1. **数据库系统给概述**
2. 数据和信息的区别和联系

信息：信息就是消息，是事物属性的标识

数据：用**符号表示**的用于反映客观世界中客体属性的记录

**数据是符号化的信息，信息是语义化的数据**

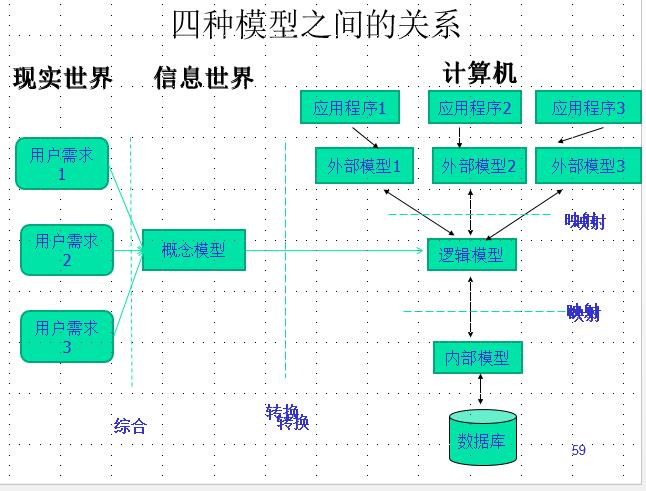
1. 数据管理技术的发展阶段

1）人工管理阶段

2）文件管理阶段

3）数据库管理阶段

1. **简答1**从现实世界、信息世界、数据世界的抽象和转换的建模过程中，涉及四种模型，简述四种模型之间的关系。



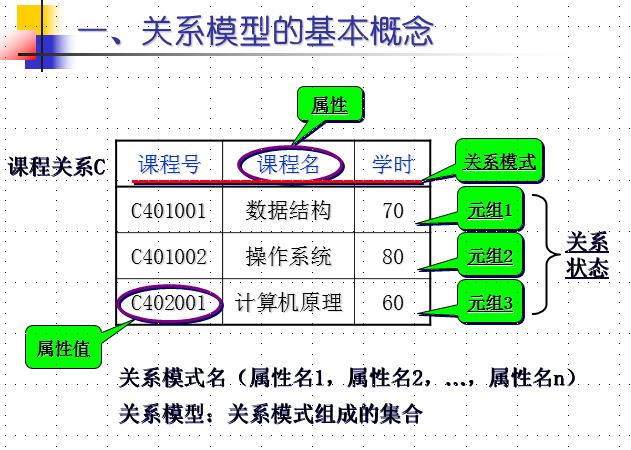
概念模型向DBMS支持的数据模型的转换，本质上是对信息世界的进一步抽象，

是信息世界的形式化和数据化

% 数据模型定义为一组面向计算机的概念集合，由**数据结构**（静态特征）、**数据操**

**作**（动态特性）、**数据约束**（数据静态特征和动态特性的限定）三部分组成

1. 关系的候选键、主键及外键，主属性和非主属性的概念



1、候选键： 如果一组属性集能唯一地标识一个关系中的元组而又不含有多余的属性

% 邮寄地址（城市名，街道名，邮政编码） 有两个候选键:

{城市名，街道名} 和 {街道名，邮政编码}

2、主键：用户选定的那个候选键 % ■ 唯一性 ■ 非冗余性■ 有效性



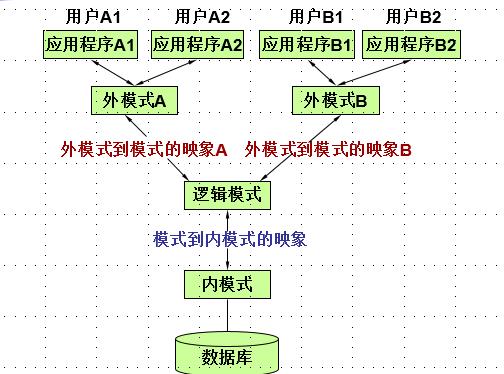
4 主属性：

包含在任何一个候选键中的属性称为主属性。

5 非主属性：

不属于任何候选键中的属性称为非主属性或非键属性 。

1. **简答2**三级模式结构及两级映像。什么是数据库的逻辑数据独立性/物理数据独立性？



逻辑数据独立性：当数据库的逻辑模式结构因某种原因修改时，只要没有改变逻辑模式中与外模式定义有关的属性及与其关系模式名的隶属关系，就可使外模式保持不变，从而不需修改应用程序。

物理数据独立性：当数据库的内模式由于某种原因要修改时，可通过对逻辑模式与内模式之间的映象的修改，使逻辑模式尽可能地保持不变，实现内模式的改变尽可能地不修改应用程序。

1. 数据库管理系统的功能。

1定义数据库

2 操纵数据库

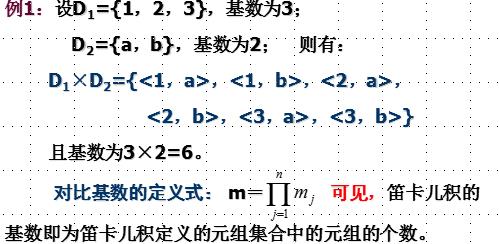
3控制数据库

4维护数据库

5通信功能

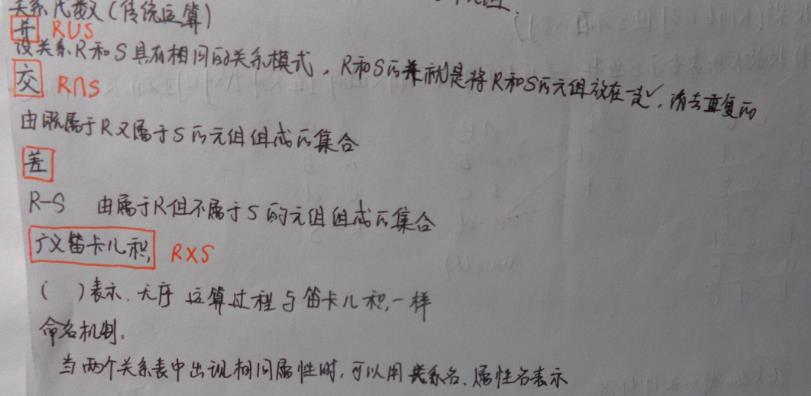
1. **关系数据库数学模型**

基数：

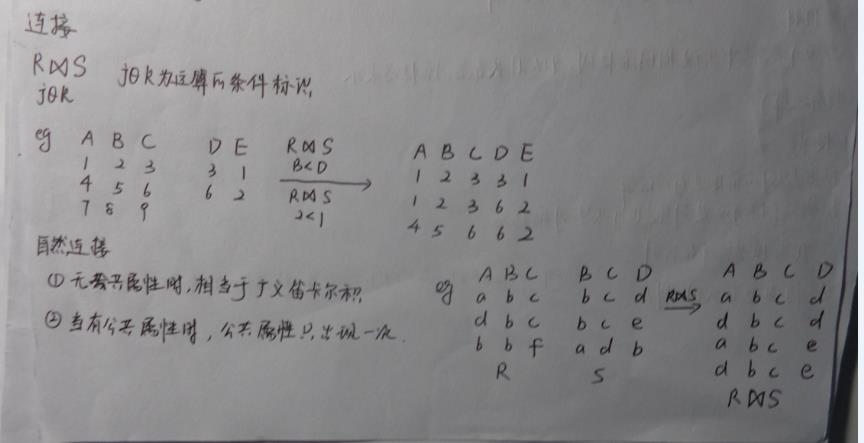
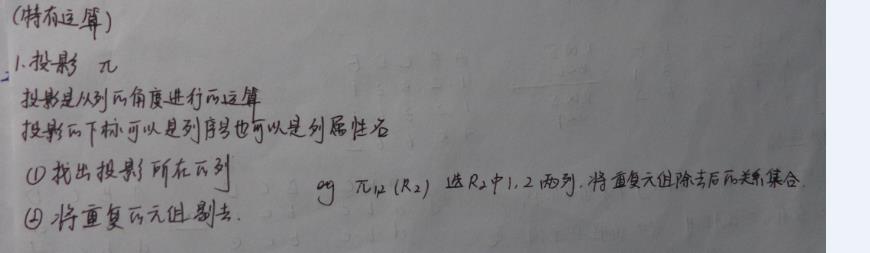


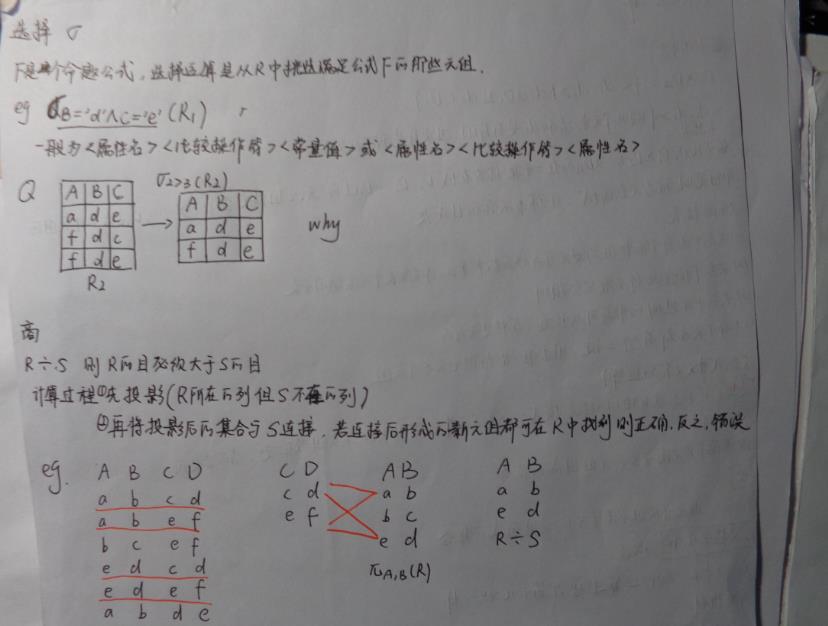
% 值域集合D1，D2，… ，Dn是关系中元组的取值范围，称为关系的域（Domain），n称为关系的目或度（Degree）。

1. 传统的**集合运算**：并，交，差，广义笛卡儿积



1. 专门的**关系运算**：选择，投影，连接，除，自然连接





1. **计算题1**用关系代数表达式表示相关的数据库查询请求及查询结果。

关系模式的表示：

1、学生关系（学号，姓名，性别，出生年月， 籍贯，专业代码，班级）

2、专业关系（专业代码，专业名称）

3、课程关系（课程号，课程名，学时）

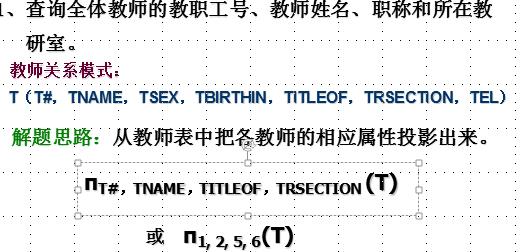
4、设置关系（专业代码，课程号）

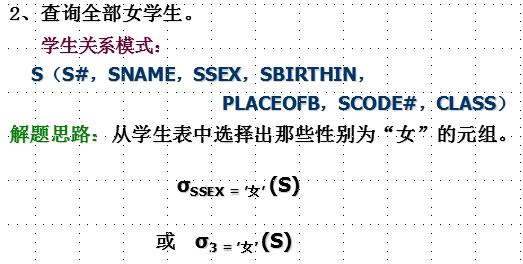
5、学习关系（学号，课程号，分数）

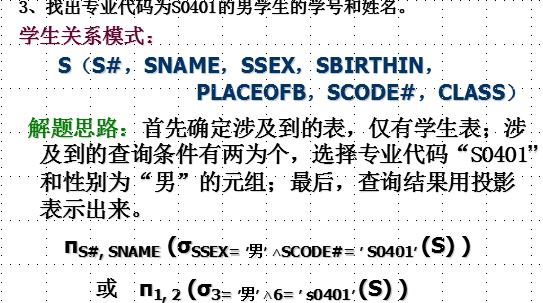
6、讲授关系（教职工号，课程号）

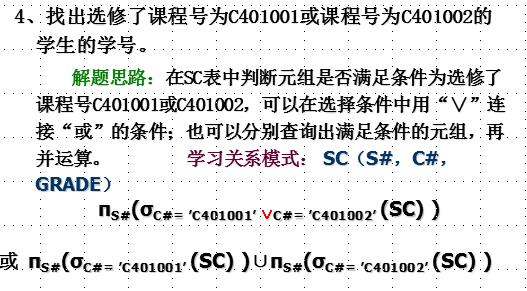
7、教师关系（教职工号，姓名，性别，出生年月，职称，教研室，电话）

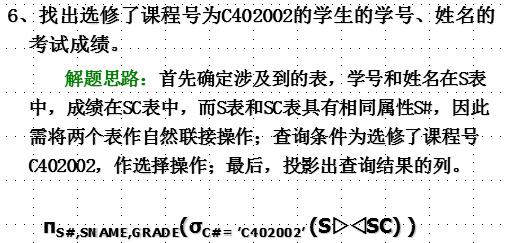


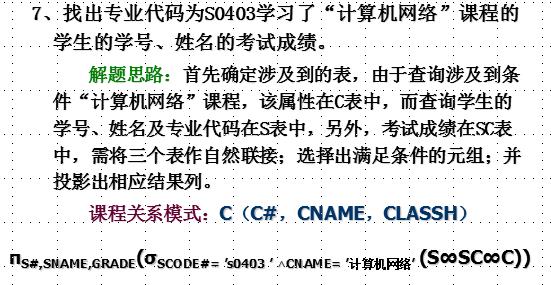












练习

12．关系代数表达式：由五个基本操作经过有限次复合的式子称为代数表达式。这种表达式的运算结果仍是一个关系。可以用关系代数表达式表示各种数据查询操作。

例：教学数据库中的四个关系如下：

教师关系T（T#，TNAME，TITLE）

课程关系C（C#，CNAME，T#）

学生关系S（S#，SNAME，AGE，SEX）

选课关系SC（S#，C#，SCORE）

使用关系代数表达式表达下列每个查询语句。

检索学习课程号为C2课程的学生学号与成绩。

πS#，SCORE（σC#=‘C2‘ （SC））或π1，3（σ2=‘C2‘ （SC））

2）检索学习课程号为C2课程的学生学号和姓名。

πS#，SNAME（σC#=‘C2‘ （S∞SC））

3）检索至少选修LIU老师所授课程中一门课程的学生学号与姓名。

πS#，SNAME（σTNAME=‘LIU‘ （S∞SC∞C∞T））

4）检索选修课程号为C2或C4课程的学生学号。

πS#（σC#=‘C2‘ ∨ C#=‘C4‘ （SC））

5）检索至少选修课程号为C2和C4课程的学生学号。

π1（σ1=4 ∧ 2=‘C2‘ ∧ 5=‘C4‘ （SC×SC））

6）检索不学C2课程的学生姓名与年龄。

πSNAME，AGE（S）-πSNAME，AGE（σC#=‘C2‘ （S×SC））

7）检索学习全部课程的学生姓名。

πSNAME（S∞（πS#，C#（SC）÷πC#（C）））

8）检索所学课程包含学号为S3学生所学课程的学生学号。

πS#，C#（SC）÷πC#（σS#=‘S3‘ （SC））

总结：查询语句的关系代数表达式的一般形式是：

π…(σ…(R×S))或π…(σ…(R∞S))

**%** 总结：将一个汉语查询语句表示成一个关系代数表达式时，应从如下几个方面入手

1、确定结果关系的属性有哪些？分别来自哪些关系? π

2、是否有选择条件？条件涉及哪些关系？应用何种运算符表示条件？σ、÷

3、查询如果涉及多个关系，应用何种运算符进行联接？ ×、∞、联接、∪、∩、－

1. （不考）关系演算分为元组关系演算与域关系演算，前者以元组为变量，后者以域为变量。

元组关系演算

一般形式： {t|ϕ(t)}

R(t) 表示：t是关系R的元组。

t[j]θu[k] 表示：元组t的第j个分量与元组u的第k个分量满足θ运算

自由元组变量和约束元组变量

　 　在一个公式中，一个元组变量称为约束元组变量，当且仅当这个元组变量前面有存在量词∃或全程量词∀，反之，则称这个元组变量为自由元组变量。

域关系演算

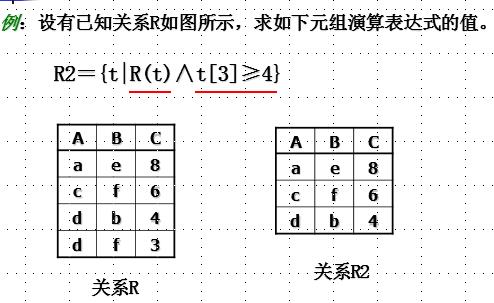
一般形式：{x1,x2,…,xn|ϕ(x1,x2,…,xn)}

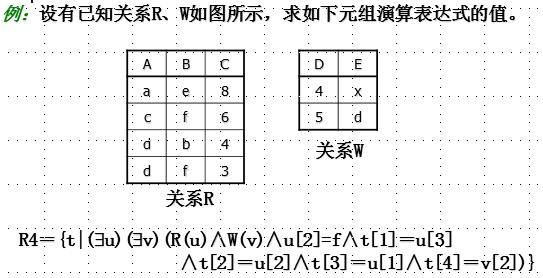
　其中，x1,x2,…,xn为域变量，也即元组变量t的各个分量，ϕ(x1,x2,…,xn)是一个域演算公式。

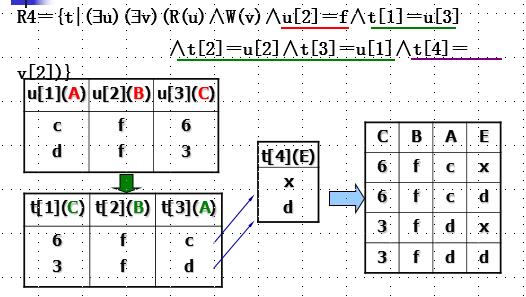
xjθyk 涵义：xj是域变量，即元组x的第j个分量，yk为元组y的第k个分量。含义为xj和yk应取使xjθyk为真的值。

1. **(不考)** 会计算（简单）元组演算表达式和域演算表达式的结果

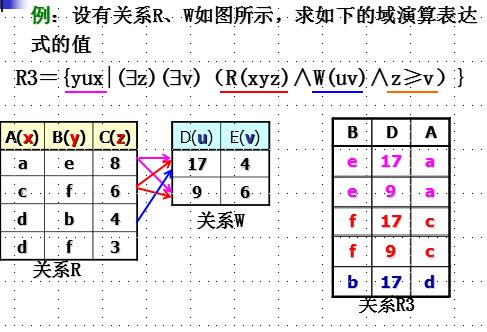
元组关系演算







域关系演算

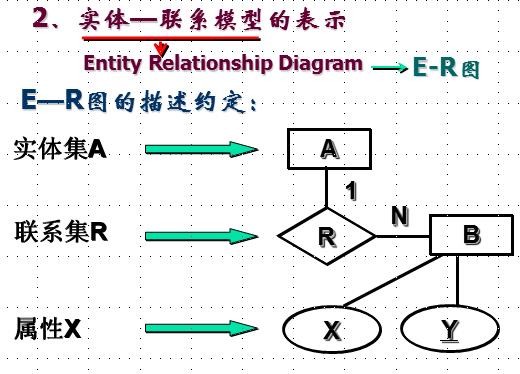


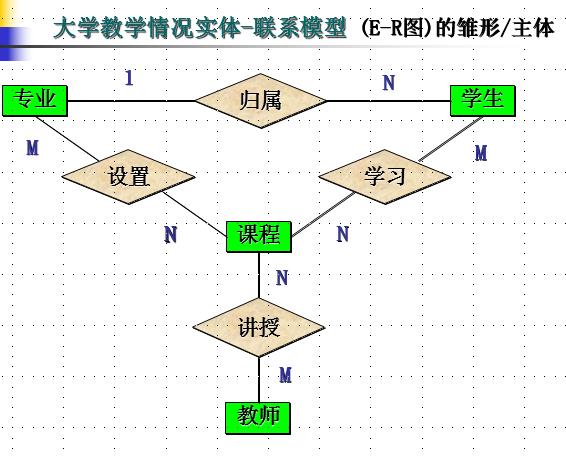
1. **数据库应用系统设计方法**
2. 数据库应用系统的生命周期分为四个时期：用户需求分析、数据库设计、数据库实现、数据库运行与维护；

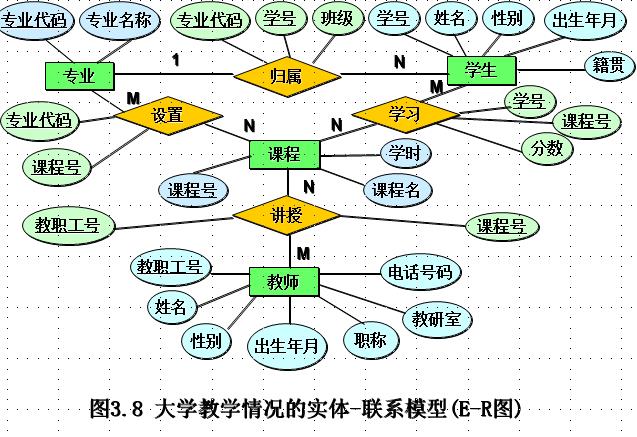
七个阶段：用户需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库结构创建、数据库应用行为设计与实现、数据库运行与系统维护

1. 数据库概念结构设计有两种方法：属性表概念结构设计和实体-联系模型

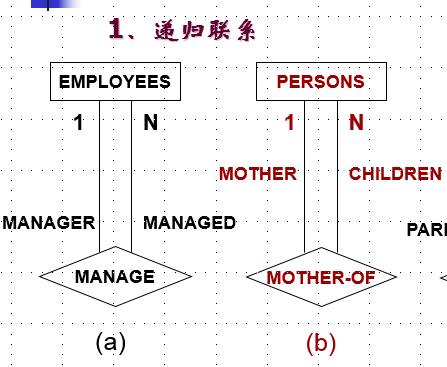
% ER图

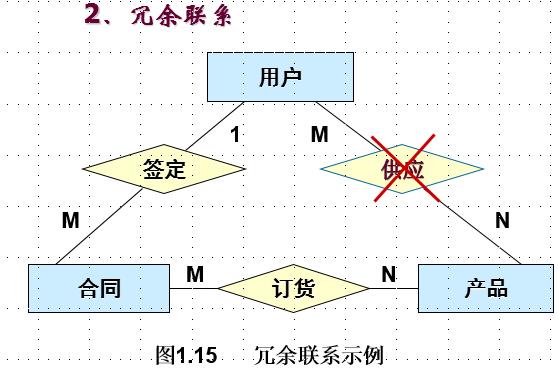


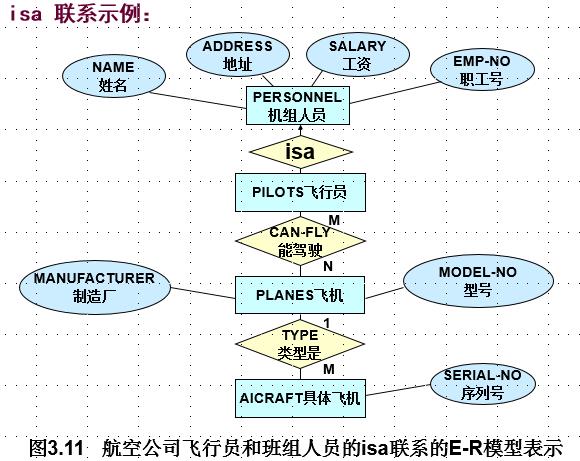


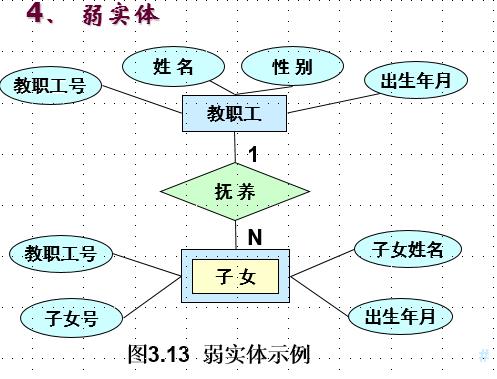


实体—联系模型设计中的一些特殊情况









1. **简答3**基于ER图的概念结构设计步骤：（1）分E—R图的设计；（2）总体E—R图的设计；（3）总体E—R图优化

%总体E—R图的设计：消除命名冲突、消除属性特征冲突、消除结构冲突

1. **应用题1**绘制ER图（参见群文件ER图）
2. 数据库逻辑结构用关系模型表达
3. **简答4**数据库逻辑结构的设计步骤：

（1）将由E-R图表示的概念结构转换成关系模型；

（2）利用规范化理论对转换好的关系模型进行规范化设计和处理；

（3）对关系模型进行优化处理。

1. 掌握ER模型向关系模型转换的规则，能够根据ER图转换为关系模型，能够确定关系模型中各关系的候选键

多对多联系向关系模型的转换

一对多联系向关系模型的转换

一对一联系向关系模型的转换

1. **简答5**数据库物理结构的设计步骤：

（1） 根据所选用的RDBMS所提供的存储结构和存储方法，为逻辑模型选定其在物理存储设备上的存储结构和存取方法；

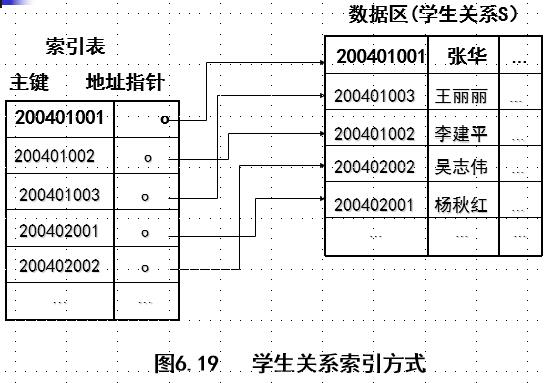
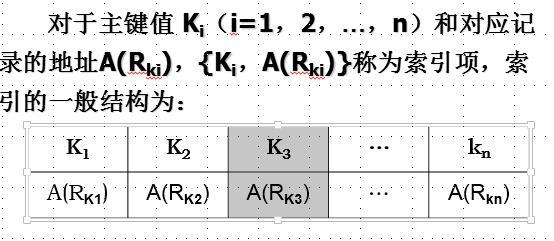
（2）评价设计的数据库存储结构和存储方法；

（3）对逻辑结构或物理结构进一步优化设计。

1. 提高关系数据库中表的查询速度的方法：

表建立索引和为多个表建立聚簇。

%索引



%索引的分类{组织方式} **线性索引 树形索引**

线性索引分为

稠密索引（按主键值的排序建立索引项，每个索引项包含一个主键值和由该主键值标识的记录的地址指针。；索引项的个数与记录的个数相等 较多）

稀疏索引 1）所有数据记录按主键值顺序存放在若干个块中 。

2）每个块的最大主键值（即该块最后一个数据记录的主键值）和该块的起始地址组成一个索引项 。

3）每个块中的索引项按主键值顺序排列组成索引表（**由于是每个块只有一个索引项**，索引项较少，所以称为稀疏索引）

1. 数据聚簇的基本思想：使那些经常在一起进行联接查询的表的数据在物理介质上尽量临近存放，也即把它们聚集地存放在一起。
2. **SQL2005数据库与数据表的创建**
3. 填空选择SQL Server2005的身份验证方式：windows 身份验证、混合模式身份验证
4. 填空选择SQL Server的数据库存储组织：一个SQL Server数据库由数据文件和日志文件组成； 文件由盘区组成； 盘区由页面组成。

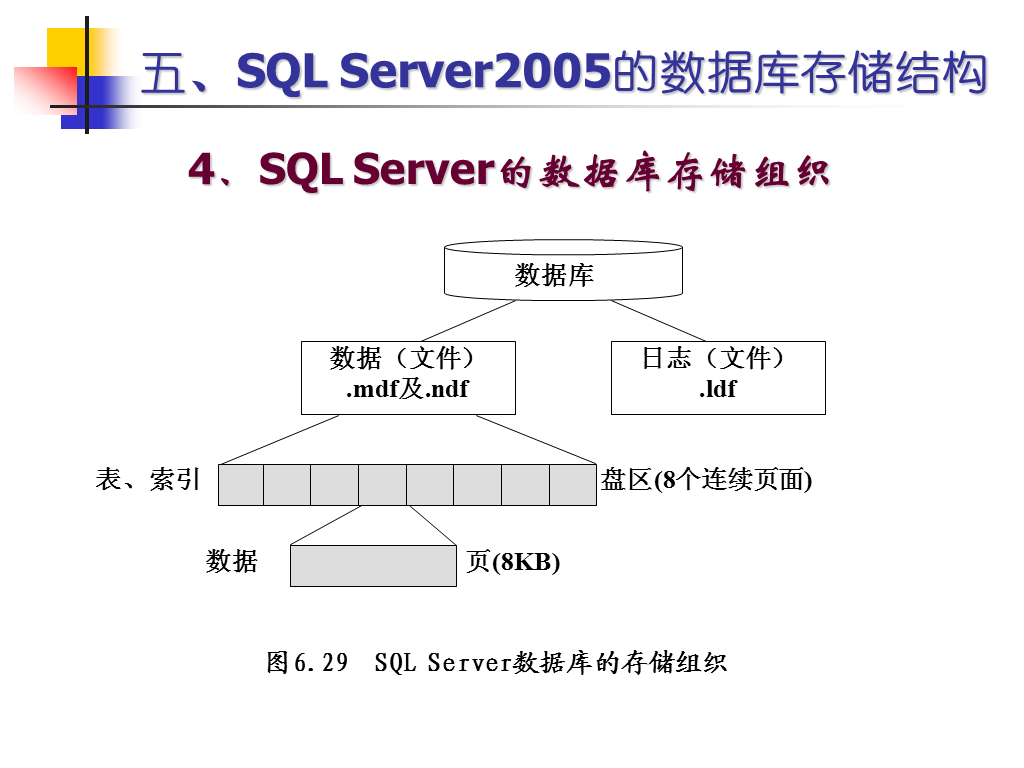


图 SQL Server数据库的存储组织

1. 填空选择安装SQL Server2005时，系统会创建四个系统数据库：Master、Model、Msdb和Tempdb
2. 填空选择 应用创建数据库的方式：SQL Server Management Studio工具和SQL语句
3. 填空选择创建表和索引的方式：表设计器和SQL语句
4. 填空选择SQL Server2005中包含聚集索引、非聚集索引和唯一索引
5. **数据库的创建与管理**
6. SQL的功能和特点

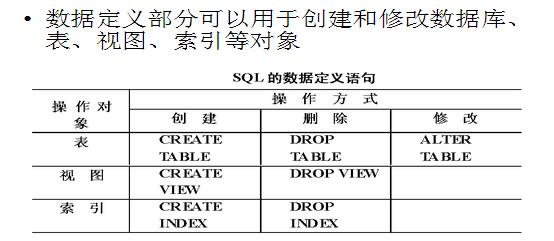
SQL的功能 数据定义功能 数据操纵功能 数据控制功能

（数据操纵功能包括：数据查询和数据更新 数据更新又分成插入、删除和修改三种操作）

SQL的特点 1 SQL具有交互式命令和嵌入式2种工作方式 2 SQL支持数据库的三级模式结构

1. 三类SQL语句：数据定义语句、数据操纵语句（数据查询语句和数据更新语句）和数据控制语句





1. 应用1熟悉表的创建，修改，撤销的SQL语句

基本表结构的创建

语法：CREATE TABLE <基本表名>

（<列名 类型>，

……

<完整性约束>，

……）

说明：完整性约束常用以下几种子句：主键子句（PRIMARY KEY）、外键子句（FOREIGN KEY）等

NOT NULL：表示不允许某一列的值为空值

例：对于教学数据库中的四个关系：

教师关系T（T#，TNAME，TITLE）

课程关系C（C#，CNAME，T#）

学生关系S（S#，SNAME，AGE，SEX）

选课关系SC（S#，C#，SCORE）

创建基本表。

CREATE TABLE T

（T# CHAR（4） NOT NULL，

TNAME CHAR（8） NOT NULL，

TITLE CHAR（10），

PRIMARY KEY（T#））

CREATE TABLE C

（C# CHAR（4） NOT NULL，

CNAME CHAR（10） NOT NULL，

T# CHAR（4），

PRIMARY KEY（C#），

FOREIGN KEY（T#） REFERENCES T（T#））

CREATE TABLE S

（S# CHAR（4） NOT NULL，

SNAME CHAR（8） NOT NULL，

AGE SMALLINT，

SEX CHAR（2），

PRIMARY KEY（S#））

CREATE TABLE SC

（S# CHAR（4），

C# CHAR（4），

SCORE SMALLINT，

PRIMARY KEY（S#，C#），

FOREIGN KEY（S#） REFERENCES S（S#），

FOREIGN KEY（C#） REFERENCES C（C#））

基本表结构的修改

1）增加新的列

语法：ALTER TABLE <基本表名> ADD <列名> <类型>

2）删除原有的列

语法：ALTER TABLE <基本表名> DROP <列名> [CASCADE | RESTRICT]

说明：CASCADE方式表示在基本表中删除某列时，所有引用到该列的视图和约束也要一起自动地被删除；RESTRICT方式表示在没有视图或约束引用该属性时，才能在基本表中删除该列，否则拒绝删除。

3）修改原有列的类型、宽度

语法：ALTER TABLE <基本表名> MODIFY <列名> <类型>

基本表的撤销

语法：DROP TABLE <基本表名> [CASCADE | RESTRICT]

1. 熟悉表中数据的插入，修改，删除的SQL语句

数据的插入 **INSERT INTO <表名>**

**[（<列名表>）]**

**VALUES（<值表>）；**

例 写出给学习关系SC中插入王丽丽同学（学号为200401003）学习计算机网络课（课程号为C403001）的成绩（89分）的插入语句。

INSERT INTO SC

（S#，C#，GRADE）

VALUES（＇200401003＇，＇C403001＇，89）；

数据的修改  **UPDATE <表名>**

**SET <列名1>＝<表达式1>[，<列名2>＝<表达式2>, …, <列名n>＝<表达式n>]**

**[WHERE <条件>]**

例 写出将学生关系S中的学生名字“王丽丽” （学号为200401003）改为“王黎丽”的数据修改语句。

UPDATE S

SET SNAME＝＇王黎丽＇

WHERE S#＝＇200401003＇；

数据的删除  **DELETE FROM <表名>**

**[WHERE <条件>]**

例 从学生关系S中删除学号为200403001的学生的信息。

DELETE FROM S

WHERE S# =＇200403001＇；

1. 熟悉Select查询语句，简单查询，连接查询，嵌套查询，集合查询

SELECT语句完整的句法

SELECT <目标表的列名或列表达式序列>

FROM <基本表名和（或）视图序列>

[WHERE <行条件表达式>]

[GROUP BY <列名序列>

[HAVING <组条件表达式>]]

[ORDER BY <列名[ASC | DESC]>,…]

说明：[]表示其中的内容根据需要可以省略。

|表示二选一。

聚合函数

COUNT（\*）：计算元组个数

COUNT（<列名>） 对一列中的值计算个数

SUM（<列名>） 求某一列值的总和（此列值为数值型）

AVG（<列名>） 求某一列值的平均值（此列值为数值型）

MAX(<列名>) 求某一列值的最大值

MIN(<列名>) 求某一列值的最小值

1. 应用2 视图的定义

**CREATE VIEW <视图名> [(<视图列名表>)]**

**AS <SELECT 语句>**

**[WITH READ ONLY | WITH CHECK OPTION]；**

例 写出教学安排视图：

TA(C#，CNAME，CLASSH，TNAME，TRSECTION）

的创建语句。

课程关系模式：C（C#，CNAME，CLASSH）

教师关系模式：T（T#，TNAME，TSEX，TBIRTHIN，TITLEOF，TRSECTION，TEL）

讲授关系模式： TEACH（T#，C#）

CREATE VIEW TA

AS SELECT C#，CNAME，CLASSH，TNAME，TRSECTION

FROM C，TEACH，T

WHERE C.C#=TEACH.C# AND TEACH.T#=T.T#；

1. 可以在视图上进行查询操作，但于基本表相比，视图中不保存数据，是一种“虚表”。
2. **？？？？应用3** 游标的声明和使用

**DECLARE <游标名> [INSENSITIVE][SCROLL] CURSOR**

**FOR <SELECT语句>**

**[FOR { READ ONLY | UPDATE [OF <列名> [，… ]]}]**

例假设专业表SS中的当前值如表5.10所示。

表5.10 专业表SS的当前值

SCODE# SSNAME

S0401 计算机科学与技术

S0402 指挥自动化

S0403 网络工程

*S0404* 信息研究与安全

利用游标机制查询专业关系中的所有数据记录。

USE JXGL

GO

DECLARE CC2 CURSOR /\* 定义游标CC2，默认为FORWARD\_ONLD游标\*/

FOR SELECT \*

FROM SS;

OPEN CC2; /\* 打开游标 \*/

FETCH NEXT FROM CC2; /\* 读取第1行数据 \*/

/\* 用WHILE循环语句控制游标的执行，当正常读出1行时，

继续循环，否则跳出循环停止FETCH操作 \*/

WHILE @@FETCH\_STATUS=0

BEGIN

FETCH NEXT FROM CC2;

END

CLOSE CC2; /\* 关闭游标 \*/

DEALLOCATE CC2; /\* 删除不再使用的游标 \*/

GO

1. 在嵌入式应用中，嵌入式SQL语句与主语言之间的通信用主变量来实现。嵌入式SQL语句中有前缀符号“@”的变量为主变量。
2. **关系数据库模式设计**
3. **简答**对关系模式进行规范化设计的必要性

当关系模式设计的不合理时，就会存在数据冗余、更新异常、插入异常、删除异常。

（不满足关系范式造成的存储异常 ）

1. 函数依赖的公理体系：自反律、增广律、传递律；合并规则、分解规则和伪传递规则
2. **计算**X关于F闭包的计算
3. **计算**最小函数依赖集的求解
4. **计算**无损连接性和保持函数依赖的验证计算
5. 完全函数依赖，部分函数依赖和传递函数依赖的概念。

部分依赖和完全依赖：对于FD W→A，如果存在X⊂W有X→A成立，那么称W→A是部分依赖（A部分依赖于W）；否则称W→A是完全依赖。

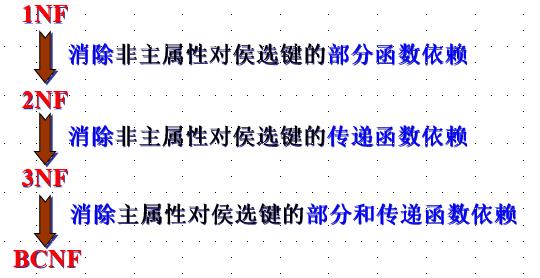
如果X→Y，Y→A，且Y→X和A⊆Y，那么称X→A是传递依赖（A传递依赖于X）。

1. **简答题**规范化理论中第一范式、第二范式、第三范式、BC范式的概念和之间的关系

第一范式（1NF）：如果关系模式R的每个关系r的属性值都是不可分的原子值，那么称R是第一范式的模式。满足1NF的关系称为规范化的关系，否则称为非规范化的关系。1NF是关系模式应具备的最起码的条件。

第二范式（2NF）：如果关系模式是1NF，且每个非主属性完全函数依赖于候选键，那么称R是第二范式（2NF）的模式。

第三范式（3NF）：如果关系模式R是2NF，且每个非主属性都不传递依赖于R的候选键，那么称R是第三范式（3NF）的模式



1. **T-SQL与存储过程**
2. T-SQL语言要素，常量、变量（局部变量、全局变量）、运算符、表达式、系统函数、自定义函数
3. 各种流程控制语句：语句块定义、判断结构、循环结构
4. 自定义函数：根据输入的参数，求两数之和，或两数的较大值，并返回结果
5. 自定义函数：定义单语句表值函数，根据输入参数，查询表中符合条件的记录，
6. **应用题**创建一个存储过程，根据输入的参数，查询表中符合条件的记录
7. **数据库应用系统体系结构与访问技术**

客户机/服务器（Client/Server，C/S）

浏览器/服务器（ Browser/Server，B/S ）

1. 数据库应用系统体系结构的变迁：集中式计算模式-》C-S结构-》B-S结构
2. **简答题**C-S结构、B-S结构及两种结构的区别

C-S结构建立在局域网的基础上，B-S结构建立在广域网的基础上

两层C-S结构 有客户机和服务器，三层C-S结构分为表示层、功能层和数据层（与三层B-S结构同）

主要区别：1硬件环境不同

2安全性要求不同（C-S结构强、B-S结构弱）

3程序架构不同（C-S结构注重流程、B-S需要对安全性和访问速度多重考虑）

4软件重用不同（C-S结构可重用性差、B-S结构 好）

5系统维护需求不同（C-S结构开销大、B-S结构开销小）

6处理问题不同（C-S结构用户面固定、B-S结构建立在广域网上，用户面分散）

7用户接口不同（C-S结构 建立在window、B-S结构 建立在浏览器）

1. ODBC体系结构及控制机制（？）

**O**DBC应用体系结构主要由：

客户端的数据库应用程序、ODBC应用程序接口（ODBC API）、 ODBC驱动程序管理器(ODBC Driver Manager)、驱动程序（Driver）、 数据源（Data Source）；

3、ADO.net组件中包含两个核心组件：SQL Server.NET Framework数据提供程序和数据集DataSet。

4、访问SQL SERVER数据源的数据提供程序是：SQL Server.NET Framework数据提供程序

5、**简答题**ADO.NET访问数据库的方法及步骤（断开式数据访问方法及其操作过程，连接式数据访问方法及其操作过程）

1 使用连接对象Connection连接数据源

2 使用命令对象Command执行SQL语句或存储过程操纵数据库

3 使用数据读取器对象DataReader读取数据

4 使用数据集对象DataSet和数据适配器DataAdapter访问数据库

1. 数据绑定概念和作用

概念 数据绑定是一种把数据绑定到一种用户界面控件的通用机制。

作用 数据绑定提供了用户界面控件和数据源之间的桥梁，数据绑定封装了对用户界面控件和数据源进行同步的复杂代码

**第九章 数据库应用程序设计**

1、VB.net + SQL Server

2、**简答题**在visual Studio环境中开发应用程序的步骤：

1）建立项目/打开项目

2）新建窗体/打开窗体

3）建立用户界面（窗体）的各个对象;

4）对象属性的可视化设置;

5）对象事件过程编码;

6）保存、调试和运行程序。

3、**简答题**在程序中连接数据库时两种用户登录验证方式：

1）'使用SQLServer用户登录验证方式连接数据库，

Private str As String = "Data Source= hostname; Initial Catalog=jxgl; User ID=sa; password=sa123"

'定义并实例化一个Connection对象

Private conn As New SqlConnection(str)

2）'使用windows用户登录验证方式连接数据库，

Private str As String = "Data Source=hostname; Initial Catalog=JXGL; Integrated Security=True"

'定义并实例化一个Connection对象

Private conn As New SqlConnection(str)

1. 引入命名空间，以使得程序可以使用ADO.NET组件的命令是:

Imports System.Data

Imports System.Data.SqlClient

5、指定窗体在屏幕中心出现，应该修改当前窗体的属性StartPosition 的属性值为 CenterScreen

6、在窗体上常用的控件有：标签label、文本框textbox、按钮 button、数据表格DataGridView、菜单menuStrip

7、常用的事件过程为按钮的单击事件。

1. 显示main窗体的命令：main.show() 隐藏主窗体的命令：main.hide()

**第十章 数据库保护技术**

1、事务的特性：原子性、一致性、隔离性、永久性

2、显式事务控制语句：

BEGIN TRANSACTION 开始一个事务

COMMIT TRANSACTION 提交一个正常完成的事务

ROLLBACK TRANSACTION 撤销一个没有正常完成的事务

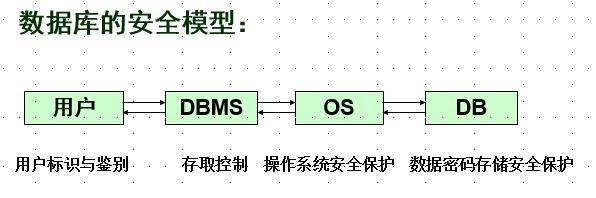
3、SQL Server的完整性约束：域完整性、实体完整性、参照完整性和用户自定义完整性，各种完整性的概念，系统为保证数据的完整性，提供了各种实现机制。

SQL Server 2005的数据完整性分类表

|  |  |  |
| --- | --- | --- |
| **完整性类型** | **实现机制** | **描述** |
| 域完整性 | DEFAULT | 指定列的默认值 |
| CHECK | 指定允许值 |
| NULL | 是否允许空值 |
| 实体完整性 | 主键约束 | 每行的惟一标识 |
| UNIQUE | 不允许有重复key |
| 参照完整性 | 外键约束 | 定义的列的值必须与某表的主键值或惟一键值一致 |
| 用户定义完整性 | CHECK | 指定允许值 |
| 触发器 | 由用户定义不属于其他任何完整性类别的特定业务规则 |

4、数据库数据的安全性威胁：数据损坏、数据篡改、数据窃取

5、**简答题**数据库系统的安全模型



6、触发器(Trigger)是一个能因某一个事件触发而由系统自动执行的SQL语句或语句序列。它可以实现查询、计算、评估、交流，及完成更复杂的功能任务。 由三部分组成：事件、条件和动作

7、给用户授权语句及格式

**GRANT <CONNECT>[，RESOURCE][，DBA]**

**TO <用户标识表>**

**[IDENTIFIED BY <口令表>]；**

收回用户授权语句及格式

**REVOKE <CONNECT>[，RESOURCE][，DBA]**

**FROM <用户名表>；**

7、数据库进行并发控制的基本方法是对被操作的数据项进行加锁。基本的锁方式有两种：排他锁（X锁）和共享锁（S锁）

**第十一章 数据库技术新发展**

1. **简答题**关系数据库系统的局限性

1 关系模型对复杂对象的表达能力差

2关系模型支持的数据类型有限

3关系数据库的管理和处理能力有限

4关系数据库的操纵语言和主语言之间存在阻抗失配

1. 各种数据库新技术的基本概念和适用领域。

分布式数据库系统 信息处理和信息管理

面向对象数据库系统 描述现实世界的实体（对象）的逻辑结构和对象间的联系与限制