

《数据库原理一》课程

复习、习题分析

2024年12月30日

第1章 数据库发展史

1. 数据管理技术的发展（特点）

人工管理阶段、文件系统阶段、倒排文件系统；

文件系统的缺陷（数据冗余性、数据不一致性、数据联系弱）。

2. 数据库技术的产生

♥ 进入数据库阶段的标志是20世纪60年代末的三件大事：

1968年IBM公司研制的IMS系统-----典型的层次DBS；

1969年美国CODASYL组织 DBTG报告提出-----网状DBS的概念；

1970年美国IBM公司的E. F. Codd发表的论文提出---关系模型的思想

♥ 数据库阶段的特点：

- ① 采用数据模型表示复杂的数据结构(数据库的三级体系结构)；
- ② 有较高的数据独立性(逻辑数据独立性、物理数据独立性)；
- ③ 数据库系统为用户提供了方便的用户接口；
- ④ 数据库系统提供四方面的数据控制功能：数据库的并发控制，
数据库的恢复，数据的完整性，数据安全性。
- ⑤ 对数据的操作以数据项为单位，增加了系统的灵活性。

3. 数据库技术的术语

4. 数据库技术的发展

第2章 数据库系统结构

1. 数据描述

从事物的特性到计算机中的具体表示，分别经历了三个阶段：

概念设计中的数据描述（实体、属性、实体集、实体标识符）

逻辑设计中的数据描述（记录、字段、文件、关键码）

物理设计中的数据描述（位、字节、字、块、桶、卷）

注意他们的差别：

概念设计中的数据描述与逻辑数据描述的差别；

逻辑数据描述与物理数据描述的差别。

存储介质的层次：图2.1

2. 数据联系的描述

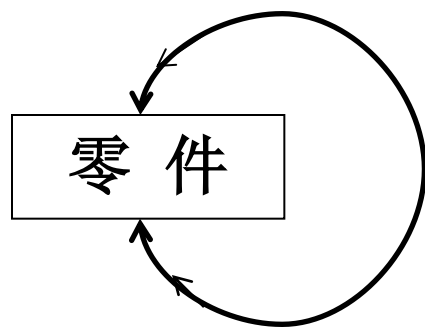
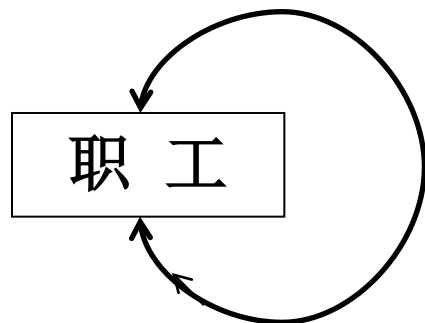
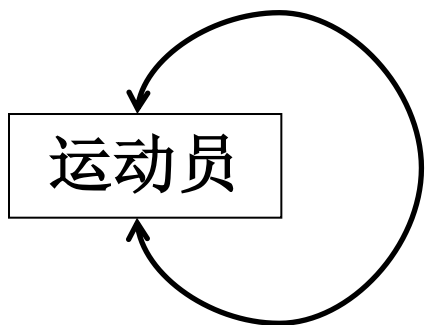
联系的元数：与一个联系有关的实体集的个数

联系的类型： 一对一 (1: 1)

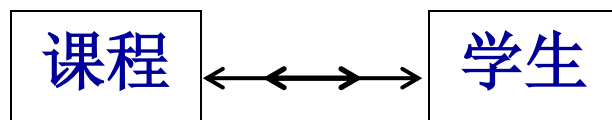
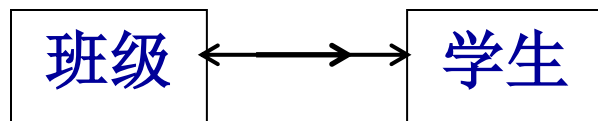
一对多 (1: n)

多对多 (m: n)

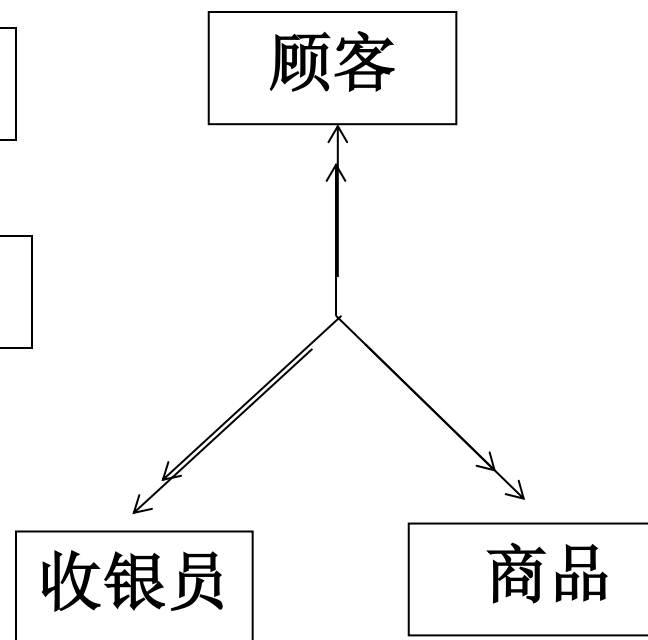
一元联系：



二元联系：



三元联系：



3. 数据模型的概念： 表示实体类型及实体间联系的模型

概念数据模型（对现实世界的第一层抽象）

是按用户的观点对数据建模，强调其语义表达能力。

(独立于计算机系统的模型，用于建立信息世界的数据库模型)。

逻辑数据模型 “结构数据模型”：（对现实世界的第二层抽象）

是直接面向数据库的逻辑结构，与DBMS有关。

(从计算机的观点对数据建模，用于建立机器世界的数据库模型)

♥ 概念数据模型--实体联系模型(ER模型)

在ER模型中有四个基本成分：

矩形框表示实体类型；

菱形框表示联系类型（实体间的联系）；

椭圆形框表示实体类型和联系类型的属性；

用直线连接： 实体与属性；

联系与属性；

实体与实体；

相应的命名均记入各种框中。

对于关键码的属性，在属性名下划一横线。

建立ER模型的步骤如下：

- ♥ 首先确定实体类型和联系类型，
- ♥ 接着把实体类型和联系类型组合成ER图；
- ♥ 然后确定实体类型和联系类型的属性，
- ♥ 再确定实体类型的键，在属于键的属性名下划一横线。

（包括第六章ER模型的扩充）

♥ **逻辑数据模型**：面向数据库逻辑结构的模型。

逻辑数据模型包含：

数据结构、数据操作和数据完整性约束。

数据结构是指对实体类型和实体间联系的表达和实现；

数据操作是指对数据库的检索和更新（包括插入、

删除、修改）两类操作的实现；

数据完整性约束给出数据及其联系应具有制约和依赖

规则。

逻辑数据模型主要有：

层次模型： 用**树型结构**来表示实体类型及实体之间联系的模型

称为层次模型。（数据之间的联系通过**指针**实现）

网状模型： 用**有向图结构**表示实体类型及实体间联系的模型

称为网状模型。（数据之间的联系通过**指针**实现）

关系模型： 用**二维表格结构**来表示实体集。（数据之间的联系

是通过**外键和主键**间联系实现的 即：公共属性...）

面向对象模型： 数据之间嵌套、递归联系是通过对象标识符实现的

4. 数据库的体系结构

三级模式结构

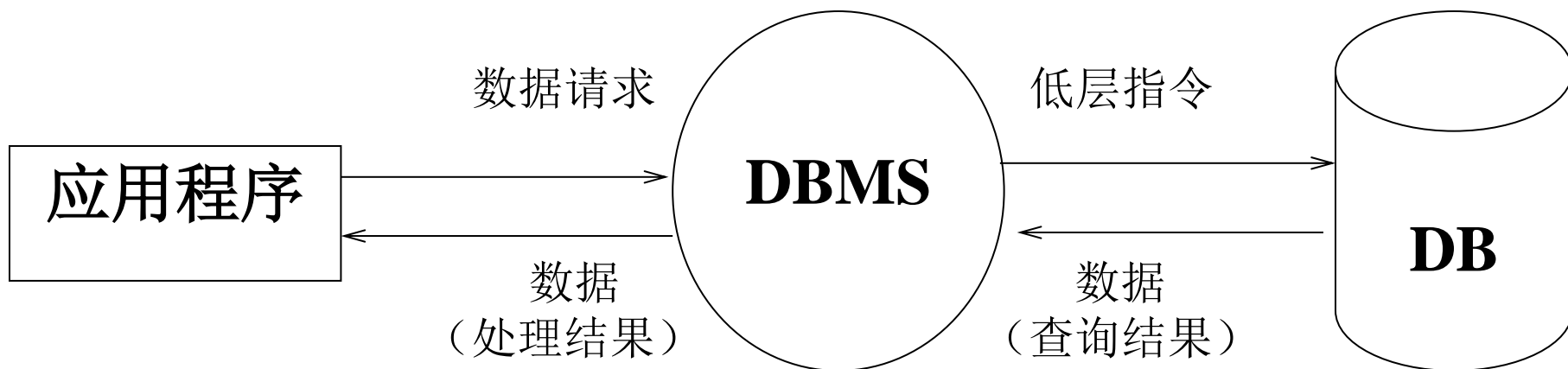
三级结构和两级映射

数据独立性定义

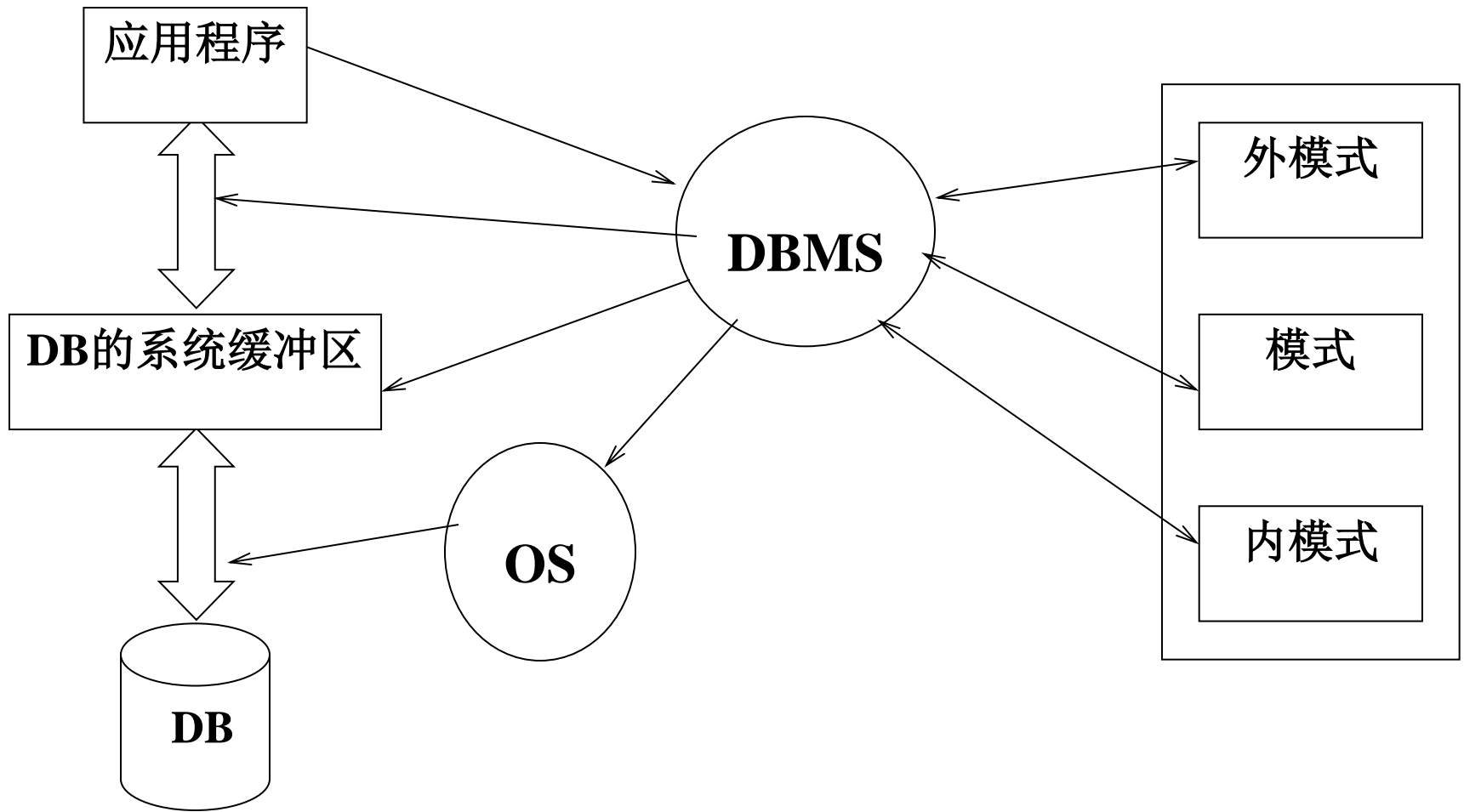
两级数据独立性： 物理数据独立性、逻辑数据独立性

5. 数据库管理系统

DBMS的工作模式:



用户访问数据的过程：



DBMS的主要功能:

1. 数据库的定义功能
2. 数据库的操纵功能
3. 数据库的保护功能
4. 数据库的维护功能
5. 数据字典

6、数据库系统DBS

DBS的组成:

数据库**DB**

硬件

软件

数据库管理员**DBA**

DBS的全局结构

DBS结构的分类

DBS的效益

数据库管理员DBA： DBA是控制数据整体结构的一组人员，负责DBS的正常运行，承担创建、监控和维护数据库结构的责任。

DBA的主要职责：

- ① 定义模式
- ② 定义内模式
- ③ 与用户的联络。包括定义外模式、应用程序的设计、
提供技术培训等专业服务。
- ④ 定义安全性规则，对用户访问数据库的授权。
- ⑤ 定义完整性规则，监督数据库的运行。
- ⑥ 数据库的转储与恢复工作。

第3章 关系运算

1、基本概念

关系模型：用二维表格表示实体集，用关键码表示实体间联系的数据模型；

关键码：超键、候选键、主键和外键；(P.37)

关系的定义和特点；（P.36-P.37,P.41）

三类完整性规则：实体完整性规则、参照完整性规则、
用户定义的完整性规则；(P.40-P.41)

ER模型到关系模型的转换规则；

过程性语言与非过程性语言：

过程性语言：编程时必须指出“干什么”及“怎么干”的语言；

非过程性语言：编程时只须指出“干什么”，不必指出“怎么干”。

♥非过程性语言与过程性语言的区别

过程性语言	非过程性语言
编程时，必须指出“怎么干”	编程时，不必指出“怎么干”
由用户进行数据导航	由系统进行数据导航
单记录处理方式	集合处理方式
属于3GL范畴	属于4GL范畴
C语言，层次、网状DML等	关系DML，软件开发工具等

关系运算包括：关系代数和关系演算两类。

2、关系代数（P.42—P.51）理论基础：集合运算

五个基本操作（并、差、笛卡儿积、投影、选择）；

四个组合操作（交、联接、自然联接、除）；

三个扩充的关系代数操作（外联接、外部并、半联接）；

3、关系演算（P.52-P.56）理论基础：谓词演算

元组关系演算和域关系演算的原子公式、公式的定义。

无限关系（指元组个数为无穷多个的关系）；

无穷验证（验证公式真假时需要进行无限次验证）。

关系演算的安全性和等价性。

4、查询优化(P.57-P.64)

关系代数表达式的优化问题；

关系代数表达式的等价变换规则；

优化的策略；

优化算法、例 3.24

♥ 关系代数表达式的运用技巧

(1) 一般规则

对于只涉及到选择、投影、联接的查询可用下列表达式表示：

$$\pi_{...}(\sigma_{...}(\mathbf{R} \times \mathbf{S})) \quad \text{或者} \quad \pi_{...}(\sigma_{...}(\mathbf{R} \bowtie \mathbf{S}))$$

(2) 对于否定的操作，一般要用差操作表示；

例如：“检索不学C2课的学生姓名”。

不能用下式表示： $\pi_{\text{SNAME}, \text{AGE}}(\sigma_{\text{CNO} \neq 'C2'}(\mathbf{S} \bowtie \mathbf{SC}))$

一定要用“差”的形式：

$$\pi_{\text{SNAME}, \text{AGE}}(\mathbf{S}) - \pi_{\text{SNAME}, \text{AGE}}(\sigma_{\text{CNO} = 'C2'}(\mathbf{S} \bowtie \mathbf{SC}))$$

(3) 对于具有“全部”或“包含”问题的检索,一般要用除法操作表示。

例：“检索学习全部课程的学生学号”：

$$\pi_{\text{SNO}, \text{CNO}} (\text{SC}) \div \pi_{\text{CNO}} (\text{C})$$

不能表示为： $\pi_{\text{SNO}} (\text{SC} \div \pi_{\text{CNO}} (\text{C}))$

- 3.12已知关系模式

S (Sno, SName, Age, Sex, Sdept)

C (Cno, CName, Cdept, Tname)

SC (Sno, Cno, Grade)

- 例1 求选修Liu所授课程的课程号和课程名

- $\Pi_{\text{cno, cname}} \sigma_{\text{tname} = \text{'liu'}} (C)$

- 例2 求年大于23岁的男学生的学号和姓名

- $\Pi_{\text{sno, sname}} \sigma_{\text{age} > 23 \wedge \text{sex} = \text{'男'}} (S)$

- 例3 求s3学生所学课程的课程名和任课教师名。

- $\Pi_{\text{cname, tname}} \sigma_{\text{sno} = 's3'} (\text{SC} \bowtie \text{C})$

- 例4 求选修liu老师所授课程女学生姓名

- $\Pi_{\text{sname}} \sigma_{\text{tname} = 'liu' \wedge \text{sex} = '女'} (\text{S} \bowtie \text{SC} \bowtie \text{C})$

- 例5 求wan同学不选修课程的课程号

- $\Pi_{\text{cno}} (\text{C}) - \Pi_{\text{cno}} \sigma_{\text{sname} = 'wan'} (\text{S} \bowtie \text{SC})$

- 例6 求至少选修两门课程的学生号码。

- $\Pi_1 \sigma_{1 = 4 \wedge 2 < 5} (SC \times SC)$

- 例7 求全部学生都选修的课程号和课程名

- $\Pi_{cno, cname} (\Pi_{sno, cno} (SC) \div \Pi_{sno} (S) \bowtie C)$

- 例8 求选修liu老师所授全部课程的学生号码。

- $\Pi_{sno, cno} (SC) \div \Pi_{cno} \sigma_{tname='liu'} (SC \bowtie C)$

第4章 关系数据库语言SQL (P. 67–P. 97)

1. SQL数据库的体系结构，SQL的组成。
2. SQL的数据定义：SQL模式、基本表和索引的创建和撤销，SQL提供的数据类型。
3. SQL的数据查询：SELECT语句的格式，单表和多表查询，基本表的联接操作，聚合和分组，集合操作。
4. SQL的数据更新：插入、删除和修改语句。
5. 视图的创建和撤销，对视图更新操作的限制。
6. 嵌入式SQL：预处理方式，使用规定，使用技术。

内容分析

SELECT语句是SQL的核心内容，应掌握下列内容。

1. SELECT语句的来历

在关系代数中最常用的式子是下列表达式：

$$\pi_{A_1, \dots, A_n}(\sigma_F(R_1 \times \dots \times R_m))$$

这里 R_1 、 \dots 、 R_m 为关系， F 是公式， A_1 、 \dots 、 A_n 为属性。

为此SQL设计成SELECT—FROM—WHERE句型：

```
SELECT  A1, ..., An  
FROM    R1, ..., Rm  
WHERE   F
```

该句型是从关系代数表达式演变来的，但WHERE子句中的条件表达式 F 要比关系代数中公式更灵活。

2. SELECT语句中出现的基本表名，应理解为基本表中的元组变量，而列名应理解为元组分量。

3. SELECT语句的语义通常有三种情况（SQL标准）：

以学生表S（SNO，SNAME，AGE，SEX）为例说明。

第一种情况：SELECT语句中未使用分组子句，也未使用聚合操作，

那么SELECT子句的语义是对查询的结果执行投影操作。

如：

```
SELECT SNO, SNAME  
  
FROM S  
  
WHERE SEX='M'；
```

第二种情况： SELECT语句中未使用分组子句，但在SELECT子句中使用了聚合操作，此时SELECT子句的语义是对查询结果执行聚合操作。

如： SELECT COUNT (*) , AVG (AGE)
 FROM S WHERE SEX='M' ;

该语句是求男同学的人数和平均年龄。

第三种情况： SELECT语句使用了分组子句和聚合操作, 此时SELECT子句的语义是对查询结果的每一分组去做聚合操作。

如：求男同学每一年龄的人数。

```
SELECT AGE, COUNT (*)
```

```
FROM S
```

```
WHERE SEX='男'
```

```
GROUP BY AGE;
```

SQL的数据更新：插入、删除和修改语句 (P.89-P.91)

* 数据插入

1、插入单个元组：

INSERT INTO 基本表名（列名表）

VALUES （元组值）

2、插入子查询的结果：

INSERT INTO 基本表名（列名表）

SELECT 查询语句；

* 数据删除

删除关系中满足条件的元组语句的句法如下：

DELETE FROM <表名>

WHERE <条件表达式>

* 数据修改

UPDATE 基本表名

SET 列名=值表达式[, 列名=值表达式...]

[WHERE 条件表达式]

视图的更新操作 (P. 91-P. 92)

对于视图元组的更新操作 (INSERT、DELET、UPDATA)，有以下

三条规则：

- ① 如果一个视图是从多个基本表使用联接操作导出的，那么不允许对这个视图执行更新操作。
- ② 如果在导出视图的过程中，使用了分组和聚合操作，也不允许对这个视图执行更新操作。
- ③ 如果视图是从单个基本表使用选择、投影操作导出的，并且包含了基本表的主键或某个候选键，那么这样的视图称为“行列子集视图”，并且可以被执行更新操作。

在SQL2中，允许更新的视图在定义时，必须加上“WITH CHECK OPTION”短语。

嵌入式SQL的使用技术：

嵌入式SQL的实现，有两种处理方式：

- 1、扩充宿主语言的编译程序，使之能处理SQL语句；
- 2、采用预处理方式。

目前多数系统采用后一种方式。

宿主语言+嵌入式SQL



预处理程序



宿主语言+函数调用



宿主语言编译程序



目标程序

* 嵌入式SQL使用时必须解决的问题

①为区分SQL语句与宿主语言语句，在所有的SQL语句前必须加上前缀标识“EXEC SQL”，并以“END EXEC”作为语句结束标志。

② 数据库工作单元和主程序工作单元之间的通讯

允许嵌入的SQL语句引用宿主语言的程序变量（称为共享变量）。在引用这些变量时必须在这些变量前加冒号“：”作为前缀标识，以示与数据库中变量有区别；

③ 引入游标机制： 将集合操作转换为单元组处理。

* 与游标有关的语句:

① 定义游标语句 (DECLARE)

② 打开游标语句 (OPEN)

③ 游标推进语句 (FETCH)

④ 关闭游标语句 (CLOSE)

* 嵌入式SQL的使用技术

第六章 实体联系模型

实体联系模型(ER模型):.....概念模型设计方法。

- 实体与属性的基本概念, 属性的分类, 联系的元数、连通词、基数。
- 扩展的实体联系模型的表示方法;
- 如何使用实体联系模型进行概念设计的方法。