高级SQL

练习练习

5.1描述您选择使用嵌入式SQL而不是单独使用SQL或仅使用通用编程语言的情况。

答：在SQL中编写查询通常比在通用编程语言中编写相同查询容易得多。但是，并非所有类型的查询都可以用SQL编写。此外，无法从SQL内部执行非详细操作，例如打印报告，与用户交互或将查询结果发送到图形用户界面。在我们想要两全其美的情况下，我们可以选择嵌入式SQL或动态SQL，而不是单独使用SQL或仅使用通用编程语言。

嵌入式SQL具有程序不那么复杂的优点，因为它避免了ODBC或JDBC函数调用的混乱，但需要专门的预处理器。

5.2使用JDBC元数据功能编写Java函数，该功能将Result-Set作为输入参数，并以表格形式打印出结果，并使用适当的名称作为列标题。

回答：

public class ResultSetTable实现TabelModel {

ResultSet结果; ResultSetMetaData元数据; int num cols;

ResultSetTable（ResultSet result）抛出SQLException {

this.result =结果;

metadata = result.getMetaData（）;

num cols = metadata.getColumnCount（）;

for（int i = 1; i <= num cols; i ++）{

System.out.print（metadata.getColumnName（i）+''''）;

1

2第5章高级SQL

}

的System.out.println（）;

while（result.next（））{

for（int i = 1; i <= num cols; i ++）{

System.out.print（result.getString（metadata.getColumnName（i）+''''））;

}

的System.out.println（）;

}

}

}

5.3使用JDBC元数据功能编写Java函数，该功能打印数据库中所有关系的列表，为每个关系显示其属性的名称和类型。

回答：

DatabaseMetaData dbmd = conn.getMetaData（）; ResultSet rs = dbmd.getTables（）;

while（rs.next（））{

System.out.println（rs.getString（''TABLE NAME''）; ResultSet rs1 = dbmd.getColumns（null，''schema-name''，

rs.getString（''TABLE NAME''），''％''）;

while（rs1.next（））{

System.out.println（rs1.getString（''COLUMN NAME''），rs.getString（''TYPE NAME''）;

}

}

5.4展示如何强制执行约束“教师不能在一个学期中在两个不同的教室中教授同一时间段。”使用触发器（请记住，教学关系的变化以及部分关系可以违反约束）。

答：填写

5.5写入触发器以在段到时间段，对段的更新和时隙中强制执行参照完整性约束。请注意，我们在图5.8中编写的内容不包括更新操作。

答：填写

5.6要维护学生关系的完整信用属性，请执行以下操作：

一个。修改take更新的触发器，以处理可能影响tot cred值的所有更新。

湾编写一个触发器来处理take关系的插入。

练习3

C。在什么假设下，合理的是不在课程关系上创建触发器？

答：填写

5.7考虑图5.25的银行数据库。让我们定义一个视图分支cust

如下：

创建视图分支cust as

选择分支名称，客户名称

来自存款人，账户

其中depositor.account number = account.account number

假设视图已实现;也就是说，计算并存储视图。编写触发器以维护视图，即在存款人或帐户的插入和删除时使其保持最新。不要打扰更新。

答：要插入物化视图分支，我们必须在插入存储器和帐户中设置数据库触发器。我们假设数据库系统使用立即绑定来执行规则。此外，假设关系的当前版本由关系名称本身表示，而新插入的元组集合通过使用前缀 - 插入来限定关系名称来表示。

此插入的有效规则如下 -

通过存款人定义触发器插入分支机构

插入存款人后

引用为每个语句插入插入分支cust的新表

选择分支名称，客户名称

来自插入，帐户

其中inserted.account number = account.account number

通过帐户定义触发器插入分支机构

在帐户上插入后

引用为每个语句插入插入分支cust的新表

选择分支名称，客户名称

来自存款人，已插入

其中depositor.account number = inserted.account number

请注意，如果执行绑定被延迟（而不是立即执行），则两个活动规则将插入一组新元组帐户与存储器新元组的连接结果，从而导致重复分支机构中相应的元组。

从分支cust删除元组类似于插入，除了从存储者或帐户的删除将导致这些关系的自然连接具有较少数量的元组。我们表示新的

4第5章高级SQL

通过使用关键字限定关系名称来删除元组集

删除。

定义通过存款人从分支机构中删除触发器

在存款人删除后

引用旧表作为已从分支cust删除的每个语句删除

选择分支名称，客户名称

来自删除，帐户

其中deleted.account number = account.account number

定义通过帐户从分支机构中删除触发器

在帐户上删除后

引用旧表作为已从分支cust删除的每个语句删除

选择分支名称，客户名称

来自存款人，已删除

其中depositor.account number = deleted.account number

5.8考虑图5.25的银行数据库。编写SQL触发器以执行以下操作：在删除帐户时，对于帐户的每个所有者，检查所有者是否有任何剩余帐户，如果没有，则将其从存款人关系中删除。

回答：

在帐户删除后创建触发器check-delete-trigger

将旧行引用为orow

对于每一行

从存款人中删除

其中depositor.customer名称不在

（从存款人处选择客户名称

账号<> orow.account号码）

结束

5.9显示如何使用汇总按立方体（a，b，c，d）表示组;你的答案应该只有一个group by子句。

回答：

groupby汇总（a），汇总（b），汇总（c），汇总（d）

5.10给定关系S（学生，科目，标记），写一个查询以找到顶部

n学生按总分数，按排名。

答：我们假设多名学生没有相同的分数，否则问题不是确定性的;下面的查询确定返回所有与n学生具有相同分数的学生，因此它可能会返回超过n个学生。

练习5

选择学生，总和（商标）为总数，

rank（）over（order by（total）desc）as trank

来自S.

小组学生

trank≤n

5.11考虑5.6节中的销售关系。 编写一个SQL查询来计算关系上的多维数据集操作，给出图5.21中的关系。 不要使用立方体构造。

回答：

（从销售中选择颜色，尺寸，总和（数量）

groupby颜色，大小

）

联盟

（从销售中选择颜色，'全部'，总和（数字）

groupby颜色

）

联盟

（从销售中选择“全部”，大小，总和（数字）

groupby大小

）

联盟

（从销售中选择“全部”，大小，总和（数字）

groupby大小

）

联盟

（选择'全部'，'全部'，总和（数字）

来自销售

）