



TT 数据库

SQL SELECT 语句基础指南

SQL SELECT 语句基础指南

本技术专题节选自《常见 SQL 查询：SQL 数据操作指南第二版》的 SQL SELECT 语句基础。我们将学习如何使用 SELECT 来访问 SQL 数据库中所有内容，以及组成 SELECT 语句的许多子句名称和功能。同时作者 John L. Viescas 和 Michael J. Hernandez 还阐述了如何使用 DISTINCT 关键字消除重复的行，以及如何正确使用 ORDER BY 子句来排序数据。

关于 SELECT

与其它主键相比，SELECT 的确是 SQL 中最核心的关键字。它是 SQL 语言中最强大和复杂的语句和从数据库表中检索信息的方法的基础。通过结合其它关键字和子句，SELECT 可以实现无数种查找和查看信息的方法。几乎所有关于谁、什么、何处、何时、或者甚至是如果和多少等问题都可以使用 SELECT 来答复。

- ❖ [如何在 SQL 中使用 SELECT 语句（上）](#)
- ❖ [如何在 SQL 中使用 SELECT 语句（中）](#)
- ❖ [如何在 SQL 中使用 SELECT 语句（下）](#)

将信息请求翻译成 SQL 语句

如何将信息请求翻译为 SQL SELECT 语句？当我们向数据库请求信息时，通常是以问题的形式或者一个隐含问题的语句形式。

- ❖ [如何将信息请求翻译为 SQL SELECT 语句（一）](#)
- ❖ [如何将信息请求翻译为 SQL SELECT 语句（二）](#)
- ❖ [如何将信息请求翻译为 SQL SELECT 语句（三）](#)
- ❖ [如何将信息请求翻译为 SQL SELECT 语句（四）](#)

使用 DISTINCT 消除重复行

当运行 SELECT 语句时，我们将不可避免地会碰上有重复记录行的结果集。SQL 对于这样的结果集的出现不会给出任务的警告。在 SELECT 语句中的使用 DISTINCT 关键字，结果集就会去除和清理所有重复的记录行。

- ❖ 在 SQL 中使用 DISTINCT 消除重复行（上）
- ❖ 在 SQL 中使用 DISTINCT 消除重复行（下）

使用 SELECT 查询的 ORDER BY 子句

在本章开头，我们就谈到 SELECT 操作可以分成三个较小的操作：SELECT 语句、SELECT 表达式和 SELECT 查询。同时，我们还谈到可以用不同的方式组合这些操作来实现复杂的请求。然而，我们也需要组合这些操作来实现结果集中记录行的排序。

- ❖ 在 SQL 中使用 SELECT 查询的 ORDER BY 子句（上）
- ❖ 在 SQL 中使用 SELECT 查询的 ORDER BY 子句（中）
- ❖ 在 SQL 中使用 SELECT 查询的 ORDER BY 子句（下）

SQL SELECT 语句和 SELECT 查询示例

目前我们已经探讨了 SELECT 语句和 SELECT 查询的基本特性，接下来让我们来看看在不同的环境中如何应用这些例子。这些示例各自包含了一个示例数据库，同时还阐述了 SELECT 语句、SELECT 查询的使用，以及两种用于为翻译语句创建字段的辅助技术。我们还涉及到这些操作将返回的示例结果集，并且这些结果集都会被直接附在 SQL 语句之后。结果集之前的名称有双重意义：它标识结果集，同时它也是我们在示例中指派给 SQL 语句的名称。

-
- ❖ SQL SELECT 语句和 SELECT 查询示例（一）
 - ❖ SQL SELECT 语句和 SELECT 查询示例（二）
 - ❖ SQL SELECT 语句和 SELECT 查询示例（三）
 - ❖ SQL SELECT 语句和 SELECT 查询示例（四）

如何在 SQL 中使用 SELECT 语句（上）

SELECT 介绍

与其它主键相比，SELECT 的确是 SQL 中最核心的关键字。它是 SQL 语言中最强大和复杂的语句和从数据库表中检索信息的方法的基础。通过结合其它关键字和子句，SELECT 可以实现无数种查找和查看信息的方法。几乎所有关于谁、什么、何处、何时、或者甚至是如何和多少等问题都可以使用 SELECT 来答复。只要我们已经正确设计好了数据库并收集了恰当的数据，那么我们就可以获得我们需要的答案并为组织作出正确的决定。当学习到第 5 部分“修改数据集”时，我们就会发现，我们将应用很多所学的 SELECT 技巧来创建 UPDATE、INSERT 和 DELETE 语句。

在 SQL 中 SELECT 操作可以分成三个较小的操作，即 SELECT 语句、SELECT 表达和 SELECT 查询。（将 SELECT 操作按照这种方式划分是为了更容易地理解它和领会它的复杂性。）这些操作中的任何一个都自身有一套关键字和子句，从而使我们能够灵活地创建最终的 SQL 语句，而这些语句正是用于解决我们的数据库操作问题的。在下面的章节中，我们甚至可以以各种不同方式来组合这些操作来处理非常复杂的问题。

在本章中，我们将从探讨 SELECT 语句开始，并简单讨论 SELECT 查询。接下来，在第 5 章“深入字段”和第 6 章“过滤数据”中，我们会更详细地探讨 SELECT 语句。

注意：在其它关于关系数据库的书籍中，有时我们也会看到用关系（relation）代替表（table），并且还可能碰上用元（tuple）或记录（record）代替行（row），以及或者是用属性（attribute）或域（field）代替字段（column）。但是，SQL Standard 特定使用的词汇是表（table）、行（row）和字段（column）来表示这些特定的数据库结构元素。我们会一直使用 SQL Standard 的术语，并且在这本书接下来的内容中继续沿用后面这三个词。

SELECT 语句

SELECT 语句组成我们发布到数据库上的每个查询的基础。当我们创建和执行一个 SELECT 语句时，我们就是在查询数据库了。（听起来这是很明显的，但是我们想保证所有读者都是在同一起点上开始的。）事实上，RDBMS 程序允许我们将 SELECT 语句保存为查询（query）、视图（view）、方法（function）或者存储过程（stored procedure）。任何时候，当有人说她要查询数据库时，我们就知道她是要执行某种 SELECT 语句。根据 RDBMS 程序，SELECT 语句可以在命令行窗口、Example (QBE) 网格的交互查询、或者在程序代码中直接执行。不管我们选择如何定义和执行它，SELECT 语句的语法总是相同的。

注意：很多数据库系统都对 SQL Standard 提供扩展来允许我们在方法和存储过程中创建复杂的程序语句（如 IF、Then、Else），但是每个不同的产品的具体语法是唯一的。本书中的内容并不涉及这些程序语言——如 Microsoft SQL Server's Transact-SQL 或者 Oracle's PL/SQL。当我们为特定的数据库系统创建方法和存储过程时，我们仍然以 SELECT 语句为基础。在本书中，我们将使用 视图（view）来指存储的 SQL 语句，即使我们可能将 SQL 语句嵌入方法或程序中。

(作者: SearchSQLServer.com 译者: 曾少宁/陈柳 来源: TT 中国)

如何在 SQL 中使用 SELECT 语句（中）

一个 SELECT 语句是由几个不同的关键字构成的，它们被称为子句（clauses）。我们可以通过使用这些子句的各种搭配来定义 SELECT 语句来检索需要的信息。有些子句是必须的，而有些则是可选的。另外，每个子句都有一个或多个关键字表示必须的或可选的值。子句会使用这些值来检索 SELECT 语句所需要的信息。图 4-1（第 73 页）显示了一个 SELECT 语句和它的子句的图表。

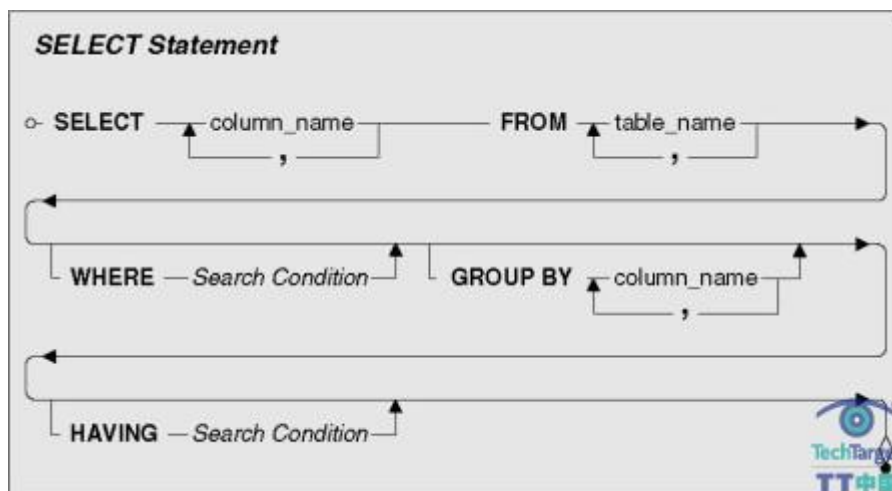


图 4-1SELECT 语句图表

注意：图 4-1 中的语法图显示的是一个基本的 SELECT 语句。随着新的关键字和子句的介绍，我们将会不断地更新和修改这个图。因此，对于在 SQL 语句方面有一定经验的读者，请耐心跟着我们一起往下探讨。

以下是对 SELECT 语句中子句的一个简单总结：

- SELECT——这是 SELECT 语句的主要子句并且是绝对必须的。它是用来指定所查询结果集的字段的。字段本身是在 FROM 子句中指定的表或视图中提取的。（我

们可以同时几个表中提取，我们将在后面的第三部分“操作多个表”中进行探讨。)同时，也可以在这个子句中使用聚集功能，如 Sum(HoursWorked)、或者数学表达式，如 Quantity * Price。

- FROM——这是 SELECT 语句中第二个重要的子句并且也是必要的。我们使用 FROM 子句来指定用于提取 SELECT 子句中所列字段的表或视图。这个子句的使用方式可以更加复杂，这个内容我们将在后面的章节中进行探讨。
- WHERE——这是一个可选的子句，我们可以用来过滤 FROM 子句返回的行。WHERE 关键字后面是一个表达式，技术上称为断言 (predicate)，它用于判断真 (true)、假 (false) 或者未知 (unknown) 值。我们可以使用标准比较操作符、Boolean 操作符或者特定操作符来测试表达式。我们将在第 6 章中探讨所有 WHERE 子句元素。
- GROUP BY——当在 SELECT 子句中的聚合多个方法来产生概括信息时，我们可以 GROUP BY 子句来将信息分成不同的组。数据库系统在 GROUP BY 关键字后面使用任意字段或字段列表作为分组字段。GROUP BY 字段是可选的，并且我们将在第 13 章“分组数据”中进行更详细探讨。
- HAVING——HAVING 子句过滤分组信息中的聚合方法的结果。当 HAVING 关键字后面是判断值为真、假或未知的表达式时，它类似于 WHERE 子句。我们可以使用标准的比较操作符，Boolean 操作符，或者特殊操作符来检测表达式。HAVING 也是一个可选的子句，我们将在第 14 章“过滤分组数据”中进行详细探讨。

首先，我们将探讨一个非常基本的 SELECT 语句，因此我们将关注于 SELECT 或 FROM 子句。随着其它章节的探讨，我们将一个接一个地添加其它子句来建立更复杂的 SELECT 语句。

(作者: SearchSQLServer.com 译者: 陈柳/曾少宁 来源: TT 中国)

如何在 SQL 中使用 SELECT 语句（下）

数据与信息

在我们进行第一个数据库查询之前，必须明确：数据（data）和信息（information）之间有着明显的不同。本质上，数据是存储到数据库上，而信息是从数据库中检索出来的。这种差别是很重要的，因为它有助于我们用正确的眼光来理解它们。记住，数据库是设计来为你的组织中的某人提供有意义的信息的。然而，只有数据库中存储了恰当的数据，并且数据库本身已经以这种方式结构化来支持该信息的情况下，才可以提供信息。让我们对这些内容进行更详细地探讨。

存储在数据库中的值是数据。从本质上看，数据在被手动或自动的过程进行修改之前，它们都是静态的。图 4-2 显示了一些数据示例。

Katherine	Ehrlich	89931	Active	79915
-----------	---------	-------	--------	-------

图 4-2 一个基本数据例子

表面上，这个数据是毫无意义的。比如，没有任何捷径可以让你确定 89931 代表的是什么意义。它是一个邮政代码吗？它是一个零件号码吗？甚至即使你知道它代表一个客户的标识码，那么它与 Katherine Ehrlich 有关吗？只有数据被处理了才可以知道这些问题的答案。在你处理了数据之后，你在使用和查看它时，它才是有意义和有用的，这样数据才成为了信息。信息是动态的，因此它随着数据库上存储的数据变化而不断地改变，同时它也随其自身被以各种方式进行处理和呈现而改变。你可以用 SELECT 语句的结果显示信息，并且以表单的形式在你的电脑屏幕上显示，或者作为报告打印到纸上。但是需要记住的是，你必须以能够以某种方式处理数据才能将它变成有意义的信息。

图 4-3 显示了前面的例子中转换为客户屏幕上的信息的数据。它说明了数据是如何以这种方式操作，然后成为现在对任何看到它的人有意义的内容的。

Customer Information			
Name (F/L):	Katherine Ehrlich	ID #:	89931
Address:	7402 Taxco Avenue	Status:	Active
City:	El Paso	Phone:	555-9284
State:	TX	ZIP:	79915
		Fax:	554-0099

图 4-3 一个关于数据被处理成信息的例子

当运行一个 SELECT 语句时，我们使用它的子句来操作数据，但是语句本身返回的是信息。理解上图了吗？

这里还有我们必须处理的最后一个问题。当我们执行 SELECT 语句时，它通常检索一行或多行信息——确切的行数是根据我们构造语句的方式决定的。这些行总体上被称为结果集 (result set)，我们在本书接下来的内容中也一直这样沿用这个术语。这个名称表达得非常清楚，因为当我们使用一个相关的数据库时，我们总是处理数据的集合。（记住，这个关系模型从某方面说是基于集合论的。）我们可以在结果集上很容易地看到信息，在很多情况下，我们可以修改它的数据。但是，再次强调，它完全是根据我们构造 SELECT 语句的方式。

因此，让我们继续进行探讨，并开始使用 SELECT 语句。

(作者: SearchSQLServer.com 译者: 曾少宁/陈柳 来源: TT 中国)

如何将信息请求翻译为 SQL SELECT 语句（一）

将请求翻译为 SQL

当我们向数据库请求信息时，通常是以问题的形式或者一个隐含问题的语句形式。比如，我们可以这样公式化语句：

```
"Which cities do our customers live in?"  
"Show me a current list of our employees and their phone numbers."  
"What kind of classes do we currently offer?"  
"Give me the names of the folks on our staff and the dates they were hired."
```

在我们明确了问题之后，我们可以将我们的请求翻译成更正式的语句。我们可以使用这种形式来组成语句：

```
Select <item> from the <source>
```

首先查看请求及其替代其中诸如“list”、“show me”、“what”、“which”和“who”以及词“Select”的单词或短语。接着，确定请求中的所有名词，并确定指定的名词是否代表我们想要的一项信息、或者可能存储该项信息的表的名称。如果它是一项信息，那么就用它替代翻译的语句中的<item>。如果它是一个表名称，那么就用它替代<source>。如果对前面列举的第一个问题进行翻译，那么语句看起来是这样的：

```
Select city from the customers table
```

在我们定义了翻译语句之后，我们需要使用如图 4-4 显示的 SQL 句法将它转换成一个完全的 SELECT 语句，然而，这一步就要清理之前翻译的语句。我们可以删除任何不是代表字段或者表的名称、或者不是特定用于 SQL 句法中的名词。在清理过后，翻译语句是这样的：

```
Select city from customers
```

删除了之后，我们现在有了一个完整的 SELECT 语句。

```
SELECT City FROM Customers
```

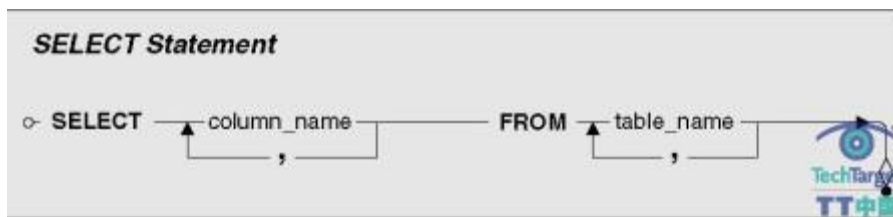


图 4-4 简单 SELECT 语句的语法

我们可以对发送到数据库上的任何请求使用我们刚刚提出的三个步骤的技巧。事实上，我们会在这本书的大多地方都使用了这个技巧。并且，我们建议你在刚开始学习如何创建这些语句时使用它。然而，随着对撰写 SELECT 语句越来越熟悉，你最后将会把这些步骤融会贯通。

在我们定义了翻译语句之后，我们需要使用如图 4-4 显示的 SQL 句法将它转换成一个完全的 SELECT 语句，然而，这一步就要清理之前翻译的语句。我们可以删除任何不是代表字段或者表的名称、或者不是特定用于 SQL 句法中的名词。在清理过后，翻译语句是这样的：

```
Select city from customers
```

删除了之后，我们现在有了一个完整的 SELECT 语句。

```
SELECT City FROM Customers
```

我们可以对发送到数据库上的任何请求使用我们刚刚提出的三个步骤的技巧。事实上，我们会在这本书的大多地方都使用了这个技巧。并且，我们建议你在刚开始学习如何创建这些语句时使用它。然而，随着对撰写 SELECT 语句越来越熟悉，你最后将会把这些步骤融会贯通。

(作者: John L. Viescas and Michael J. Hernandez 译者: 曾少宁/陈柳 来源: TT 中国)

如何将信息请求翻译为 SQL SELECT 语句（二）

记住，当开始学习如何使用 SQL 时，我们大多数情况下都是与字段和表打交道。图 4-4 的语法图在 SELECT 子句中使用 `column_name` 和在 FROM 子句中使用 `table_name` 来反映这个情况。在下一章中，我们将学习如何在这些子句中使用其它项来创建更复杂的 SELECT 语句。

可能你已经注意到了，我们在上面的例子中所使用的要求是相对简单的。重新定义它作为翻译语句并同时确定出在语句中显示的字段名称是很容易的。但是，如果请求并不是这样简单且容易翻译的话，同时确定出 SELECT 子句所需要的字段又很困难时，该怎么处理？最简单的做法就是重新定义请求并让它更具体些。比如，我们可以将这样一个请求：“Show me the information on our clients” 重新改为更清楚的 “List the name, city, and phone number for each of our clients”。如果重新定义要求不能解决问题，那么我们仍然有两个选择。第一，我们可以先确定在 SELECT 语句的 FROM 子句中指定的表是否包含任何有助于使请求更清晰的字段名称，这样我们就可以用它来更容易地定义翻译语句。第二，我们可以更仔细地检查请求并确定一个词或者短语是否包含任何字段名称的暗示（implies）。我们是否可以使用一种或者两种选择是取决于请求本身。只要记住，当感到定义一个翻译语句很困难时，一定要还有着其它行之有效的技巧可利用。让我们来看看各个技巧的例子以及如何在一些典型的场景中应用它。

关于第一个技巧，让我们来尝试翻译下面这个请求。

“I need the names and addresses of all our employees.”

表面上看来这是一个直接的请求。但是，如果再次看一下这个请求，我们会发现一个小的问题：虽然我们可以确定翻译语句所需要的表（Employees），但是在请求中没有任何有助于我们为 SELECT 子句定义所需要的特定字段。虽然词“names”和“addresses”在要

求中出现了，但是它们是通用的语汇。我们可以通过检查请求中所对应的表并确认它是否包含任何可以替代这些项的字段来解决这个问题。如果有，那么就可以在翻译语句中使用该字段名称。（为了有助于我们更清楚地看到语句，我们可以选择在翻译语句中使用一般形式的字段名称。然而，我们将需要使用 SQL 句法中的实际字段名称。）在这种情况下，在图 4-5 中查找 Employees 表的可以用来代替词“names”和“addresses”的字段名称。

Employees	
EmployeeID	PK
EmpFirstName	
EmpLastName	
EmpStreetAddress	
EmpCity	
EmpState	
EmpZipCode	
EmpPhoneNumber	
EmpAreaCode	

图 4-5 Employees 表的结构

为了充分满足“names”和“addresses”的查询需要，我们将实际需要使用这个表中的 6 个字段。EmpFirstName 和 EmpLastName 将同时替代请求中的“names”，而 EmpStreetAddress、EmpCity、EmpState 和 EmpZipCode 将替代“addresses”。现在，对请求应用整个翻译过程，也就是重复之前的步骤。（我们将在翻译语句使用通用形式的字段名称，而在 SQL 语句中使用实际字段名称。）

“I need the names and addresses of all our employees.”

Translation	Select first name, last name, street address, city, state, and ZIP Code from the employees table
Clean Up	Select first name, last name, street address, city, state, ZIP Code from employees
SQL	SELECT EmpFirstName, EmpLastName, EmpStreetAddress, EmpCity, EmpState, EmpZipCode FROM Employees

注意：这个例子也清楚地演示了如何在 SELECT 子句中使用多个字段。我们将在后面详细讨论这个技术。

下一个例子阐述的是第二个技巧，它涉及到查找请求中暗含的字段。假设我们正在尝试完成下面这个请求的翻译。

“What kind of classes do we currently offer?”

(作者: John L. Viescas and Michael J. Hernandez 译者: 曾少宁/陈柳 来源: TT 中国)

如何将信息请求翻译为 SQL SELECT 语句（三）

第一眼看来，可能根据请求来定义一个翻译语句有点难度。请求中并没有显示任何字段名称，甚至没有一个项可以选择，因此我们也无法创建一个完整的翻译语句。现在我们应该怎么做呢？仔细地查看请求中的每个词并确定它是否暗含了 Classes 表中一个字段名称。在我们继续更进一步分析之前，让我们花费点时间来再次研究一下这个请求。我们可以找到这样的词吗？在这种情况下，“kind”这个词可能隐含着 Classes 表的一个字段名称。为什么呢？因为一种 Class 同时也可以被认为是一个 Class 的分类。如果在 Classes 表中有一个分类字段，那么我们就有了用来完成翻译语句、以及后续 SELECT 语句所需要的字段名称。假设在 Classes 表中有一个分类字段，那么就可以对请求两次执行前面的三个步骤。

“What kind of classes do we currently offer?”

Translation	Select category from the classes table
Clean Up	Select category from classes
SQL	SELECT Category FROM Classes

如例所示，这个技术涉及到使用同义词替代请求中的某些词或者短语。如果我们确定了一个词或短语可能隐含一个字段名称，那么我们要尝试用同义词替代它。我们所选择的同义词可能是实际存在数据库上的一个字段。然而，如果第一个出现的同义词不能用，那么可以尝试另外一个。重复这个过程，直到我们找到了可以确定字段名称的同义词，或者我们始终不满意原始的词汇或者任何代表的字段名称的同义词。

注意：除非特别说明，否则在所有例子的 SQL 语句部分的字段名称和表名称都是来自 Appendix B 的 Schema for the Sample Databases 中的示例数据库。这个适用于本书中剩下的部分所举的例子。

扩大域的范围

我们可以在 SELECT 语句中像检索一个字段一样检索多个字段。列出我们想要在 SELECT 语句中使用到的字段名称，同时用逗号分开名称。在图 4-6 显示的语法图中，选择使用多个字段是由 column_name 下面从右到左的线表示的。行中间的逗号表示我们必须在 SELECT 语句中使用的下一个字段名称之前插入一个逗号。

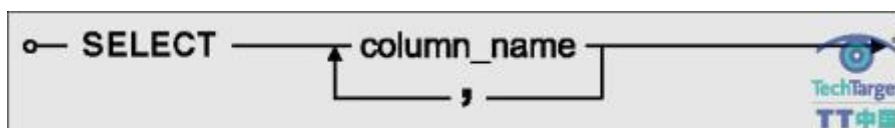


图 4-6 SELECT 语句中使用多字段的语法

选择在 SELECT 语句中使用多字段的可以帮助我们回答这样一些问题。

“Show me a current list of our employees and their phone numbers.”

Translation	Select the last name, first name, and phone number of all our employees from the employees table
Clean Up	Select last name, first name, phone number from employees
SQL	SELECT EmpLastName, EmpFirstName, EmpPhoneNumber FROM Employees

“What are the names and prices of the products we carry, and under what category is each item listed?”

Translation	Select the name, price, and category of every product from the products table
Clean Up	Select name, price, category from products
SQL	SELECT ProductName, RetailPrice, Category FROM Products

当我们在 SELECT 语句中使用多个字段时，我们就获得了看到更宽范围的信息。有时，在 SELECT 语句中的字段顺序并不重要——我们可以按任意顺序列出字段。这样给我们提供了用多种方式查看相同信息的灵活性。

比如，假设我们正在图 4-7 显示的表上操作，同时我们被要求在数据库上发送下面的请求。

“Show me a list of subjects, the category each belongs to, and the code we use in our catalog. But I’d like to see the name first, followed by the category and then the code.”

(作者: John L. Viescas and Michael J. Hernandez 译者: 曾少宁/陈柳 来源: TT 中国)

如何将信息请求翻译为 SQL SELECT 语句（四）

Subjects	
SubjectID	PK
CategoryID	FK
SubjectCode	
SubjectName	
SubjectDescription	

图 4-7 Subjects 表的结构

我们仍然可以将这个请求翻译成一个恰当的 SELECT 语句，即使提出这个请求的人想看到特定顺序的字段。但我们只需按照定义翻译语句时的特定顺序列出字段名称。下面就是当我们将这个要求翻译成 SELECT 语句时的过程。

Translation	Select the subject name, category ID, and subject code from the subjects table
Clean Up	Select subject name, category ID, subject code from subjects
SQL	SELECT SubjectName, CategoryID, SubjectCode FROM Subjects

要求所有字段的捷径

SELECT 子句对于指定字段数目并没有限制——事实上，我们可以列出源表上的所有字段。下面的例子显示了我们用来指定 Subjects 表中所有字段的 SELECT 语句，如图 4-7。

SQL	SELECT SubjectID, CategoryID, SubjectCode,
-----	--

```
SubjectName, SubjectDescription  
FROM Subjects
```

如果要指定源表的所有字段，并且表又包含大量的字段，那么我们将需要输入很多的字段名！幸运的是，SQL Standard 指定了星号作为一个捷径，这样，我们就可以考虑使用缩短的语句。图 4-8 语法图表显示我们可以使用星号作为 SELECT 子句中一系列字段的替代。

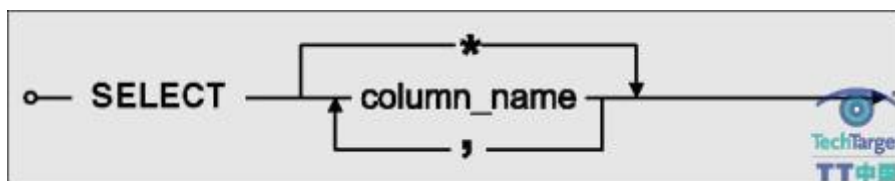


图 4-8 星号快捷语法

当想要在 FROM 子句中指定源表的所有字段时，可以将星号紧跟在 SELECT 子句后面。比如，下面是使用了星号的 SELECT 语句。

```
SQL SELECT * FROM Subjects
```

使用这个语句确实减少输入的字符！然而，以这种方式创建的 SELECT 语句也会有一个问题：星号代表了目前存在于源表中的所有字段，表字段的添加或删除会影响我们在 SELECT 语句的结果集。（真够奇怪的，SQL Standard 宣称添加或删除字段不会影响我们的结果集。）只有当我们必须在结果集中不间断地看到相同的字段时，这才会引起问题。当我们在 SELECT 子句中使用了星号，如果字段已经在源表中删除了，那么我们的数据库将不会发出警告，但是当它找不到我们明确指定的字段时，它将会发出警告。虽然这对于我们的目标并不构成任何问题，但是，当我们深入探讨 SQL 程序时，它将是一个很重大的问题。重要的原则是：只有当我们需要“临时应急”地查询指定表的所有信息时，才使用星号。最后，查询将返回我们所需要的正确的信息并且还将更加的自我文档化。

目前，我们所看到的例子都是根据只需要来自一个表的字段的简单请求。在第三部分，我们将学习如何运行各种更复杂的请求，这些请求是来自几个表的字段的。

(作者: John L. Viescas and Michael J. Hernandez 译者: 曾少宁/陈柳 来源: TT 中国)

在 SQL 中使用 DISTINCT 消除重复行（上）

消除重复行

当运行 SELECT 语句时，我们将不可避免地会碰上有重复记录行的结果集。SQL 对于这样的结果集的出现不会给出任务的警告。在 SELECT 语句中的使用 DISTINCT 关键字，结果集就会去除和清理所有重复的记录行。如 4-9 显示了 DISTINCT 关键字的语法图。

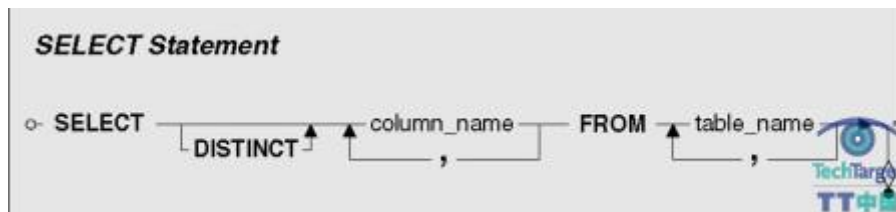


图 4-9DISTINCT 关键字的语法

如图所示，DISTINCT 是一个可选的关键字，它处理 SELECT 子句中指定的一组字段。DISTINCT 关键字会要求数据库系统逐行地将所有字段作为一个单元进行值检查，同时删除所有发现的冗余行。接着，将剩下的唯一的行返回结果集。下面的例子显示了在恰当的环境中 DISTINCT 关键字所带来的不同之处。

假设我们向数据库上发送了下面的请求。

```
"Which cities are represented by our bowling league membership?"
```

问题看起来非常简单，因此我们直接进入翻译过程。

Translation	Select city from the bowlers table
Clean Up	Select city from bowlers

SQL	SELECT City FROM Bowlers
-----	--------------------------

这里问题在于这个 SELECT 语句的结果集将显示在 Bowlers 表中查找到的每个城市名称的每次出现。比如，如果有 20 个人来自 Bellevue，7 个人来自 Kent 和 14 个人来自 Seattle，那么结果集就显示 20 个 Bellevue 出现的城市，7 个 Kent 出现的城市和 14 个 Seattle 出现的城市。很明显，这样冗余的信息是不必要的。我们所想要看到的是在 Bowlers 表中查找到的每个城市名称的一次出现。我们可以通过在 SELECT 语句中使用 DISTINCT 关键字来消除冗余信息来解决这个问题。

现在我们使用 DISTINCT 关键字，并再次运行这个请求的翻译过程。注意，我们同时在 Translation 步骤和 Clean Up 步骤中包含了词 “distinct”。

“Which cities are represented by our bowling league membership?”

Translation	Select the distinct city values from the bowlers table
Clean Up	Select distinct city from bowlers
SQL	SELECT DISTINCT City FROM Bowlers

(作者: John L. Viescas and Michael J. Hernandez 译者: 曾少宁/陈柳 来源: TT 中国)

在 SQL 中使用 DISTINCT 消除重复行（下）

这个 SELECT 语句的结果集明确地显示了我们正在 Bowlers 表中查找的——每个不同的（或唯一的）城市的单次出现。

我们也可以在多字段上使用 DISTINCT 关键字。让我们修改一下上面的例子，改为同时查询 Bowlers 表中的城市和州。那么新的 SELECT 语句会变成这样：

```
SELECT DISTINCT City, State FROM Bowlers
```

SELECT 语句返回一个结果集，它包含的记录都是独一无二的，并显示了相同名称的城市之间明确的区别。比如，它显示了“Portland, ME”、“Portland, OR”、“Hollywood, CA”和“Hollywood, FL”之间的区别。值得注意的是，大多数数据系统都会按我们所指定的字段的顺序来排序输出，因此，我们将看到这些排序的值“Portland, ME”、“Portland, OR”、“Hollywood, CA”和“Hollywood, FL”。然而，SQL Standard 并没有要求按这个顺序排序结果。如果你想要保证分类排序，那么可以阅读下个部分关于 ORDER BY 子句的内容。

在恰当的时候使用 DISTINCT 关键字是非常有用的。请当你确切想要结果集中的记录行是独一无二时使用它。

注意：对于包含图形界面的数据库系统，我们通常可以要求查询的结果集是以可更新的网格行和列显示的。我们可以在一行中的一列输入新的值，然后数据库系统会更新表中存储的值。（数据库系统事实上在后台为我们执行一个 UPDATE 查询——我们可以在第 15 章“更新数据集”中阅读更多这方面的内容。）

然而，在所有我们研究的数据库系统中，如果我们使用了 DISTINCT 关键字，那么结果集的记录行就不能更新。为了更新记录行中的字段，数据库系统必须能够唯一地确定我们想要修改的具体的行和字段。如果使用了 DISTINCT，在每个行中所看到的值可能是十几个重复行的估值结果。如果我们尝试更新其中的一个字段，那么数据库将无法获知应该修改哪一行。同时，数据库系统也无法获知是否我们打算将所有的行修改为同一个重复值。

(作者: John L. Viescas and Michael J. Hernandez 译者: 曾少宁/陈柳 来源: TT 中国)

在 SQL 中使用 SELECT 查询的 ORDER BY 子句（上）

排序信息

在本章开头，我们就谈到 SELECT 操作可以分成三个较小的操作：SELECT 语句、SELECT 表达式和 SELECT 查询。同时，我们还谈到可以用不同的方式组合这些操作来实现复杂的请求。然而，我们也需要组合这些操作来实现结果集中记录行的排序。

一般情况下，SELECT 语句返回的结果集的记录行是无序的。它们的顺序一般是基于它们在表中的物理位置的。（实际的顺序往往是由我们的数据库系统动态确定的，而这个系统则自己决定如何最有效地满足我们的要求。）排序结果集的唯一方法是在 SELECT 查询中嵌入另一个 SELECT 查询，如图 4-10 所示。我们使用一个有 ORDER BY 子句的 SELECT 语句定义 SELECT 查询。SELECT 查询的 ORDER BY 子句可以让我们指定最终结果集中记录行的顺序。在后面的文章中，我们将学习到如果在实际中将另一个 SELECT 语句或 SELECT 表达式嵌入到一个 SELECT 语句来回答非常复杂的问题。然而，SELECT 查询并不是可以随便嵌套的。

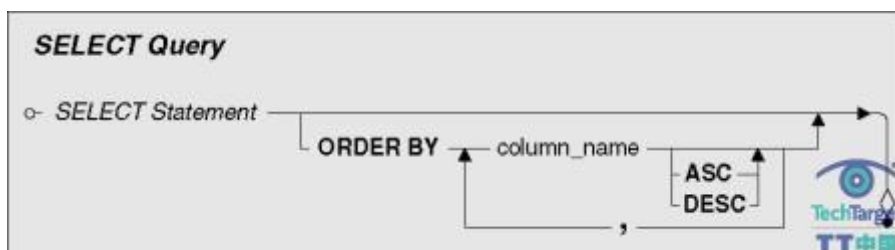


图 4-10 SELECT 查询的语法图

注意：在本书中，我们使用的术语是 SQL Standard 或者大多数数据库系统中常见的术语。然而，2003 SQL Standard 将 ORDER BY 子句定义为游标（cursor，一个可以定义在应用程序内部的对象）的一部分，数组（array，一系列组成一个逻辑表的值，如子查

询<subquery>，将在第 11 章“子查询”中探讨)的一部分，或者标量子查询 (scalar subquery, 只返回一个值的子查询)的一部分。此书并不涉及游标和数组的全面探讨。因为几乎所有 SQL 的执行都允许我们在 SELEC 语句后面加上一个 ORDER BY 子句，我们可以将它保存在视图中，所以我们用 SELECT query 这个词汇来描述这种类型的语句。我们也会对用于联机显示或者报表用途的查询最终输出的排序进行探讨。我们都知道 2007/2008 标准草案允许在更多的地方使用 ORDER BY，但是在本书中我们将用单独的章节对此主题进行探讨。

ORDER BY 子句可以以一个或多个字段对指定的 SELECT 语句的结果集进行排序，同时为每个字段指定升序或降序排序顺序。在 ORDER BY 子句中我们可以使用的字段是那些已经在 SELECT 子句中列出的。(虽然这些要求都在 SQL Standard 中规定了，但是有些供应商的实现允许我们完全忽略它。然而，在本书中我们的所有例子都是遵从这个要求的。)当我们在 ORDER BY 子句中使用两个或多个字段时，要用逗号将每个字段分开。当排序完成后，SELECT 查询就返回一个最终的结果集。

注意：ORDER BY 子句并不影响表中行的物理顺序。如果我们确实需要修改记录行的物理顺序，那么请参照数据库软件文档中的正确操作流程。

首要事情：排序序列

在我们列举一些使用 SELECT 查询的例子之前，排序序列其实就是顺序。ORDER BY 子句排序信息的方式是取决于数据库软件中所使用的排序序列的。排序序列确定了我们的操作系统中所指定的语言字符集中列出的每个字符的先后顺序。比如，它定义了小写字母是否存储在大写字母之前，或者是否大小写敏感。检查数据库软件的文档，或者咨询数据库管理员来确定数据库的默认排序序列。关于排序序列的更多信息，可以阅读 Comparing String Values: A Caution in Chapter 6。

(作者: John L. Viescas and Michael J. Hernandez 译者: 曾少宁/陈柳 来源: TT 中国)

在 SQL 中使用 SELECT 查询的 ORDER BY 子句（中）

关于顺序

有了 ORDER BY 子句，我们可以以更有意义的方式来表现我们从数据库中检索到的信息。这对于简单的以及复杂的请求都是一样适用的。我们现在可以修改我们的请求，使它们包括一些排序的需求。比如，诸如“What are the categories of classes we currently offer?”的问题可以修改为“List the categories of classes we offer and show them in alphabetical orde.”。

Translation	Select category from the classes table and order by category
Clean Up	Select category from classes order by category
SQL	SELECT Category FROM Classes ORDER BY Category

在运行 SELECT 查询之前，我们需要调整定义翻译语句的方式。这包括在翻译语句的后面添加一个新的部分来解释请求中指定的新排序要求。使用这个新的形式来定义翻译语句。

```
Select <item> from the <source> and order by <column(s)>
```

现在我们的请求中包含了诸如“sort the results by city”、“show them in order by year”或“list them by last name and first name”的短语，仔细地研究请

求，然后确定我们需要在哪个字段或哪些字段来实现排序。这是一个简单的练习，因为大多数人都使用这些类型的短语，并且排序所需要的字段通常都是很明显的。在我们确定了恰当的一个或多个字段之后，就可以用它们替代翻译语句中的<column(s)>。让我们来看一下这个简单的请求是如何构造的。

```
"List the categories of classes we offer and show them in alphabetical order."
```

在这个例子中，我们可以假设将使用 Category 来排序，因为它是请求中唯一出现的字段。我们也可以假设排序是按照升序进行的，因为请求中并没有任何相反的要求。因此这是一个很可靠的假设。根据 SQL Standard，如果我们没有指定排序顺序，那么升序排列是自动假设的。然而，如果我们想要更明确点，那么我们可以在 ORDER BY 子句的后面插入 ASC。

在下面的请求中，对于排序所需要的字段作了更明确地定义。

```
"Show me a list of vendor names in ZIP Code order."
```

Translation	Select vendor name and ZIP Code from the vendors table and order by ZIP Code
Clean Up	Select vendor name ZIP Code from vendors order by ZIP Code
SQL	SELECT VendName, VendZipCode FROM Vendors ORDER BY VendZipCode

一般来说，大多数的人如果想看到降序排列的信息，他们都会明确地说明。当出现这种情况时，我们需要以相反的排列显示结果集，也就是在 ORDER BY 子句的特定字段后插入 DESC 关键字。比如，当我们想看到以 ZIP 代码降序排列的信息时，我们可以这样修改前面例子中的 SELECT 语句。

SQL	SELECT VendName, VendZipCode FROM Vendors ORDER BY VendZipCode DESC
-----	---

下面的例子是一个要求多字段排序的更加复杂的请求。这个例子与前面的例子之间的不同在于在这个例子中的 ORDER BY 子句使用了更多的字段。注意，字段是由逗号分开的，这与图 4-10 所显示的语法图是一致的。

“Display the names of our employees, including their phone number and ID number, and list them by last name and first name.”

Translation	Select last name, first name, phone number, and employee ID from the employees table and order by last name and first name
Clean Up	Select last name, first name, phone number, employee ID from employees order by last name first name
SQL	SELECT EmpLastName, EmpFirstName, EmpPhoneNumber, EmployeeID FROM Employees ORDER BY EmpLastName, EmpFirstName

(作者: John L. Viescas and Michael J. Hernandez 译者: 曾少宁/陈柳 来源: TT 中国)

在 SQL 中使用 SELECT 查询的 ORDER BY 子句（下）

在 ORDER BY 子句的字段中有一个很有趣的地方是我们可以为每个字段指定一个不同的排序顺序。在前面的例子中，我们可以为包含“姓”的字段指定降序排序，而对包含“名”的字段指定升序排序。修改之后，SELECT 语句变成这样：

```
SELECT EmpLastName, EmpFirstName, EmpPhoneNumber, EmployeeID
FROM Employees
ORDER BY EmpLastName DESC, EmpFirstName ASC
```

虽然我们并不需要明确地使用 ASC 关键字，但是如果我们将它包含在内，那么语句会更加清晰。

前面这个例子引发了一个有趣的问题：在 ORDER BY 子句中字段的顺序是否很重要？答案是肯定的！顺序是很重要的，因为数据库系统将从左到右评估 ORDER BY 子句中的字段。同时，顺序的重要性是与我们所使用的字段的数目成正比的。要总是保证 ORDER BY 子句中的字段的顺序恰当，这样结果才会按照恰当的顺序排序。

注意：微软的数据库产品 (Microsoft Office Access and Microsoft SQL Server) 包含一个很有趣的扩展，它通过在 SELECT 子句中使用 TOP 关键字来允许根据 ORDER BY 子句请求一个子集的记录行。比如，我们可以通过请求找到 Sales Orders 数据库中五个最贵的产品：

```
SELECT TOP 5 ProductName, RetailPrice
FROM Products
ORDER BY RetailPrice DESC
```


数据库按照价格降序的方式排序 Products 表中的记录行，然后返回前面五行记录。这两个数据库系统都允许我们以百分数的形式指定返回的行数。比如，我们可以通过请求价格排在前 10%的产品：

```
SELECT TOP 10 PERCENT ProductName, RetailPrice
FROM Products
ORDER BY RetailPrice DESC
```

事实上，如果我们想要在一个视图中指定 ORDER BY，那么 SQL Server 会要求我们使用 TOP 关键字。如果我们想要所有的行，那么我们必须指定 TOP 100 PERCENT。因此，我们将看到所有包含 ORDER BY 子句的 SQL Server 的示例视图都指定了 TOP 100 PERCENT。而在 Microsoft Access 则没有这样的限制。

保存工作

保存 SELECT 语句——每个主流的数据库软件程序都提供给我们一种保存 SQL 的方法。当我们需要在数据库上发出相同请求时，保存语句将减少我们重复创建的时间。当我们保存了 SELECT 语句后，设置一个有意义的名称将有助于我们记住该语句所提供了什么样的信息。同时，如果我们的数据库软件支持这样的操作，那么可以写入一个简明的描述说明语句的作用。当我们有一段时间没有查看特定的 SELECT 语句，而我们又需要记住我们之前为什么创建它时，这个简单的描述将变非常有用。

一个已保存的 SELECT 语句是在不同的数据库程序中可能被分类为不同的对象，有的是一个查询，有的是一个视图、方法或者存储过程等。除了名称，每个数据库程序会提供给我们一种执行、或者运行已保存语句以及操作运行结果集的方法。

注意：对于接下来的探讨，我们将使“查询”来代表已保存的 SELECT 语句，并用“执行”代表操作它的方法。

执行查询常见的方法有两种。第一种是通过交互式设备（如工具栏或查询网格上的命令）操作，而第二种是在程序代码中操作。第一种方法的使用是很广泛的。而第二种方法只有当我们使用数据库软件的程序语言时才需要。虽然教导你如何创建和使用 SQL 语句是我们的职责所在，然而，在你的数据库软件程序上创建、存储和执行它们则是你应该去学习的。

(作者: John L. Viescas and Michael J. Hernandez 译者: 曾少宁/陈柳 来源: TT 中国)

SQL SELECT 语句和 SELECT 查询示例（一）

示例语句

目前我们已经探讨了 SELECT 语句和 SELECT 查询的基本特性，接下来让我们来看看在不同的环境中如何应用这些例子。这些示例各自包含了一个示例数据库，同时还阐述了 SELECT 语句、SELECT 查询的使用，以及两种用于为翻译语句创建字段的辅助技术。我们还涉及到这些操作将返回的示例结果集，并且这些结果集都会被直接附在 SQL 语句之后。结果集之前的名称有双重意义：它标识结果集，同时它也是我们在示例中指派给 SQL 语句的名称。

此时，你可能会疑惑为何我们要给每个 SQL 语句指派名称，这是因为我们要保存它们！事实上，我们会对这些示例以及整本书中的所有 SQL 语句都进行命名和保存。它们都存储在恰当的示例数据库中（如示例所示），同时我们还在与本章相关的查询的名称前加了前缀“CH04”。你可以按照本书的“介绍”部分的操作说明将示例下载到你的计算机上。这样，在你着手自己写 SQL 查询之前，你可以有机会先熟悉一下这些语句。

注意：小提示：在这些示例中的所有字段名称和表名称都是引用自 Appendix B 的示例数据库。

销售定单数据库

“Show me the names of all our vendors.”

Translation	Select the vendor name from the vendors table
Clean Up	Select vendor name from vendors
SQL	SELECT VendName

	FROM Vendors
--	--------------

CH04_Vendor_Names (10 rows)

VendName
Shinoman, Incorporated
Viscount
Nikoma of America
ProFormance
Kona, Incorporated
Big Sky Mountain Bikes
Dog Ear
Sun Sports Suppliers
Lone Star Bike Supply
Armadillo Brand

“What are the names and prices of all the products we carry?”

Translation	Select product name, retail price from the products table
Clean Up	Select product name, retail price from products
SQL	SELECT ProductName, RetailPrice FROM Products

CH04_Product_Price_List (40 rows)

ProductName	Retail Price
Trek 9000 Mountain Bike	\$1,200.00

Eagle FS-3 Mountain Bike	\$1,800.00
Dog Ear Cyclecomputer	\$75.00
Victoria Pro All Weather Tires	\$54.95
Dog Ear Helmet Mount Mirrors	\$7.45
Viscount Mountain Bike	\$635.00
Viscount C-500 Wireless Bike Computer	\$49.00
Kryptonite Advanced 2000 U-Lock	\$50.00
Nikoma Lok-Tight U-Lock	\$33.00
Viscount Microshell Helmet	\$36.00
< < more rows here > >	

"Which states do our customers come from?"

Translation	Select the distinct state values from the customers table
Clean Up	Select distinct state from customers
SQL	SELECT DISTINCT CustState FROM Customers

CH04_Customer_States (4 rows)

CustState
CA
OR
TX
WA

(作者: John L. Viescas and Michael J. Hernandez 译者: 曾少宁/陈柳 来源: TT 中国)

SQL SELECT 语句和 SELECT 查询示例（二）

娱乐经纪公司数据库

“List all entertainers and the cities they’re based in, and sort the results by city and name in ascending order.”

Translation	Select city and stage name from the entertainers table and order by city and stage name
Clean Up	Select city stage name from entertainers order by city stage name
SQL	<pre>SELECT EntCity, EntStageName FROM Entertainers ORDER BY EntCity ASC, EntStageName ASC</pre>

CH04_Entertainer_Locations (13 rows)

EntCity	EntStageName
Auburn	Caroline Coie Quartet
Auburn	Topazz
Bellevue	Jazz Persuasion
Bellevue	Jim Glynn
Bellevue	Susan McLain
Redmond	Carol Peacock Trio
Redmond	JV & the Deep Six

Seattle	Coldwater Cattle Company
Seattle	Country Feeling
Seattle	Julia Schnebly
< < more rows here > >	

“Give me a unique list of engagement dates. I’m not concerned with how many engagements there are per date.”

Translation	Select the distinct start date values from the engagements table
Clean Up	Select distinct start date from engagements
SQL	<pre>SELECT DISTINCT StartDate FROM Engagements</pre>

CH04_Engagement_Dates (64 rows)

StartDate
2007-09-01
2007-09-10
2007-09-11
2007-09-15
2007-09-17
2007-09-18
2007-09-24
2007-09-29
2007-09-30
2007-10-01

< < more rows here > >

教学安排数据库

"Can we view complete class information?"

Translation	Select all columns from the classes table
Clean Up	Select * from classes
SQL	SELECT * FROM Classes

CH04_Class_Information (76 rows)

ClassID	SubjectID	ClassRoomID	Credits	StartTime	Duration	< <other columns> >
1000	11	1231	5	10:00	50	...
1002	12	1619	4	15:30	110	...
1004	13	1627	4	08:00	50	...
1006	13	1627	4	09:00	110	...
1012	14	1627	4	13:00	170	...
1020	15	3404	4	13:00	110	...
1030	16	1231	5	11:00	50	...
1031	16	1231	5	14:00	50	...
1156	37	3443	5	08:00	50	...
1162	37	3443	5	09:00	80	...
< < more rows						

here > >						
----------	--	--	--	--	--	--

“Give me a list of the buildings on campus and the number of floors for each building. Sort the list by building in ascending order.”

Translation	Select building name and number of floors from the buildings table, ordered by building name
Clean Up	Select building name number of floors from buildings , order by building name
SQL	<pre>SELECT BuildingName, NumberOfFloors FROM Buildings ORDER BY BuildingName ASC</pre>

CH04_Building_List (6 rows)

BuildingName	NumberOfFloors
Arts and Sciences	3
College Center	3
Instructional Building	3
Library	2
PE and Wellness	1
Technology Building	2

保龄球联盟数据库

“Where are we holding our tournaments?”

Translation	Select the distinct tourney location values from the tournaments table
Clean Up	Select distinct tourney location from tournaments
SQL	SELECT DISTINCT TourneyLocation FROM Tournaments

CH04_Tourney_Locations (7 rows)

TourneyLocation
Acapulco Lanes
Bolero Lanes
Imperial Lanes
Red Rooster Lanes
Sports World Lanes
Thunderbird Lanes
Totem Lanes

“Give me a list of all tournament dates and locations. I need the dates in descending order and the locations in alphabetical order.”

Translation	Select tourney date and location from the tournaments table and order by tourney date in descending order and location in ascending order
Clean Up	Select tourney date location from tournaments order by tourney date descending location ascending
SQL	SELECT TourneyDate, TourneyLocation

	FROM Tournaments ORDER BY TourneyDate DESC, TourneyLocation ASC
--	--

CH04_Tourney_Dates (14 rows)

TourneyDate	TourneyLocation
2008-08-15	Totem Lanes
2008-08-08	Imperial Lanes
2008-08-01	Sports World Lanes
2008-07-25	Bolero Lanes
2008-07-18	Thunderbird Lanes
2008-07-11	Red Rooster Lanes
2007-12-04	Acapulco Lanes
2007-11-27	Totem Lanes
2007-11-20	Sports World Lanes
2007-11-13	Imperial Lanes
< < more rows here > >	

食谱数据库

“What types of recipes do we have, and what are the names of the recipes we have for each type? Can you sort the information by type and recipe name?”

Translation	Select recipe class ID and recipe title from the recipes table and order by recipe class ID and recipe title
Clean Up	Select recipe class ID recipe title from recipes order by recipeclass ID recipe title

SQL	<pre>SELECT RecipeClassID, RecipeTitle FROM Recipes ORDER BY RecipeClassID ASC, RecipeTitle ASC</pre>
-----	---

CH04_Recipe_Classes_And_Titles (15 rows)

RecipeClassID	RecipeTitle
1	Fettuccini Alfredo
1	Huachinango Veracruzana (Red Snapper, Veracruz style)
1	Irish Stew
1	Pollo Picoso
1	Roast Beef
1	Salmon Filets in Parchment Paper
1	Tourtière (French-Canadian Pork Pie)
2	Asparagus
2	Garlic Green Beans
3	Yorkshire Pudding
< < more rows here > >	

“Show me a list of unique recipe class IDs in the recipes table.”

Translation	Select the distinct recipe class ID values from the recipes table
Clean Up	Select distinct recipe class ID from recipes
SQL	SELECT DISTINCT RecipeClassID

```
FROM Recipes
```

CH04_Recipe_Class_Ids(6 rows)

RecipeClassID
1
2
3
4
5
6

(作者: John L. Viescas and Michael J. Hernandez 译者: 曾少宁/陈柳 来源: TT 中国)

SQL SELECT 语句和 SELECT 查询示例（三）

总结

在本章中，我们介绍了 SELECT 操作，它是 SQL 中 4 个数据处理操作的其中一个。（其它的是 UPDATE、INSERT 和 DELETE，它们会在第 5 部分中介绍。）同时，我们还探讨了 SELECT 操作是如何分成三个较小的操作的：SELECT 语句、SELECT 表达式和 SELECT 查询。

接着，我们还探讨了 SELECT 语句的组成子句。其中，SELECT 和 FROM 子句是用于检索数据库信息的基本子句，而其余的 WHERE、GROUP BY 和 HAVING——是用于有条件地处理和过滤 SELECT 子句返回的信息。

我们还简单地探讨了数据和信息之间的不同。存储在数据库上的数值是数据，而信息则是已经以某种方式处理的数据，并且它使数据变得有意义。同时，我们还介绍 SELECT 语句所返回的信息行就是结果集。

接下来的探讨主题是检索信息，首先探讨的是 SELECT 语句的基本形式。我们学习了如何通过三个步骤的技巧建立恰当的 SELECT 语句，这三个步骤能将一个请求翻译成恰当的 SQL 语句。同时，我们还学习了在 SELECT 子句中使用两个或多个字段来扩大从数据库中检索信息范围。在这个部分中，我们还简单介绍了 DISTINCT 关键字，这是一种从结果集中删除重复行的方法。

接着，我们还介绍了 SELECT 查询，以及它是如何与 SELECT 语句一起来排序 SELECT 语句的结果集。由于 SELECT 查询是唯一包含 ORDER BY 子句的 SELECT 操作，因此它是非常必要的。我们还探讨了通过一个字段或者多字段来排序信息的 ORDER BY 子句，这其中每一个字段都可以指定升序或降序的排序方式。接下来是简单探讨了关于保存 SELECT 语句的问题，我们知道了可以将语句保存为一个查询或者视图。

最后，我们列举了使用示例数据库中不同表中的大量例子。这些例子演示了本章中所涉及到的不同概念和技巧是如何在一些典型的环境和应用中使用的。在接下来的章节中，我们还仔细探讨了 SELECT 子句，并且还阐述了如何从一系列字段检索出除信息以外的东西。

下面是一些我们可以自己实践的请求。

(作者: John L. Viescas and Michael J. Hernandez 译者: 曾少宁/陈柳 来源: TT 中国)

SQL SELECT 语句和 SELECT 查询示例（四）

查询实践

下面列出的是示例数据库中的请求语句以及方案查询的名称。你可以练习一下，尝试写出每个请求所需要的 SQL，并通过示例中已保存的查询来检查你的答案。不要担心你的语句与已存的查询语句不同——只要结果集一样就可以。

销售定单数据库

1. *"Show me all the information on our employees."*

结果在“CH04_Employee_Information”表中，共 8 行。

2. *"Show me a list of cities, in alphabetical order, where our vendors are located, and include the names of the vendors we work with in each city."*

结果在“CH04_Vendor_Locations”表中，共 10 行。

娱乐经纪公司数据库

1. *"Give me the names and phone numbers of all our agents, and list them in last name/first name order."*

结果在“CH04_Agent_Phone_List”表中，共 9 行。

2. *"Give me the information on all our engagements."*

结果在“CH04_Engagement_Information”表中，共 111 行。

3. *"List all engagements and their associated start dates. Sort the records by date in descending order and by engagement in ascending order."*

结果在“CH04_Scheduled_Engagements”表中，共 111 行。

课程表数据库

1. *"Show me a complete list of all the subjects we offer."*

结果在“CH04_Subject_List”表中，共 56 行。

2. *"What kinds of titles are associated with our faculty?"*

结果在“CH04_Faculty_Titles”表中，共 3 行。

3. *"List the names and phone numbers of all our staff, and sort them by last name and first name."*

结果在“CH04_Staff_Phone_List”表中，共 27 行。

保龄球联盟数据库

1. *"List all of the teams in alphabetical order."*

结果在“CH04_Team_List”表中，共 8 行。

2. *"Show me all the bowling score information for each of our members."*

结果在“CH04_Bowling_Score_Information”表中，共 1,344 行。

3. *"Show me a list of bowlers and their addresses, and sort it in alphabetical order."*

结果在“CH04_Bowler_Names_Addresses”表中，共 32 行。

食谱数据库

1. *“Show me a list of all the ingredients we currently keep track of.”*

结果在“CH04_Complete_Ingredients_List”表中，共 79 行。

2. *“Show me all the main recipe information, and sort it by the name of the recipe in alphabetical order.”*

结果在“CH04_Main_Recipe_Information”表中，共 15 行。

(作者: John L. Viescas and Michael J. Hernandez 译者: 曾少宁/陈柳 来源: TT 中国)