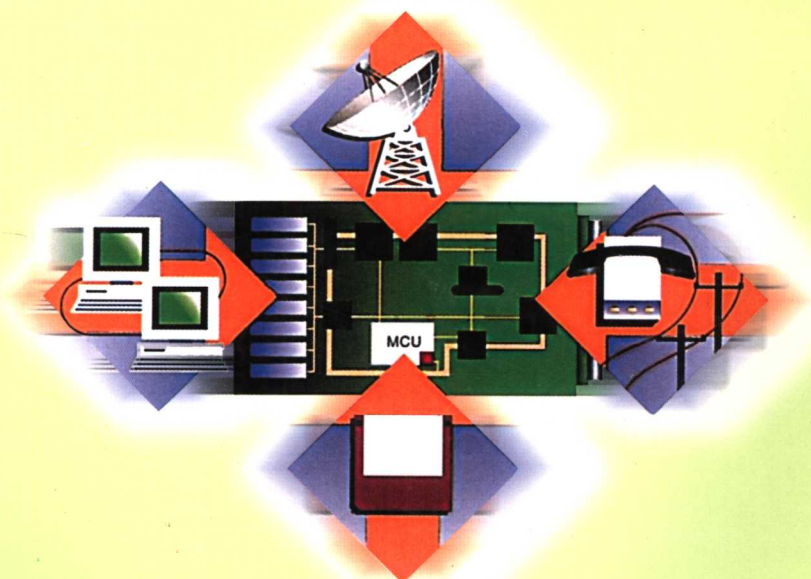


 新世纪高等学校
计算机专业教材系列

数据库原理及其应用教程

——学习指导、例题分析、习题解答与标准试题库

黄德才 许 芸 王文娟 编著
沈良忠 伍汝祺



 科学出版社
www.sciencep.com

新世纪高等学校计算机专业教材系列

数据库原理及其应用教程

——学习指导、例题分析、习题解答 与标准试题库

黄德才 许 芸 王文娟 编 著
沈良忠 伍汝祺

科 学 出 版 社

北 京

内 容 简 介

本书为配合《数据库原理及其应用教程》(黄德才主编,科学出版社,第二版)一书的教学而编写,通过每章的主要知识点介绍、典型例题分析、教程习题解答和标准试题的练习,帮助读者充分掌握数据库原理及其应用方面的主要概念、原理和方法。

本书的结构与教材一一对应,共分为8章。在每一章中都给出了该章的学习要求与重点、主要知识点、典型例题分析和教程习题解答,并在此基础上给出了一定数量的标准试题库,包括填空题、选择题、简答题和综合题等,并给出了标准试题库的参考答案,最后附录中还给出了一套模拟试题及参考答案。

本书习题覆盖面广、重点内容的覆盖密度大,对教程中的某些内容,还从不同角度进行了讨论,不仅可作为高等院校计算机专业、软件工程专业、管理信息系统专业“数据库原理及其应用”课程的教学参考书,也可作为成人教育、自学考试学员或相关专业人士学习数据库知识和复习应考的参考书。

图书在版编目(CIP)数据

数据库原理及其应用教程:学习指导、例题分析、习题解答与标准试题库/
黄德才等编著. —北京:科学出版社,2006

(新世纪高等学校计算机专业教材系列)

ISBN 7-03-017116-0

I. 数… II. 黄… III. 数据库系统-高等学校-教学参考资料 IV. TP311.13

中国版本图书馆 CIP 数据核字(2006)第 031458 号

责任编辑:段博原 贾瑞娜 / 责任校对:刘小梅

责任印制:张克忠 / 封面设计:陈 敬

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2006年7月第 一 版 开本:16(787×1092)

2006年7月第一次印刷 印张:14 1/2

印数:1—4 000

字数:330 000

定价:18.00 元

(如有印装质量问题,我社负责调换〈路通〉)

前 言

本书为配合《数据库原理及其应用教程》(黄德才主编,科学出版社,第二版)一书的教学而编写,通过主要知识点概括、典型例题分析、教程习题解答和标准试题的练习,帮助读者充分理解数据库原理及其应用方面的基本概念、基本原理和应用方法,同时使读者能够掌握本门课程中主要问题的求解思路与方法,深化对基本概念的理解,提高分析问题和解决问题的能力。

本书的结构与教程一一对应,共分为8章:第1章关系数据库标准语言——SQL;第2章数据管理与数据库;第3章关系数据库模型;第4章关系模式的规范化设计理论;第5章数据库的安全与保护;第6章数据库设计与实施;第7章数据库应用系统开发;第8章数据库技术新发展。

每一章都包括学习要求与重点、主要知识点、典型例题分析、教程习题解答和标准试题库(填空题、选择题、简答题和综合题),以及标准试题库参考答案等6部分,最后的附录给出一套模拟试题及参考答案。

“学习要求与重点”旨在帮助读者明确学习各章需要掌握的主要知识及其重点知识。

“主要知识点”是对该章学习内容的总结和归纳,旨在帮助读者理清学习思路和线索,便于读者复习、巩固并掌握教程中学过的知识。

“典型例题分析”围绕每章学习的要点、重点和难点知识问题给出求解思路和方法,旨在帮助读者提高分析和解决问题的能力。

“教程习题解答”是对教程中课后习题的详细解答,旨在帮助读者全面理解和掌握所学的知识。

“标准试题库”为读者提供选择题、填空题、简答题和综合题4种题型,对选择题、填空题、简答题直接给出答案,对综合题还给出了分析和解题的完整过程。这不仅可以帮助读者全面巩固所学知识,还可帮助读者提高知识的灵活运用能力。

“标准试题库参考答案”帮助读者进一步检验自己的学习效果。

本书习题覆盖面广、重点内容的覆盖密度大。既收集了较容易的题目,也收集了难度适中和一些较难的题目,对教程中的某些内容,还从不同角度进行了分析讨论,可作为高等院校计算机专业、软件工程专业、管理信息系统专业“数据库原理及其应用”课程的教学参考书,也可作为成人教育,自学考试学员或相关专业人士学习数据库知识和复习应考的参考书。

本书由黄德才组织编写,草拟提纲、规划各章节基本内容,并负责各章学习要求与重点、主要知识点、标准试题库及其参考答案的编写工作。各章典型例题分析和教程习题解答部分编写的分工是:沈良忠负责第2章、第7章和第8章,伍汝祺负责第1章,

许芸负责第3章、第4章，王文娟负责第5章和第6章。最后由黄德才完成全书的修改和统稿工作。

本书习题较多，解答上可能存在不准确或不全面之处，敬请广大读者批评指正。

联系方式：邮箱地址 hdc@zjut.edu.cn。

编 者

2006年2月于杭州

目 录

前言

第 1 章 关系数据库标准语言——SQL	1
1.1 学习要求与重点	1
1.2 主要知识点	1
1.3 典型例题分析	8
1.4 教程习题解答	12
1.5 标准试题库	19
1.6 标准试题库参考答案	24
第 2 章 数据管理与数据库	29
2.1 学习要求与重点	29
2.2 主要知识点	29
2.3 典型例题分析	35
2.4 教程习题答案	39
2.5 标准试题库	44
2.6 标准试题库参考答案	48
第 3 章 关系数据库模型	51
3.1 学习要求与重点	51
3.2 主要知识点	51
3.3 典型例题分析	59
3.4 教程习题解答	65
3.5 标准试题库	71
3.6 标准试题库参考答案	77
第 4 章 关系模式的规范化设计理论	83
4.1 学习要求与重点	83
4.2 主要知识点	83
4.3 典型例题分析	91
4.4 教材习题解答	96
4.5 标准试题库	106
4.6 标准试题库参考答案	112
第 5 章 数据库的安全与保护	119
5.1 学习要求与重点	119
5.2 主要知识点	119
5.3 典型例题分析	130

5.4	教程习题解答	133
5.5	标准试题库	137
5.6	标准试题库参考答案	140
第6章	数据库设计与实施	144
6.1	学习要求与重点	144
6.2	主要知识点	144
6.3	典型例题分析	153
6.4	教程习题解答	161
6.5	标准试题库	166
6.6	标准试题库参考答案	171
第7章	数据库应用系统开发	176
7.1	学习要求与重点	176
7.2	主要知识点	176
7.3	典型例题分析	183
7.4	教程习题解答	187
7.5	标准试题库	188
7.6	标准试题库参考答案	192
第8章	数据库技术新发展	195
8.1	学习要求与重点	195
8.2	主要知识点	195
8.3	典型例题分析	205
8.4	教程习题解答	208
8.5	标准试题库	212
8.6	标准试题库参考答案	215
参考文献		217
附录 A	数据库原理及其应用教程样卷	218
附录 B	数据库原理及其应用教程样卷参考答案	222

第 1 章 关系数据库标准语言——SQL

1.1 学习要求与重点

1.1.1 学习要求

本章要求学生全面理解关系数据库的标准语言——SQL 语言的发展历程、SQL 语言的特点；掌握 SQL 的数据定义、数据更新、数据查询和数据控制命令，并能熟练地应用 SQL 语言完成对关系数据库进行定义、更新、查询和定义视图的操作，了解嵌入式 SQL 的有关概念。

1.1.2 学习重点

本章学习的重点是 SQL 语言的数据定义、数据更新、数据查询和数据控制命令，难点是嵌入式 SQL 的有关概念。

1.2 主要知识点

1.2.1 SQL 概述

SQL 语言主要有以下特点：

- 1) 综合统一。SQL 语言将数据定义语言 DDL、数据操纵语言 DML、数据控制语言 DCL 的功能集于一体，语言风格统一，可以独立完成数据库生命周期中的全部活动。
- 2) 高度非过程化。用 SQL 语言进行数据操作只要提出“做什么”，而无需指明“怎么做”。
- 3) 面向集合操作。SQL 语言操作的对象和操作的结果都是集合——关系。
- 4) 一种语法，两种使用方式。SQL 语言既是自含式语言，又是嵌入式语言。
- 5) 语言简捷，易学易用。完成核心功能只用 9 个动词，SQL 语言接近英语句子。
- 6) 支持三级模式结构。外模式对应于视图（view），模式对应于基本表，内模式对应于存储文件。

1. 基本表

数据库中独立存在的表。每个基本表对应一个关系模式和关系。基本表由 SQL 语言的 CREATE TABLE 命令创建。

2. 视图（虚拟表）

由一个或几个基本表导出的虚拟表，由 SQL 语言的 CREATE VIEW 命令创建。

视图就是用户模式（外模式），它存放着视图的定义及其关联的基本表名称等信息，而不存放视图对应的具体数据。用户可以在视图上再定义视图。

数据库使用视图机制的主要优点：

- 1) 简化用户的操作。
- 2) 多角度看待同一数据。
- 3) 视图为数据库重构提供了一定程度的逻辑独立性。
- 4) 对数据安全保护。

3. 存储文件

若干个基本表组成一个存储文件。存储文件的逻辑结构组成了关系数据库的内模式。

1.2.2 SQL 的数据定义

1. 定义基本表

SQL 语言使用 CREATE TABLE 语句定义基本表，其一般格式为

```
CREATE TABLE <表名> (<列名> <数据类型> [列级完整性约束条件]
                        [, <列名> <数据类型> [列级完整性约束条件]]...
                        [, <表级完整性约束条件>]);
```

2. 修改基本表

SQL 语言用 ALTER TABLE 语句修改基本表，其一般格式为

```
ALTER TABLE <表名>
[ADD <新列名> <数据类型> [完整性约束]]
[DROP [完整性约束名]]
[MODIFY <列名> <数据类型>];
```

3. 删除基本表

用 DROP TABLE 语句可删除基本表。其一般格式为

```
DROP TABLE <表名>;
```

4. 向表中添加元组

向表中添加单个元组使用 INSERT 命令，其语句格式为

```
INSERT
INTO <表名> [( <属性列 1> [, <属性列 2> ... ] ) ]
VALUES ( <常量 1> [, <常量 2> ] ... );
```

5. 建立索引

使用 CREATE INDEX 命令，其语句格式为

```
CREATE [UNIQUE] [CLUSTERED] INDEX<索引名>
ON <表名> (<列名> [<排序方式>] [, <列名> [<排序方式>]]...);
```

6. 删除索引

使用 DROP INDEX 语句，其一般格式为

```
DROP INDEX <表名. 索引名>;
```

1.2.3 SQL 的数据查询

1. SELECT 语句的一般格式

```
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]];
```

2. 无条件查询

在 SELECT 语句中不使用 [WHERE <条件表达式>] 子句。

3. 条件查询

在 SELECT 语句中使用 [WHERE <条件表达式>] 子句。常见的条件有：

1) 比较条件。例如：

```
WHERE Sdept = '数学';
WHERE Sage >= 18 AND Sage <= 22;
```

2) 谓词条件。例如：

```
WHERE Sage BETWEEN 18 AND 22;
WHERE Sage NOT BETWEEN 18 AND 22;
WHERE Sdept IN ('自动化', '数学', '计算机');
```

还有用 LIKE 或 NOT LIKE 谓词、通配符 % 和 _ 构成的谓词条件，如：

```
WHERE Sname LIKE '刘 %';
```

4. 查询结果排序

使用 ORDER BY 子句以及 DESC 或 ASC 命令关键字。

5. 使用集函数

在 SELECT 语句中可以使用 COUNT ()、AVG () 等集函数。

6. 查询结果分组

在 SELECT 语句中使用 GROUP BY 子句将查询结果表按照某一列或多列的值分组,使其列值相等的为一组。还可以用 HAVING 短语指定结果组满足的条件。

7. 不同表之间的连接查询

一般是在 WHERE 子句中将两(多)个表的属性列名,作为连接条件,其格式通常为

[<表名 1>.]<列名 1> <比较运算符> [<表名 2>.]<列名 2>

或者

[<表名 1>.]<列名 1> BETWEEN [<表名 2>.]<列名 2> AND [<表名 2>.]<列名 3>

8. 自身连接

将同一个表与自己进行连接,其基本方法是在 SELECT 语句的 FROM 子句中将同一个表取两个不同的别名。比如

```
SELECT A. cno,A. cname,B. Pre_Cno
FROM Courses A, Courses B
WHERE A. Pre_Cno = B. Cno;
```

9. 外连接

选定连接操作中涉及的某个表 A 为基准,即使另一个表 B 中没有与之匹配的记录,其结果表中也要求包括表 A 的所有元组。若表 B 中没有与之匹配的记录,结果表中涉及表 B 的属性列全部取空值。

10. 嵌套查询

将一个查询块(SELECT-FROM-WHERE)嵌套在另一个查询块的 WHERE 子句或 HAVING 短语条件中的查询。

11. 常用的嵌套查询

- 1) 带谓词 IN 的嵌套查询。
- 2) 带有比较运算符的嵌套查询。
- 3) 带谓词 ANY 或 ALL 的嵌套查询。
- 4) 带谓词 EXISTS 的嵌套查询。

12. 并操作查询

标准 SQL 提供了并操作运算命令 UNION,将多个 SELECT 语句的结果进行传统

的集合并操作。但这个操作要求参加 UNION 操作的各个结果表的列数必须相同且对应属性的数据类型也相同。

13. 其他

标准 SQL 中没有直接提供集合的交和差的操作，但可用其他条件查询来实现。

1.2.4 SQL 的数据更新

1. 插入数据

将一个 SELECT 查询语句的结果集插入到表中的 INSERT 命令格式为

```
INSERT  
INTO <表名>[(<属性列 1>[,<属性列 2>...])]  
SELECT 子查询语句;
```

2. 修改数据

修改 (UPDATE) 操作语句的一般格式为

```
UPDATE <表名>  
SET <列名 1> = <表达式 1>[,<列名 2> = <表达式 2>...]  
[WHERE <条件>];
```

3. 删除数据

删除 (DELETE) 语句的一般格式为

```
DELETE  
FROM <表名>  
[WHERE <条件>];
```

说明：

1) DELETE 语句的功能是从指定表中删除满足 WHERE 子句条件的所有元组。如果省略 WHERE 子句，表示删除表中全部元组，但表的模式定义仍在字典中，即 DELETE 语句使表成为空表。

2) 增删改操作每次只能对一个表进行，且必须注意关系之间的参照完整性和操作顺序，否则就会操作失败甚至造成数据库的不一致。因此，执行删除操作时，必须将有关参照表的相关元组一起删除。一般使用以下的两种策略：

①系统自动地删除参照表中相应的元组；②系统检查参照表中是否存在相应的元组，如果存在，则操作失败。

1.2.5 SQL 的视图

1. 定义视图

SQL 语言用 CREATE VIEW 命令建立视图，其一般格式为

```
CREATE VIEW <视图名> [(<列名>[,<列名>]...)]
```

```
AS <SELECT 子查询语句>
[WITH CHECK OPTION];
```

2. 删除视图

删除视图 (DROP VIEW) 的语句格式为

```
DROP VIEW <视图名>;
```

3. 查询视图

对于已经定义的视图，用户应用 SELECT 语句，就像查询基本表一样查询视图。

4. 更新视图

1) 更新视图的含义。指通过视图来插入 (INSERT)、删除 (DELETE) 和修改 (UPDATE) 数据。由于视图是不实际存储数据的虚拟表，因此对视图的更新，系统将自动转换为对基本表的更新。

2) 更新视图的限制。由于有些视图的更新操作不能唯一的有意义地转换成对相应基本表的更新操作，因此，用视图完成数据的更新 (增加、删除、修改) 操作有以下限制规则：

①如果一个视图是从多个表通过连接操作导出的，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，使用了分组和集函数，也不允许对这个视图执行更新操作；③如果一个视图是从单个基本表使用选择和投影操作导出的，且视图的属性集包含了基本表的一个候选键，这种视图称为“行列子集视图”，允许对这个视图执行更新操作。

1.2.6 SQL 的数据控制

1. 授权

向用户授予操作权限的 GRANT 语句的一般格式为

```
GRANT <权限>[, <权限>]...
[ON <对象类型><对象名>]
TO <用户>[, <用户>]...
[WITH GRANT OPTION];
```

2. 不同对象类型允许的操作权限 (表 1.1)

表 1.1 不同对象类型允许的权限操作

对 象	对象类型	操 作 权 限
属性列视图	TABLE	SELECT, INSERT, UPDATE, DELETE, ALL PRIVILEGES
基本表	TABLE	SELECT, INSERT, UPDATE, DELETE, ALTER, INDEX, ALL PRIVILEGES
数据库	DATABASE	CREATETAB

3. 收回权限

DBA 或其授权者可以用 REVOKE 语句收回向用户授予的权限，其语句格式为

```
REVOKE<权限>[,<权限>]...  
[ON<对象类型><对象名>]  
FROM<用户>[,<用户>]...;
```

1.2.7 嵌入式 SQL

1. 嵌入式 SQL

能够嵌入到高级语言中进行混合编程的 SQL 语言叫嵌入式 SQL，而被 SQL 嵌入的高级语言叫宿主语言或主语言。

2. DBMS 处理嵌入式 SQL 的方法

- 1) 预编译的方法。
- 2) 修改和扩充主语言使之能处理 SQL。

3. 主语言如何区分嵌入式 SQL

区分主语言语句与 SQL 语句的标志一般随主语言的不同而不同。通常有：

- 1) 在嵌入的 SQL 语句前加前缀 EXEC SQL。
- 2) 在嵌入的 SQL 语句后面加结束标志 END-EXEC 或者分号。
- 3) 在嵌入的 SQL 语句分别加前缀和结束标志。
- 4) 把 SQL 扩充为主语言的一部分，这时在使用上同交互式 SQL 就没有区别了。

4. 嵌入式 SQL 语句与主语言之间的通信

1) 嵌入式 SQL 与高级语言的分工。SQL 语句负责操纵数据库，高级语言语句负责控制程序流程和处理数据。

2) 嵌入式 SQL 与主语言之间必须互相通信才能协调工作。

3) 数据库工作单元与源程序工作单元之间的通信主要包括：

①向主语言传递 SQL 语句的执行状态信息，使主语言能够据此信息控制程序流程，主要通过定义 SQL 通信区（SQL communication area, SQLCA）实现；②主语言向 SQL 语句提供参数，主要通过主语言中声明的变量——主变量（host variable）来实现；③将 SQL 语句查询数据库的结果数据交给主语言进行进一步地处理，这主要通过主变量和游标（cursor）实现。

5. SQL 通信区——SQLCA

SQL 语句执行后，DBMS 要反馈给应用程序若干信息，主要包括描述系统当前工作状态和运行环境的各种数据。这些信息将送到 SQL 通信区 SQLCA 中。应用程序从

SQLCA 中取出这些状态信息, 据此决定接下来执行的语句。

SQLCA 是一个数据结构, 其定义方法随主语言的不同而不同。有的应用程序中用 EXEC SQL INCLUDE SQLCA 加以定义。SQLCA 中有一个存放每次 SQL 语句执行情况 (成功与否) 的变量 SQLCODE。应用程序每执行完一条 SQL 语句之后都应该测试一下 SQLCODE 的值, 以了解该 SQL 语句执行情况并做相应处理。如果 SQLCODE 等于预定义的常量 SUCCESS (有的规定为 0), 则表示 SQL 语句成功, 否则在 SQLCODE 中存放表示错误类型的代码。

6. 游标

游标是计算机系统为用户开设的一个数据缓冲区, 存放 SQL 语句的结果数据集, 每个游标区都有一个名字。用户可以通过游标逐一读取数据记录, 然后赋值给主变量, 再交给主语言程序作进一步地处理。

1.3 典型例题分析

例题 1 仓库管理关系模型中有如下 5 个关系模式:

零件: PART (P#, Pname, Color, Weight)。

项目: PROJECT (J#, Jname, Pdate)。

供应商: SUPPLIER (S#, Sname, Saddr)。

供应: P_P (J#, P#, Total)。

采购: P_S (P#, S#, Quantity)。

1) 试用 SQL 的 DDL 语句定义上述 5 个基本表, 并说明主键和外键。

2) 试将 PROJECT、P_P、PART 这 3 个基本表的自然连接定义为一个视图 VIEW1, PART, P_S, SUPPLIER 这 3 个基本表的自然连接定义为一个视图 VIEW2。

3) 试在上述两个视图的基础上进行数据查询: ①检索上海的供应商所供应的零件的编号和名字; ②检索项目 J4 所用零件的供应商的编号和名字。

解 本题考核用 SQL 语言定义基本表、视图和表示查询需求等命令及其有关概念的理解和应用能力, 有关概念和方法在教材的第 1 章中介绍。

1) CREATE TABLE PART

(P# CHAR(6),

Pname CHAR(10) NOT NULL,

Color CHAR(6),

Weight FLOAT(5),

PRIMARY KEY(P#));

CREATE TABLE PROJECT

(J# CHAR(6),

Jname CHAR(12) NOT NULL,

```
Pdate DATE,  
PRIMARY KEY(J#));
```

```
CREATE TABLE SUPPLIER  
(S# CHAR(8),  
Sname CHAR(12) NOT NULL,  
Saddr VARCHAR(30),  
PRIMARY KEY(S#));
```

```
CREATE TABLE P_P  
(J# CHAR(6),  
P# CHAR(6),  
Total INTEGER,  
PRIMARY KEY(J#,P#),  
FOREIGN KEY(J#) REFERENCES PROJECT(J#),  
FOREIGN KEY(P#) REFERENCES PART(P#));
```

```
CREATE TABLE P_S  
(P# CHAR(6),  
S# CHAR(8),  
Quantity INTEGER,  
PRIMARY KEY(P#,S#),  
FOREIGN KEY(P#) REFERENCES PART(P#),  
FOREIGN KEY(S#) REFERENCES SUPPLIER(S#));
```

2) CREATE VIEW VIEW1

```
AS SELECT PROJECT. *, P_P.P#, P_P.Total, PART.Pname,  
PART.Color, PART.Weight  
FROM PROJECT, P_P, PART  
WHERE PROJECT.J# = P_P.J# AND P_P.P# = PART.P#;
```

CREATE VIEW VIEW2

```
AS SELECT PART. *, P_S.Quantity,  
SUPPLIER.Sname, SUPPLIER.Saddr  
FROM PART, P_S, SUPPLIER  
WHERE PART.P# = P_S.P# AND P_S.S# = SUPPLIER.S#;
```

3) ① SELECT P#, Pname

```
FROM VIEW2
```



```
WHERE Saddr = '上海';  
② SELECT S#, Sname  
FROM VIEW2  
WHERE P# IN  
(SELECT P#  
FROM VIEW1  
WHERE J# = 'J4');
```

例题 2 在宿主语言的程序中使用标准 SQL 语句有哪些规定？

解 本题主要考核对在宿主语言程序中使用 SQL 语句的有关规定的理解，同时了解这些规定在具体程序语言中的变化。有关内容详见教材 1.7 节。

有三条规定：

1) 在程序中要区分 SQL 语句与宿主语言语句，所有 SQL 语句必须加前缀标识“EXECSQL”，以及语句结束标志“END _ EXEC”。

2) 允许嵌入的 SQL 语句引用宿主语言的程序变量（共享变量），而主语句不能引用数据库中的字段变量。

3) SQL 的集合处理方式与宿主语言的单记录处理方式用游标机制协调。

说明 随着面向对象程序技术的应用和发展，在有些宿主语言的程序中使用标准 SQL 语句的方式，已经有了很大的变化。比如，在 PowerBuilder 中使用 SQL 语句时，只要在末尾加分号即可，而在 Delphi 中使用 SQL 语句时，则必须先申明（定义）一个 TQuery 对象，然后在 TQuery 对象中指定所使用的 SQL 语句。

例题 3 SQL 的集合处理方式与宿主语言的单记录处理方式之间如何协调？

解 本题主要考核对 SQL 的集合处理方式与宿主语言的单记录处理方式的理解决，以及游标的概念和使用方法。有关内容详见教材 1.7 节。

由于 SQL 语句处理的是记录集合，而宿主语言语句一次只能处理一条记录，因此通过游标机制来协调，并将集合操作转换成单记录处理方式。

游标是系统为用户开设的一个数据缓冲区，存放 SQL 语句的执行结果（元组的集合），每个游标区都有一个名字（相当于一个文件）。用户可以通过访问游标逐一获取元组（数据记录），并将其赋给主变量，交由主语言作进一步处理。

与游标有关的 SQL 语句有 4 个：游标定义、游标打开、游标推进和游标关闭等。

例题 4 嵌入式 SQL 的 DML 语句何时不必涉及游标？何时必须涉及游标？

解 本题主要考核对在宿主语言中如何使用游标机制的理解。有关内容详见教材 1.7 节。

1) 不涉及游标的 DML 语句有下面几种情况：

① INSERT、DELETE、UPDATE 语句，只要加上前缀标识和结束标识，就能嵌入在宿主语言程序中使用；② 对于 SELECT 语句，如果已知查询结果肯定是单个元组，也可直接嵌入在主程序中使用。

2) 涉及游标的 DML 语句有下面几种情况：