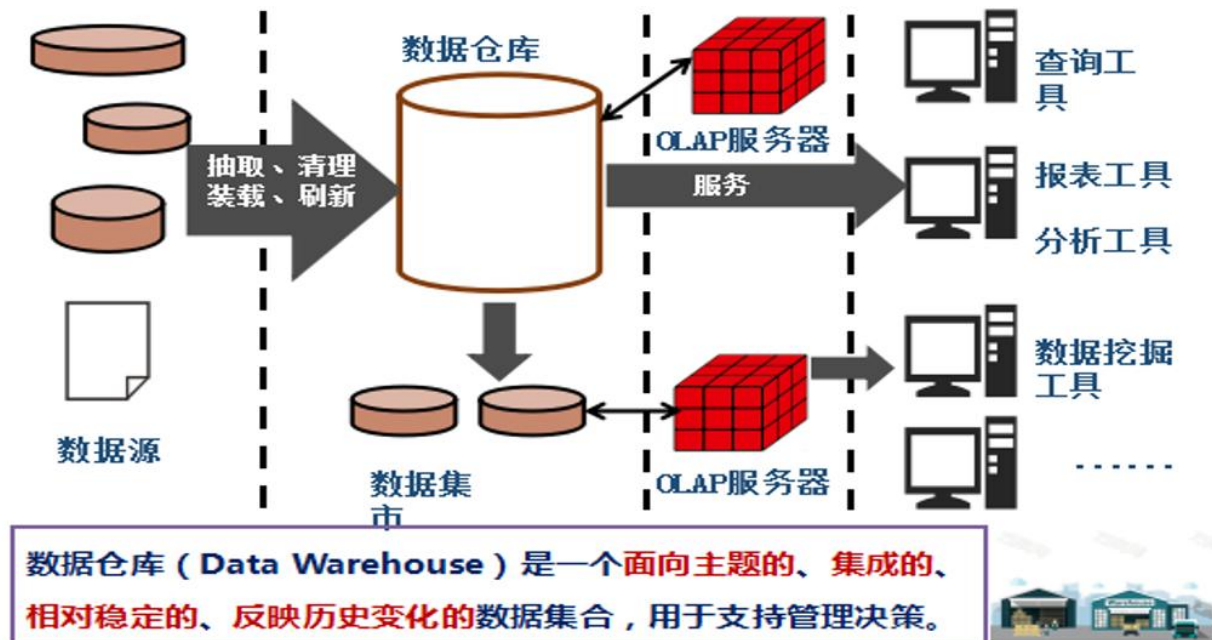
◆数据仓库的结构通常包含四个层次，如下图所示：（★）

1.数据源：是数据仓库系统的基础，是整个系统的数据源泉。

2.数据的存储与管理：是整个数据仓库系统的核心。

3.OLAP(联机分析处理)服务器：对分析需要的数据进行有效集成，按多维模型组织，以便进行多角度、多层次的分析，并发现趋势。

4.前端工具：主要包括各种报表工具、查询工具、数据分析工具、数据挖掘工具以及各种基于数据仓库或数据集市的应用开发工具。



商业智能

◆商业智能（BI系统）主要包括数据预处理、建立数据仓库、数据分析和数据展现四个主要阶段。（★）

- 数据预处理是整合企业原始数据的第一步，它包括数据的抽取(Extraction)、转换(Transformation)和加载(Load)三个过程(ETL过程)；
- 建立数据仓库则是处理海量数据的基础；
- 数据分析是体现系统智能的关键，一般采用联机分析处理(OLAP)和数据挖掘两大技术。联机分析处理不仅进行数据汇总/聚集，同时还提供切片、切块、下钻、上卷和旋转等数据分析功能，用户可以方便地对海量数据进行多维分析。数据挖掘的目标则是挖掘数据背后隐藏的知识，通过关联分析、聚类和分类等方法建立分析模型，预测企业未来发展趋势和将要面临的问题；
- 数据展现则主要保障系统分析结果的可视化。

本章重点回顾

- 1、三级模式两级映像
- 2、关系代数
- 3、规范化理论
- 4、数据库设计过程
- 5、分布式数据库

THANKS