

IBM[®] DB2[®] 通用数据库



SQL 入门

版本 7

IBM[®] DB2[®] 通用数据库



SQL 入门

版本 7

在使用本资料 and 它支持的产品之前，请参阅第109页的『附录C. 注意事项』中的一般信息。

本文档包含 IBM 的专利信息。它在许可协议下提供，并受版权法保护。本出版物包含的信息不包括任何产品保证，且本手册提供的任何声明不应作如此解释。

通过您当地的 IBM 代表或 IBM 分部可订购出版物，或者，通过致电 1-800-879-2755（在美国）或 1-800-IBM-4YOU（在加拿大）来订购出版物。

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 对于您所提供的任何信息，有权利以任何它认为适当的方式使用或散发，而不必对您负任何责任。

© Copyright International Business Machines Corporation 1993, 2000. All rights reserved.

目录

欢迎使用	v	第5章 表达式和子查询	33
本书的相关文档	v	标量全查询	33
突出显示约定	v	转换数据类型	33
第1章 关系数据库和 SQL	1	条件表达式	34
第2章 组织数据	3	表表达式	35
表	3	嵌套表表达式	35
视图	3	公共表表达式	36
模式	4	相关名	37
数据类型	4	相关子查询	38
第3章 创建表和视图	9	实现相关子查询	40
创建表	9	第6章 在查询中使用运算符和谓词	45
插入数据	10	用集合运算符组合查询	45
更改数据	12	UNION 运算符	45
删除数据	12	EXCEPT 运算符	46
创建视图	13	INTERSECT 运算符	47
使用视图来处理数据	14	谓词	47
第4章 使用 SQL 语句存取数据	17	使用 IN 谓词	48
连接数据库	18	使用 BETWEEN 谓词	48
调查错误	18	使用 LIKE 谓词	49
选择列	19	使用 EXISTS 谓词	49
选择行	20	定量谓词	50
将行进行排序	22	第7章 高级 SQL	51
除去重复行	23	用约束和触发器实施商业规则	51
运算次序	24	关键字	51
使用表达式来计算值	24	唯一约束	52
给表达式命名	25	参考完整性约束	52
从多个表中选择数据	25	表检查约束	53
使用子查询	26	触发器	54
使用函数	27	连接	57
列函数	27	复杂查询	61
标量函数	28	ROLLUP 和 CUBE 查询	61
表函数	29	递归查询	62
分组	29	OLAP 函数	62
将 WHERE 子句与 GROUP BY 子句一起	30	第8章 定制和增强数据操作	65
使用	30	用户定义类型	65
在 GROUP BY 子句之后使用 HAVING 子	30	用户定义函数	66
句	30	大对象 (LOB)	67
		操作大对象 (LOB)	67

专用寄存器	68	Adamson 的相片	87
目录视图的介绍	68	Adamson 的简历	87
从系统目录中选择行	68	Walker 的相片	88
		Walker 的简历	89
附录A. 样本数据库表	71		
样本数据库	72	附录B. 使用 DB2 资料库	91
创建样本数据库	72	DB2 PDF 文件和打印的书籍	91
要擦除样本数据库	72	DB2 信息	91
CL_SCHED 表	72	打印 PDF 书籍	99
DEPARTMENT 表	73	订购打印书籍	100
EMPLOYEE 表	73	DB2 联机文档	101
EMP_ACT 表	76	访问联机帮助	101
EMP_PHOTO 表	78	查看联机信息	103
EMP_RESUME 表	78	使用 DB2 向导	105
IN_TRAY 表	79	设置文档服务器	106
ORG 表	79	搜索联机信息	107
PROJECT 表	79		
SALES 表	81	附录C. 注意事项	109
STAFF 表	82	注册商标	111
STAFFG 表	83		
数据类型为 BLOB 和 CLOB 的样本文件	84	索引	113
Quintana 的相片	84		
Quintana 的简历	84	与 IBM 联系	119
Nicholls 的相片	85	产品信息	119
Nicholls 的简历	86		

欢迎使用

本书是为“结构化查询语言”(SQL)和关系数据库的初级用户编写的。本书将:

- 讨论在 DB2 产品中使用的 SQL 的基本概念。
- 说明如何执行数据库操作任务。
- 通过简单示例演示任务。

如果您是系统管理员,在尝试本书中的任何示例之前,您应该:

- 按照快速入门一书中概述的那样为您的操作系统安装和配置服务器。使用“第一步骤”选项来创建 SAMPLE 数据库。还可从命令行提示创建 SAMPLE 数据库。参见 *SQL Reference* 以获取详情。注意,不要将自己的数据放入 DB2 SAMPLE 数据库。
- 通过遵循快速入门一书中的指导创建 DB2 管理员用户 ID。

如果您并非系统管理员,则要确保您具有有效的用户 ID 及适当的权限和特权才可存取 SAMPLE 数据库。

本书的相关文档

您可能会发现下列出版物对您有用:

<i>快速入门</i>	包含安装和使用数据库管理程序所需要的信息。
<i>SQL Reference</i>	包含 SQL 参考信息。
<i>管理指南</i>	包含设计、实现与维护要在本地或客户机/服务器环境中存取的数据库所需的信息。
<i>Application Development Guide</i>	讨论应用程序开发过程以及如何编码、编译和执行以下这样的应用程序:使用嵌入式 SQL 来存取数据库的应用程序,或作为 DB2 存储过程(使用 SQL Procedure 语言或其它受支持的程序设计语言)运行的应用程序。

突出显示约定

本书使用下列约定。

黑体	在示例中,它表示系统预定义的命令和关键字。
<i>斜体字</i>	表示下面其中一种情况: <ul style="list-style-type: none">• 介绍新术语• 引用另一个信息源。

大写字体

表示下面其中一种情况:

- 由系统预定义的命令和关键字。
 - 特定数据值或列名的示例。
-

第1章 关系数据库和 SQL

关系数据库中，数据存储在表中。表是行和列的集合。参见第3页的图1 可获得表的图形示例。列（垂直）和行（水平）已在图上标出。结构化查询语言 (SQL) 是用于通过指定列、表以及它们之间的各种关系来检索或更新数据的。

SQL 是在关系数据库中定义和处理数据的标准化语言。SQL 语句由数据库管理程序执行。数据库管理程序是管理数据的计算机程序。

分区关系数据库是在多个分区（也称为节点）上管理数据的关系数据库。理解分区的一个简单方法是将每个分区视为一台物理计算机。在本书中，我们将把重点集中在单一分区数据库上。

可以使用一个界面（如“命令行处理器” (CLP) 或“命令中心” (CC)），通过交互式 SQL 来存取样本数据库并尝试运行本书中的所有示例。

第2章 组织数据

本章给出了表、视图以及模式的重要概念说明。该说明是一个一般概述，显示了关系数据库的不同构件块之间的连接。最后一节简要讨论了某些重要的和较常用的数据类型。

表

表是由确定的列数和可变的行数组成的逻辑结构。列是一组数据类型相同的值。行是组成表中单个记录的连续的值。在表中不必对行进行排序。要对结果集进行排序，必须在从表中选择数据的 SQL 语句中显式指定排序。在每个列和行的相交处是一个称为值的特定数据项。例如，在图1中，'Sanders' 是表中的一个值。

基表存放用户数据，且它是用 CREATE TABLE 语句创建的。结果表是一组行，数据库管理程序从一个或多个基表选择或生成这组行以满足查询要求。

图1说明了表的一部分。列和行已标记。

The diagram shows a table with four columns and six rows. The columns are labeled 'ID', 'NAME', 'DEPT', and 'J'. The rows are labeled with values: 10, 20, 30, 40, 50, and 60. A bracket above the columns is labeled 'Column' and a bracket to the left of the rows is labeled 'Row'. The table is enclosed in a hand-drawn black border.

	ID	NAME	DEPT	J
10		Sanders	20	Mg
20		Pernal	20	Sa
30		Marengi	38	Mg
40		O'Brien	38	Sa
50		Hanes	15	Mg
60		Quigley	38	Sa

图 1. 表的直观图

视图

视图提供了在一个或多个表中查看数据的替代方法。它是表上的一个动态窗口。

视图允许多个用户查看同一数据的不同表示。例如，几个用户可以同时存取一个关于雇员的数据表。经理可看到关于他的或她的雇员的数据，但看不到其它部门的雇员的数据。招募人员可看到所有雇员的任职日期但看不到他们的工资情况，而财会人员可看到工资情况但看不到任职日期。每个这样的用户用一个从实表派生的视图进行操作。每个视图都显示为一个表并有自己的名称。

使用视图的优点是您可以使用它们来控制对敏感数据的存取。所以，不同的人可以存取数据的不同列或行。

模式

模式是已命名的对象（如表和视图）的集合。模式提供了数据库中对象的逻辑分类。

当创建表、视图或任何其它命名对象时，即隐式创建了模式。或者，可以使用 `CREATE SCHEMA` 语句显式创建它。

在创建命名对象时，可用特定模式的名称来限定（关联）该对象的名称。命名对象的名称有两个部分，其中，名称的前一个部分是对其指定对象的模式名。如果不指定模式名，则对对象指定缺省模式。（缺省模式的名称是执行语句的用户的权限 *ID*。）

对于交互式 SQL（一种用来执行本书中的示例的方法），权限 *ID* 为用 `CONNECT` 语句指定的用户 *ID*。例如，如果表名为 `STAFF`，所指定的用户 *ID* 为 `USERXYZ`，则限定的表名为 `USERXYZ.STAFF`。参见第18页的『连接数据库』以获取关于 `CONNECT` 语句的详情。

某些模式名是系统保留的。例如，当预安装的用户定义函数属于 `SYSFUN` 模式时，内部函数处于 `SYSIBM` 模式。参考 *SQL Reference* 以获取关于 `CREATE SCHEMA` 语句的详情。

数据类型

数据类型定义常数、列、宿主变量、函数、表达式以及专用寄存器可接受的值。本节描述示例中引用的数据类型。有关其他数据类型的完整列表和说明，参考 *SQL Reference*。

字符串

字符串是一个字节序列。字符串的长度为该序列中的字节数。如果长度为零，则该字符串的值称为空字符串。

定长字符串

CHAR(x) 是定长字符串。长度属性 x 必须在 1 和 254 之间且包括两者。

变长字符串

变长字符串有三种类型：VARCHAR、LONG VARCHAR 以及 CLOB。

VARCHAR(x) 类型是变长字符串，因此，可以将长度为 9 的字符串插入 VARCHAR(15) 中，而该字符串的长度将仍然为 9。

参见第67页的『大对象 (LOB)』以获取关于 CLOB 的详情。

图形字符串

图形字符串是一个双字节字符数据序列。

定长图形字符串

GRAPHIC(x) 是定长字符串。长度属性 x 必须在 1 和 127 之间，并包括 1 和 127。

变长图形字符串

变长图形字符串有三种类型：VARGRAPHIC、LONG VARGRAPHIC 以及 DBCLOB。参见第67页的『大对象 (LOB)』以获取关于 DBCLOB 的详情。

二进制字符串

二进制字符串是一个字节序列。它用于保存非传统数据，如图象等。“二进制大对象”(BLOB) 是二进制字符串。参见第67页的『大对象 (LOB)』以了解详情。

数字

所有数字都有符号和精度。精度是除符号位以外的位数或数字数。

SMALLINT

SMALLINT (小型整数) 是精度为 5 位的两字节整数。

INTEGER

INTEGER (大型整数) 是精度为 10 位的四字节整数。

BIGINT

一个 BIGINT (大整数) 是一个精度为 19 位的 8 字节整数。

REAL REAL (单精度浮点数) 是实数的 32 位近似值。

DOUBLE

DOUBLE (双精度浮点数) 是实数的 64 位近似值。DOUBLE 也称 FLOAT。

DECIMAL(p,s)

DECIMAL 是一个十进制数。小数点的位置由数字的精度(*p*) 和小数位 (*s*) 确定。精度是数字的总位数, 必须小于 32。小数位是小数部分数字的位数且总是小于或等于精度值。如果未指定精度和小数位, 则十进制值的缺省精度为 5, 缺省小数位为 0。

日期时间值

日期时间值是日期、时间以及时间戳记(一个格式为 *yyyymmddhhmmss* 的表示有效日期和时间的 14 位字符串)的表示法。日期时间值可以用于某些算术运算和字符串运算并且与某些字符串是相容的, 但是它们既非字符串也非数字。¹

日期 日期值分为三个部分(年、月以及日)。

时间 时间是用 24 小时制式来指定一天内的时间的值, 分为三个部分(小时、分钟以及秒)。

时间戳记

*时间戳记*是指定日期和时间的值, 分为七个部分(年、月、日、小时、分钟、秒以及微秒)。

空值

空值是一个区别于所有非空值的特殊值。它意味着行中的那一列无任何其他值。所有数据类型都存在空值。

下表突出显示示例中所使用的数据类型的特性。所有数字数据类型都定义在某一确定范围内。该数字数据类型范围也包括在此表中。可以使用此表作为正确数据类型用法的快速参考。

数据类型	类型	特性	示例或范围
CHAR(15)	定长字符串	最大长度为 254	'Sunny day'
VARCHAR(15)	变长字符串	最大长度为 32672	'Sunny day'
SMALLINT	数字	长度为 2 个字节精度为 5 位	范围为 -32768 至 32767
INTEGER	数字	长度为 4 个字节精度为 10 位	范围为 -2147483648 至 2147483647
BIGINT	数字	长度为 8 个字节精度为 19 位	范围为 -9223372036854775808 至 9223372036854775807

1. 本书中我们沿用日期时间值的 ISO 表示。

数据类型	类型	特性	示例或范围
REAL	数字	单精度浮点 32 位近似值	范围是 $-3.402E+38$ 至 $-1.175E-37$, 或 $1.175E-37$ 至 $-3.402E+38$, 或为 0
DOUBLE	数字	双精度浮点 64 位近似值	范围是 $-1.79769E+308$ 至 $-2.225E-307$ 或 $2.225E-307$ 至 $1.79769E+308$, 或为 0
DECIMAL(5,2)	数字	精度是 5 小数位是 2	范围是 $-10^{31}+1$ 至 $10^{31}-1$
DATE	日期时间	由三部分组成的值	1991-10-27
TIME	日期时间	由三部分组成的值	13.30.05
TIMESTAMP	日期时间	由七部分组成的值	1991-10-27-13.30.05.000000

参见 *SQL Reference* 中的数据类型兼容性表以获取详情。

第3章 创建表和视图

本章描述了如何才能在 DB2 通用数据库中创建和操作表和视图。读者可通过图表和示例来研究表和视图的关系。

本章包括:

- 创建表和创建视图
- 插入数据
- 更改数据
- 删除数据
- 使用视图来处理数据

创建表

使用 `CREATE TABLE` 语句创建自己的表，指定列名和类型以及约束。约束在第 51 页的『用约束和触发器实施商业规则』中进行讨论。

下列语句创建一个名称为 `PERS` 的表，该表与 `STAFF` 表类似，但有一个附加列 `BIRTH_DATE`。

```
CREATE TABLE PERS
( ID          SMALLINT          NOT NULL,
  NAME        VARCHAR(9),
  DEPT        SMALLINT WITH DEFAULT 10,
  JOB         CHAR(5),
  YEARS       SMALLINT,
  SALARY      DECIMAL(7,2),
  COMM        DECIMAL(7,2),
  BIRTH_DATE  DATE)
```

此语句创建一个不包含数据的表。下一节将描述如何将数据插入新表。

如示例中所示，对每一列都指定了名称和数据类型。数据类型在第 4 页的『数据类型』中进行讨论。`NOT NULL` 是可选的，可以指定它以表示列中不允许有空值。缺省值也是可选的。

可以在 `CREATE TABLE` 语句中指定许多其他选项，如唯一约束或参考约束。有关所有选项的详情，参见 *SQL Reference* 中的 `CREATE TABLE` 语句。

插入数据

当创建新表时，新表不包含任何数据。要将新的行输入表中，使用 `INSERT` 语句。此语句有两种通用格式：

- 通过第一种格式，您可使用 `VALUES` 子句来指定一行或多行的列值。以下三个示例均使用此通用格式将数据插入表中。
- 通过另一种格式，您可指定全查询而不是指定 `VALUES`，来标识其他表和 / 或视图中的行的列。

全查询是 `INSERT` 或 `CREATE VIEW` 语句中所使用的选择语句、或者是跟在谓词后面的选择语句。括在括号中的全查询通常称为子查询。

根据创建表时已选择的缺省选项，对于每个插入的行，为每一列提供一个值或者接受一个缺省值。各种数据类型的缺省值在 *SQL Reference* 中进行讨论。

下列语句使用 `VALUES` 子句将一行数据插入 `PERS` 表中：

```
INSERT INTO PERS
VALUES (12, 'Harris', 20, 'Sales', 5, 18000, 1000, '1950-1-1')
```

以下语句使用 `VALUES` 子句将三行插入 `PERS` 表中，此表中仅 `ID`、姓名以及工作是已知的。如果列定义为 `NOT NULL` 且没有缺省值，则必须为该列指定一个值。

`CREATE TABLE` 语句中的列定义上的 `NOT NULL` 子句可以用单词 `WITH DEFAULT` 扩充。如果某一列定义为 `NOT NULL WITH DEFAULT` 或常数缺省值（如 `WITH DEFAULT 10`），并且您未在列列表中指定该列，则缺省值插入至已插入行的该列中。例如，在 `CREATE TABLE` 语句中，仅为 `DEPT` 列指定了缺省值并将该值定义为 10。因此，部门号 (`DEPT`) 设置为 10，而任何其他未显式给定值的列都设置为空。

```
INSERT INTO PERS (NAME, JOB, ID)
VALUES ('Swagerman', 'Prgmr', 500),
       ('Limoges', 'Prgmr', 510),
       ('Li', 'Prgmr', 520)
```

下列语句返回插入的结果：

```
SELECT *
FROM PERS
```

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM	BIRTH_DATE
12	Harris	20	Sales	5	18000.00	1000.00	01/01/1950

500 Swagerman	10 Prgmr	-	-	- -
510 Limoges	10 Prgmr	-	-	- -
520 Li	10 Prgmr	-	-	- -

注意，在此情况下，并未给每个列指定值。空值显示为破折号 (-)。为此，列名列表的次序和数据类型都必须与 VALUES 子句中提供的值对应。如果省略列名列表（如第一个示例中那样），则 VALUES 之后的数据值列表的次序必须与它们所插入至的表中的列次序相同，值的数目必须等于表中列的数目。

每个值必须与它所插入至的列的数据类型相容。如果某列定义为可空，且未指定该列的值，则将空值赋给插入行中的该列。

因为未给行中的那些列指定值，所以下列示例将空值插入 YEARS、COMM 和 BIRTH_DATE 中。

```
INSERT INTO PERS (ID, NAME, JOB, DEPT, SALARY)
VALUES (410, 'Perna', 'Sales', 20, 20000)
```

INSERT 语句的第二种格式对于用另一表中的行的值来填充表非常方便。如上所述，指定全查询而不是指定 VALUES，以标识其他表和 / 或视图中的行中的列。

下列示例从 STAFF 表中选择部门 38 的成員的数据，并将它插入 PERS 表中：

```
INSERT INTO PERS (ID, NAME, DEPT, JOB, YEARS, SALARY)
SELECT ID, NAME, DEPT, JOB, YEARS, SALARY
FROM STAFF
WHERE DEPT = 38
```

在此插入之后，下列 SELECT 语句与 INSERT 语句中全查询产生的结果相同。

```
SELECT ID, NAME, DEPT, JOB, YEARS, SALARY
FROM PERS
WHERE DEPT = 38
```

结果为：

ID	NAME	DEPT	JOB	YEARS	SALARY
30	Marenghi	38	Mgr	5	17506.75
40	O'Brien	38	Sales	6	18006.00
60	Quigley	38	Sales	-	16808.30
120	Naughton	38	Clerk	-	12954.75
180	Abrahams	38	Clerk	3	12009.75

更改数据

使用 UPDATE 语句来更改表中的数据。使用此语句，可以更改满足 WHERE 子句搜索条件的每行中的一列或多列的值。

下列示例更新 ID 为 410 的雇员的信息：

```
UPDATE PERS
  SET JOB='Prgmr', SALARY = SALARY + 300
  WHERE ID = 410
```

SET 子句指定要更新的列并提供值。

WHERE 子句是可选的，它指定要更新的行。如果省略 WHERE 子句，则数据库管理程序用您提供的值更新表或视图中的每一行。

在此示例中，首先命名表 (PERS)，然后指定要更新行的条件。员工号码 410 的信息已更改：该雇员的工作职位更改为 Prgmr，工资增加了 \$300。

可以通过包括应用于两行或更多行的 WHERE 子句来更改多行数据。下列示例给每个销售员的工资增加 15%：

```
UPDATE PERS
  SET SALARY = SALARY * 1.15
  WHERE JOB = 'Sales'
```

删除数据

使用 DELETE 语句，根据在 WHERE 子句中指定的搜索条件从表中删除数据行。下列示例删除其中雇员 ID 为 120 的行：

```
DELETE FROM PERS
  WHERE ID = 120
```

WHERE 子句是可选的，它指定要删除的行。如果省略 WHERE 子句，则数据库管理程序删除表或视图中的所有行。

可以使用 DELETE 语句删除多行。下列示例删除其中雇员部门 (DEPT) 为 20 的所有行：

```
DELETE FROM PERS
  WHERE DEPT = 20
```

当删除某一行时，是除去整个行，而不是除去该行中的特定列值。

要删除表的定义及其内容，可发出 `DROP TABLE` 语句，如 *SQL Reference* 中所述。

创建视图

如在第3页的『视图』中所讨论的那样，视图提供在一个或多个表中查看数据的替代方法。通过创建视图，可以对想让各种用户查看的信息进行限制。下列图表显示视图和表之间的关系。

在图2中，`View_A` 被限制为只可存取 `TABLE_A` 的列 `AC1` 和 `AC2`。

`View_AB` 允许存取 `TABLE_A` 中的列 `AC3` 和 `TABLE_B` 中的列 `BC2`。

通过创建 `View_A`，将用户可以具有的存取权限限制为 `TABLE_A`，而通过创建 `VIEW_AB`，则是将存取权限限制为两个表中的某些列。

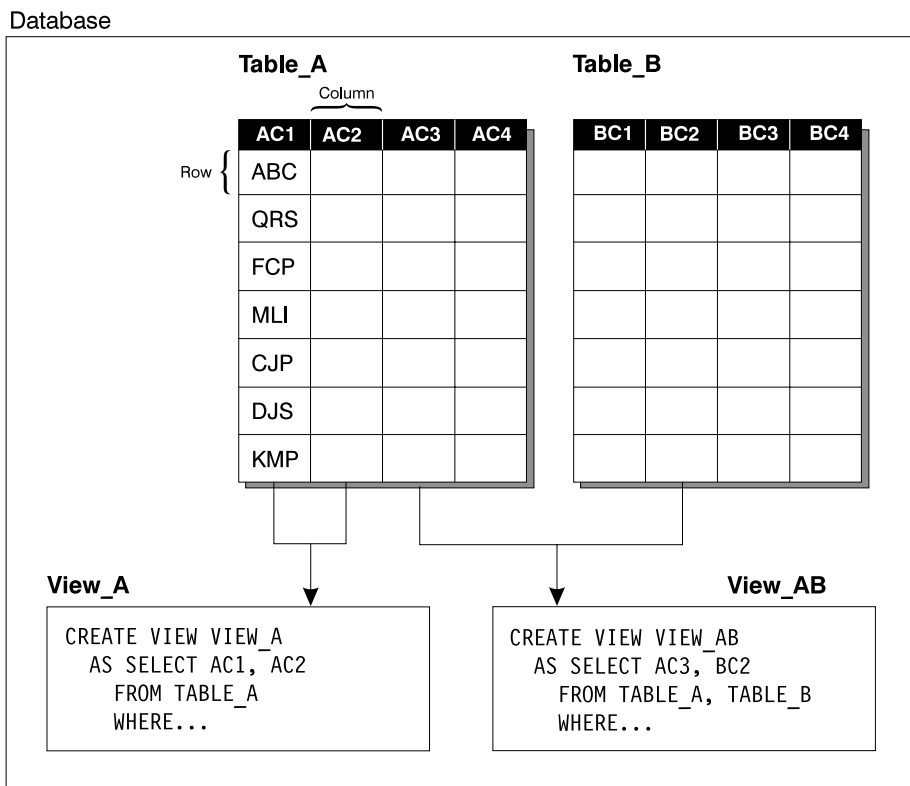


图 2. 表和视图之间的关系

下列语句创建 STAFF 表中部门 20 内的非经理人员的视图，其中工资和佣金不通过基表显示。

```
CREATE VIEW STAFF_ONLY
AS SELECT ID, NAME, DEPT, JOB, YEARS
FROM STAFF
WHERE JOB <> 'Mgr' AND DEPT=20
```

在创建视图之后，下列语句显示视图的内容：

```
SELECT *
FROM STAFF_ONLY
```

此语句产生下列结果：

ID	NAME	DEPT	JOB	YEARS
20	Pernal	20	Sales	8
80	James	20	Clerk	-
190	Sneider	20	Clerk	8

作为一个更深入的示例，我们可使用 STAFF 和 ORG 表来创建一个列出每个部门名称和部门经理姓名的视图。以下语句创建此视图：

```
CREATE VIEW DEPARTMENT_MGRS
AS SELECT NAME, DEPTNAME
FROM STAFF, ORG
WHERE MANAGER = ID
```

创建视图时，可以使用 WITH CHECK OPTION 子句，将附加约束添加到通过视图对表进行的插入和更新。此子句导致数据库管理程序验证对视图的任何更新或插入是否符合该视图的定义，并拒绝那些不符合定义的更新或插入。如果省略此子句，则不检查违反视图定义的插入和更新。有关 WITH CHECK OPTION 如何起作用的详情，参考 *SQL Reference* 中的 CREATE VIEW 语句。

使用视图来处理数据

象 SELECT 语句一样，INSERT、DELETE 以及 UPDATE 语句应用于视图，就好象视图是一个实表那样。这些语句处理基本基表中的数据。因此当再次存取该视图时，使用最新的基表对它进行计算。如果不使用 WITH CHECK OPTION 子句，则使用视图修改的数据可能不在视图的重复存取中出现，原因是该数据可能不再符合原来的视图定义。

以下示例将更新应用于视图 FIXED_INCOME：

```

CREATE VIEW FIXED_INCOME (LNAME, DEPART, JOBTITLE, NEWSALARY)
AS SELECT NAME, DEPT, JOB, SALARY
   FROM PERS
   WHERE JOB <> 'Sales' WITH CHECK OPTION

UPDATE FIXED_INCOME
SET NEWSALARY = SALARY * 1.10
WHERE LNAME = 'Li'

```

除了校验选项以外，先前视图中的更新等效于对基表 PERS 的更新：

```

UPDATE PERS
SET SALARY = SALARY * 1.10
WHERE NAME = 'Li'
AND JOB <> 'Sales'

```

注意，由于视图是在 CREATE VIEW FIXED_INCOME 中对约束 JOB <> 'Sales' 使用 WITH CHECK OPTION 创建的，所以当 Limoges 调去做销售时不允许下列更新：

```

UPDATE FIXED_INCOME
SET JOBTITLE = 'Sales'
WHERE LNAME = 'Limoges'

```

不能更新由表达式 SALARY + COMM 或 SALARY * 1.25 定义的列。如果定义一个包含一个或多个这样的列的视图，则拥有者不具有对这些列的 UPDATE 特权。在包含这样的列的视图上不允许 INSERT 语句，但允许 DELETE 语句。

现在我们讨论一个没有任何一列定义为 NOT NULL 的表 PERS。可以通过 FIXED_INCOME 视图将行插入表 PERS 中，即使该视图不包含基本表 PERS 的 ID、YEARS、COMM 或 BIRTHDATE。整个视图中看不到的列被适当地设置为空值或缺省值。

但是表 PERS 确实已将列 ID 定义为 NOT NULL。如果尝试通过 FIXED_INCOME 视图插入行，则系统试图将空值插入在整个视图中“看不到”的所有 PERS 列。由于 ID 列未包括在视图中并且该列不允许空值，所以系统不允许通过该视图进行插入。

有关修改视图的规则和限制，参考 *SQL Reference* 中的 CREATE VIEW 语句。

第4章 使用 SQL 语句存取数据

本节讲述如何使用 SQL 语句来连接数据库和检索数据。

在示例中，我们给出了要输入的语句，（大多数情况下）后面跟着在对样本数据库发出该语句时将显示的结果。注意，虽然我们用大写字母显示语句，但也可以用大小写字母混合来输入这些语句（用单引号（'）或双引号（''）括起来的语句除外）。

DB2 通用数据库包括的 SAMPLE 数据库由几个表组成，如第71页的『附录A. 样本数据库表』中所列。可使用“第一步骤”安装启动板来创建此数据库。还可从命令行创建 SAMPLE 数据库。参见 *SQL Reference* 以了解详情。

注意，DB2 通用数据库包含其它样本数据库，它们演示“数据仓库中心”和“OLAP Starter 工具箱”功能。此书中的示例仅使用一般 SAMPLE 数据库。

视数据库设置方式的不同，您可能必须通过给所使用的表名加上模式名和一个句点作为前缀来限定这些表名。对于本书中的示例，假定缺省模式为 USERID。所以可以将表 ORG 引用为 USERID.ORG。可询问管理员这样做是否有必要。

本章包括：

- 连接数据库
- 调查错误
- 选择列和选择行
- 将行进行排序和除去重复行
- 运算次序
- 使用表达式来计算值
- 给表达式命名
- 从多个表中选择数据
- 使用子查询
- 使用函数
- 分组

连接数据库

您必须先与数据库连接，才能使用 SQL 语句来查询或操作该数据库。CONNECT 语句使数据库连接与用户名相关联。

例如，要连接 SAMPLE 数据库，在 DB2 命令行处理器中输入下列命令：

```
CONNECT TO SAMPLE USER USERID USING PASSWORD
```

（确保选择的是在服务器系统上有效的用户 ID 和口令）。

在此示例中，USER 的值为 USERID，USING 的值为 PASSWORD。

下列信息告诉您连接成功：

数据库连接信息

数据库产品	= DB2/NT 7.1.0
SQL 权限 ID	= USERID
本地数据库别名	= SAMPLE

一旦连接上，就可以开始操作数据库。有关连接的详情，参考 *SQL Reference* 中的 CONNECT 语句。

调查错误

每当在任何示例中出现输入错误时，或者在执行 SQL 语句期间出错时，数据库管理程序会返回错误信息。错误信息由信息标识符、简要说明以及 SQLSTATE 组成。

SQLSTATE 错误是 DB2 系列产品的公共错误码。SQLSTATE 错误符合 ISO/ANSI SQL92 标准。

例如，如果在 CONNECT 语句中用户 ID 或口令不正确，则数据库管理程序将返回信息标识符 SQL1403N 以及 SQLSTATE 08004。该信息如下：

```
SQL1403N 提供的用户名和 / 或口令不正确。 SQLSTATE=08004
```

可以通过输入一个问号 (?)，然后输入信息标识符或 SQLSTATE 来获取关于错误信息的详情：

```
? SQL1403N  
或  
? SQL1403  
或  
? 08004
```

注意，错误 SQL1403N 的说明中倒数第二行表明 SQLCODE 为 -1403。SQLCODE 为产品特定的错误码。以 N（通知）或 C（严重）结尾的信息标识符表示一个错误，并且具有负 SQLCODE。以 W（警告）结尾的信息标识符表示一个警告，并且具有正 SQLCODE。

选择列

使用 SELECT 语句从表中选择特定的列。在该语句中指定用逗号分隔的列名列表。此列表称为选择列表。

下列语句从 SAMPLE 数据库的 ORG 表中选择部门名称 (DEPTNAME) 和部门号 (DEPTNUMB):

```
SELECT DEPTNAME, DEPTNUMB
FROM ORG
```

以上语句产生下列结果:

DEPTNAME	DEPTNUMB
Head Office	10
New England	15
Mid Atlantic	20
South Atlantic	38
Great Lakes	42
Plains	51
Pacific	66
Mountain	84

通过使用星号 (*) 可从表中选择所有列。下一个示例列出了 ORG 表中的所有的列和行:

```
SELECT *
FROM ORG
```

此语句产生下列结果:

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
10	Head Office	160	Corporate	New York
15	New England	50	Eastern	Boston
20	Mid Atlantic	10	Eastern	Washington
38	South Atlantic	30	Eastern	Atlanta
42	Great Lakes	100	Midwest	Chicago
51	Plains	140	Midwest	Dallas
66	Pacific	270	Western	San Francisco
84	Mountain	290	Western	Denver

选择行

要从表中选择特定行，在 **SELECT** 语句之后使用 **WHERE** 子句指定要选择的行必须满足的条件。从表中选择行的标准是搜索条件。

搜索条件由一个或多个谓词组成。谓词指定关于某一行是真或是假（或未知）的条件。可使用下列基本谓词在 **WHERE** 子句中指定条件：

谓词	功能
$x = y$	x 等于 y
$x \neq y$	x 不等于 y
$x < y$	x 小于 y
$x > y$	x 大于 y
$x \leq y$	x 小于或等于 y
$x \geq y$	x 大于或等于 y
IS NULL/IS NOT NULL	测试空值

在构造搜索条件时，要注意只对数字数据类型执行算术运算，并只在相容数据类型之间进行比较。例如，不能将文本字符串与数字进行比较。

如果正在根据字符值来选择行，则该值必须用单引号括起来（例如，**WHERE JOB = 'Clerk'**），并且输入的每个字符值必须与数据库中的完全一样。如果数据值在数据库中是小写的，而您用大写形式来输入它，将不选择行。如果正在根据数字值来选择行，该值不能用引号括起来（例如，**WHERE DEPT = 20**）。

下列示例只从 **STAFF** 表中选择部门 20 的行：

```
SELECT DEPT, NAME, JOB
FROM STAFF
WHERE DEPT = 20
```

此语句产生下列结果：

```
DEPT  NAME      JOB
-----
    20 Sanders  Mgr
    20 Pernal  Sales
    20 James  Clerk
    20 Sneider Clerk
```

下一个示例使用 **AND** 来指定多个条件。可以指定任意多个条件。该示例从 **STAFF** 表中选择部门 20 中的职员：

```

SELECT DEPT, NAME, JOB
FROM STAFF
WHERE JOB = 'Clerk'
AND DEPT = 20

```

此语句产生下列结果:

DEPT	NAME	JOB
20	James	Clerk
20	Sneider	Clerk

未在其中输入值且不支持缺省值的列中出现空值。将值特别设置为空值的地方也可以出现空值。空值只能在定义为支持空值的列中出现。第9页的『创建表』中讨论了在表中定义和支持空值。

使用谓词 `IS NULL` 和 `IS NOT NULL` 来检查空值。

下列语句列出了佣金未知的雇员:

```

SELECT ID, NAME
FROM STAFF
WHERE COMM IS NULL

```

此语句产生下列结果:

ID	NAME
10	Sanders
30	Marenghi
50	Hanes
100	Plotz
140	Fraye
160	Molinare
210	Lu
240	Daniels
260	Jones
270	Lea
290	Quill

值零与空值不相同。下列语句选择表中佣金为零的每个人:

```

SELECT ID, NAME
FROM STAFF
WHERE COMM = 0

```

因为样本表中的 `COMM` 列中没有零值，所以返回的结果集为空。

下一个示例选择 `STAFF` 表中 `YEARS` 的值大于 9 的所有行:

```
SELECT NAME, SALARY, YEARS
FROM STAFF
WHERE YEARS > 9
```

此语句产生下列结果:

NAME	SALARY	YEARS
Hanes	20659.80	10
Lu	20010.00	10
Jones	21234.00	12
Quill	19818.00	10
Graham	21000.00	13

将行进行排序

您可能想要信息按特定次序返回。使用 **ORDER BY** 子句将信息按一个或多个列中的值进行排序。

下列语句显示部门 84 中按雇用年数排序的雇员:

```
SELECT NAME, JOB, YEARS
FROM STAFF
WHERE DEPT = 84
ORDER BY YEARS
```

此语句产生下列结果:

NAME	JOB	YEARS
Davis	Sales	5
Gafney	Clerk	5
Edwards	Sales	7
Quill	Mgr	10

指定 **ORDER BY** 作为整个 **SELECT** 语句中的最后一个子句。在此子句中命名的列可以是表达式或表的任何列。**ORDER BY** 子句中的列名不必在选择列表中指定。

可通过在 **ORDER BY** 子句中显式指定 **ASC** 或 **DESC** 将行按升序或降序进行排序。如果既未指定 **ASC**，也未指定 **DESC**，则自动按升序将行进行排序。下列语句按雇用年数以降序显示部门 84 中的雇员:

```
SELECT NAME, JOB, YEARS
FROM STAFF
WHERE DEPT = 84
ORDER BY YEARS DESC
```

此语句产生下列结果:

NAME	JOB	YEARS
Quill	Mgr	10
Edwards	Sales	7
Davis	Sales	5
Gafney	Clerk	5

可以按字符值以及数字值将行进行排序。下列语句按姓名字母顺序显示部门 84 的雇员:

```
SELECT NAME, JOB, YEARS
FROM STAFF
WHERE DEPT = 84
ORDER BY NAME
```

此语句产生下列结果:

NAME	JOB	YEARS
Davis	Sales	5
Edwards	Sales	7
Gafney	Clerk	5
Quill	Mgr	10

除去重复行

当使用 `SELECT` 语句时, 您可能不想要返回重复信息。例如, `STAFF` 有一个其中多次列出了几个部门号的 `DEPT` 列, 以及一个其中多次列出了几个工作说明的 `JOB` 列。

要消除重复行, 在 `SELECT` 子句上使用 `DISTINCT` 选项。例如, 如果将 `DISTINCT` 插入该语句, 则部门中的每项工作仅列出一次:

```
SELECT DISTINCT DEPT, JOB
FROM STAFF
WHERE DEPT < 30
ORDER BY DEPT, JOB
```

此语句产生下列结果:

DEPT	JOB
10	Mgr
15	Clerk
15	Mgr
15	Sales
20	Clerk
20	Mgr
20	Sales

DISTINCT 已消除了在选择语句中指定的一组列中所有包含重复数据的行。

运算次序

考虑运算次序是很重要的。一个子句的输出是下一个子句的输入，如下面列表中所所述。第25页的『给表达式命名』中给出一个要考虑其中运算次序的示例。

下列运算顺序不一定是 DB2 代码内执行运算的方法。这个简单解释仅允许以更直观的方式考虑查询。运算顺序如下：

1. FROM 子句
2. WHERE 子句
3. GROUP BY 子句
4. HAVING 子句
5. SELECT 子句
6. ORDER BY 子句

使用表达式来计算值

表达式是包括在语句中的计算或函数。下列语句计算，如果部门 38 中每个雇员都收到 \$500 的奖金，则每人的工资将是多少：

```
SELECT DEPT, NAME, SALARY + 500
FROM STAFF
WHERE DEPT = 38
ORDER BY 3
```

此结果为：

DEPT	NAME	3
38	Abrahams	12509.75
38	Naughton	13454.75
38	Quigley	17308.30
38	Marenghi	18006.75
38	O'Brien	18506.00

注意第三列的列名是一个数字。这是一个系统生成的数字，因为 SALARY+500 未指定列名。以后此数字在 ORDER BY 子句中用来表示第三列。第25页的『给表达式命名』论及如何给表达式取有意义的名称。

可使用基本算术运算符加 (+)、减 (-)、乘 (*) 和除 (/) 来构成算术表达式。

这些运算符可对几种不同类型操作数的数值进行运算，其中某些操作数为：

- 列名（例如在 `RATE*HOURS` 中）
- 常数值（例如在 `RATE*1.07` 中）
- 标量函数（例如在 `LENGTH(NAME) + 1` 中）。

给表达式命名

可选的 `AS` 子句允许您给表达式指定有意义的名称，这就使得以后再引用该表达式更容易。可使用 `AS` 子句为选择列表中的任何项提供名称。

下列语句显示其工资加佣金少于 \$13,000 的所有雇员。表达式 `SALARY + COMM` 命名为 `PAY`：

```
SELECT NAME, JOB, SALARY + COMM AS PAY
FROM STAFF
WHERE (SALARY + COMM) < 13000
ORDER BY PAY
```

此语句产生下列结果：

NAME	JOB	PAY
Yamaguchi	Clerk	10581.50
Burke	Clerk	11043.50
Scoutten	Clerk	11592.80
Abrahams	Clerk	12246.25
Kermisch	Clerk	12368.60
Ngan	Clerk	12714.80

通过使用 `AS` 子句，可以在 `ORDER BY` 子句中引用特定的列名而不是系统生成的数字。在此示例中，我们在 `WHERE` 子句中将 `(SALARY + COMM)` 与 13000 进行比较，而不是使用名称 `PAY`。这是运算次序的结果。在给定 `(SALARY + COMM)` 名称 `PAY` 之前计算 `WHERE` 子句的值，原因是 `SELECT` 子句在 `WHERE` 子句后执行。因此，不能在该谓词中使用 `PAY`。

从多个表中选择数据

可使用 `SELECT` 语句从两个或多个表中生成包含信息的报告。这通常称为连接。例如，可以连接 `STAFF` 和 `ORG` 表中的数据以形成一个新表。要连接两个表，在 `SELECT` 子句中指定想要显示的列，在 `FROM` 子句中指定表名，在 `WHERE` 子句中指定搜索条件。`WHERE` 子句是可选的。

下一个示例使每个经理的姓名与部门名称关联。需要从两个表中选择信息，因为雇员信息（`STAFF` 表）和部门信息（`ORG` 表）是独立存储的。下列查询分别选择 `STAFF` 和 `ORG` 表的 `NAME` 和 `DEPTNAME` 列。搜索条件将选择范围缩小为 `MANAGER` 列中的值与 `ID` 列中的值相同的行：

```

SELECT DEPTNAME, NAME
FROM ORG, STAFF
WHERE MANAGER = ID

```

图3 演示如何比较两个不同表中的列。加框线的值表示满足搜索条件的匹配项。

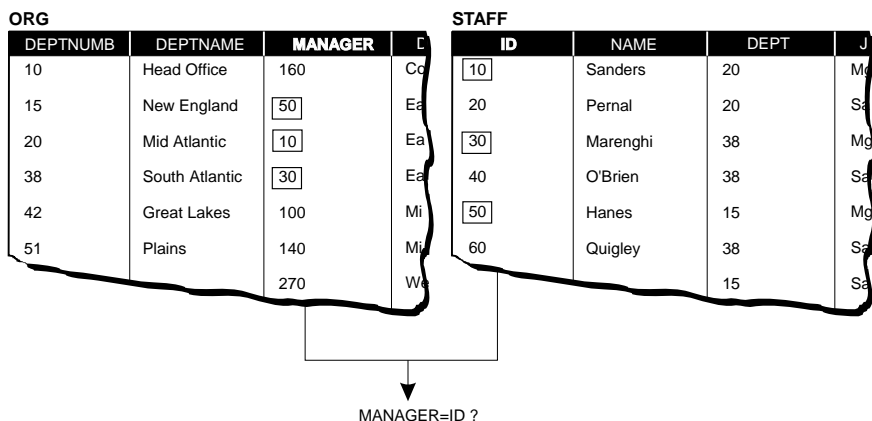


图3. 从 STAFF 和 ORG 表中选择

SELECT 语句产生下列结果:

```

DEPTNAME      NAME
-----
Mid Atlantic   Sanders
South Atlantic Marenghi
New England   Hanes
Great Lakes   Plotz
Plains        Fraye
Head Office   Molinare
Pacific       Lea
Mountain      Quill

```

该结果列出了每个经理的姓名和他或她的部门。

使用子查询

在编写 SQL SELECT 语句时，可在 WHERE 子句中放置附加的 SELECT 语句。每个附加的 SELECT 启动一个子查询。

然后，子查询本身又可包括其结果代入原始查询的 WHERE 子句的另一子查询。另外，WHERE 子句可将子查询包括在多个搜索条件中。子查询可引用与主查询中所使用的不同的表和列。

下列语句从 `ORG` 表中选择 `STAFF` 表中其 `ID` 为 280 的雇员的分部和位置:

```
SELECT DIVISION, LOCATION
FROM ORG
WHERE DEPTNUMB = (SELECT DEPT
                  FROM STAFF
                  WHERE ID = 280)
```

在处理语句时, `DB2` 首先确定子查询的结果。因为 `ID` 为 280 的雇员在部门 66, 所以此示例的子查询的结果为 66。然后, 最终结果从其 `DEPTNUMB` 列具有值 66 的 `ORG` 表的行中得出。最终结果是:

```
DIVISION    LOCATION
-----
Western     San Francisco
```

当使用子查询时, 数据库管理程序计算该子查询并将结果值直接代入 `WHERE` 子句。

在第38页的『相关子查询』中进一步讨论了子查询。

使用函数

本节简要介绍了将用于全书示例的函数。数据库函数是一组输入数据值和一个结果值之间的关系。

函数可以是内部的或用户定义的。`DB2` 通用数据库提供很多内部函数和预安装的用户定义函数。

可找到 `SYSIBM` 模式的内部函数, 而预安装的用户定义函数为 `SYSFUN` 模式。`SYSIBM` 和 `SYSFUN` 是保留模式。

内部函数和预安装的用户定义函数始终不能满足所有的用户需求。因此, 应用程序开发者可能需要创建自己的一套专门针对他们的应用程序的函数。用户定义函数使这成为可能, 例如通过扩展 `DB2` 通用数据库的范围以包括定制的商业或科学函数。这在第66页的『用户定义函数』中进一步讨论。

列函数

列函数对列中的一组值进行运算以得到单个结果值。下列就是一些列函数的示例。要获取完整列表, 参考 *SQL Reference*。

AVG	返回某一组中的值除以该组中值的个数的和
COUNT	返回一组行或值中行或值的个数
MAX	返回一组值中的最大值

MIN 返回一组值中的最小值

下列语句从 STAFF 表中选择最高工资:

```
SELECT MAX(SALARY)
FROM STAFF
```

此语句从 STAFF 样本表中返回值 22959.20。

下一个示例选择其收入超过平均收入但在公司的年数少于平均年数的雇员姓名和工资。

```
SELECT NAME, SALARY
FROM STAFF
WHERE SALARY > (SELECT AVG(SALARY) FROM STAFF)
AND YEARS < (SELECT AVG(YEARS) FROM STAFF)
```

此语句产生下列结果:

NAME	SALARY
Marenghi	17506.75
Daniels	19260.25
Gonzales	16858.20

在以上示例的 WHERE 子句中, 在子查询中说明了列函数, 而不是直接实现列函数 (例如: WHERE SALARY > AVG(SALARY))。不能在 WHERE 子句中说明列函数。这是由运算次序导致的结果。将 WHERE 子句考虑为在 SELECT 子句之前进行计算。因此, 当正在计算 WHERE 子句时, 列函数没有对该组值的存取权。稍后由 SELECT 子句选择这组值。

可使用 DISTINCT 元素作为列函数自变量的一部分, 以在应用函数之前消除重复值。因此, COUNT(DISTINCT WORKDEPT) 计算不同部门的个数。

标量函数

标量函数对一个单一值进行某个运算以返回另一个单一值。下列就是一些由 DB2 通用数据库提供的标量函数的示例。

ABS 返回数的绝对值

HEX 返回值的十六进制表示

LENGTH 返回自变量中的字节数 (对于图形字符串则返回双字节字符数。)

YEAR 抽取日期时间值的年份部分

有关标量函数的详细列表和说明, 参考 *SQL Reference*。

下列语句返回 ORG 表中的部门名称以及其每个名称的长度:

```
SELECT DEPTNAME, LENGTH(DEPTNAME)
FROM ORG
```

此语句产生下列结果:

DEPTNAME	2
Head Office	11
New England	11
Mid Atlantic	12
South Atlantic	14
Great Lakes	11
Plains	6
Pacific	7
Mountain	8

注意, 由于未使用 AS 子句给 LENGTH(DEPTNAME) 取一个有意义的名称, 所以第二列中出现系统生成的数字。

表函数

表函数返回表的列, 类似于由单一 CREATE TABLE 语句创建的表。

表函数仅可用于 SQL 语句的 FROM 子句。

DB2 通用数据库中当前唯一支持的表函数是 SQLCACHE_SNAPSHOT。

SQLCACHE_SNAPSHOT

将 DB2 动态 SQL 语句高速缓存的瞬象的结果作为表返回。

分组

DB2 通用数据库具有基于表的特定列对数据进行分析的能力。

可根据 GROUP BY 子句中定义的分组结构来组织行。最简单的格式为, 一个组就是一组行, 每一组在 "GROUP BY" 列中都具有完全相同的值。SELECT 子句中的列名必须为分组列或列函数。列函数对 GROUP BY 子句定义的每个组返回一个值。每一组由结果集中的单一行表示。下列示例产生一个列出了每个部门号的最高工资的结果:

```
SELECT DEPT, MAX(SALARY) AS MAXIMUM
FROM STAFF
GROUP BY DEPT
```

此语句产生下列结果:

DEPT	MAXIMUM
10	22959.20
15	20659.80
20	18357.50
38	18006.00
42	18352.80
51	21150.00
66	21000.00
84	19818.00

注意，计算的是每个部门（由 `GROUP BY` 子句定义的组）而不是整个公司的 `MAX(SALARY)`。

将 `WHERE` 子句与 `GROUP BY` 子句一起使用

分组查询可以在形成组和计算列函数之前具有消除非限定行的标准 `WHERE` 子句。必须在 `GROUP BY` 子句之前指定 `WHERE` 子句。例如：

```
SELECT WORKDEPT, EDLEVEL, MAX(SALARY) AS MAXIMUM
FROM EMPLOYEE
WHERE HIREDATE > '1979-01-01'
GROUP BY WORKDEPT, EDLEVEL
ORDER BY WORKDEPT, EDLEVEL
```

结果为：

WORKDEPT	EDLEVEL	MAXIMUM
D11	17	18270.00
D21	15	27380.00
D21	16	36170.00
D21	17	28760.00
E11	12	15340.00
E21	14	26150.00

注意，在 `SELECT` 语句中指定的每个列名也在 `GROUP BY` 子句中提到。未在这两个地方提到的列名将产生错误。`GROUP BY` 子句对 `WORKDEPT` 和 `EDLEVEL` 的每个唯一组合各返回一行。

在 `GROUP BY` 子句之后使用 `HAVING` 子句

可将限定条件应用于各个组，以便 `DB2` 仅对满足条件的组返回结果。为此，在 `GROUP BY` 子句后面包含一个 `HAVING` 子句。`HAVING` 子句可包含一个或多个用 `AND` 和 `OR` 连接的谓词。每个谓词将组特性（如 `AVG(SALARY)`）与下列之一进行比较：

- 该组的另一个特性

例如：

```
HAVING AVG(SALARY) > 2 * MIN(SALARY)
```

- 常数

例如:

```
HAVING AVG(SALARY) > 20000
```

例如, 下列查询查找雇员数超过 4 的部门的最高和最低工资:

```
SELECT WORKDEPT, MAX(SALARY) AS MAXIMUM, MIN(SALARY) AS MINIMUM
FROM EMPLOYEE
GROUP BY WORKDEPT
HAVING COUNT(*) > 4
ORDER BY WORKDEPT
```

此语句产生下列结果:

WORKDEPT	MAXIMUM	MINIMUM
D11	32250.00	18270.00
D21	36170.00	17250.00
E11	29750.00	15340.00

有可能 (虽然很少见) 查询有 **HAVING** 子句但没有 **GROUP BY** 子句。在此情况下, **DB2** 将整个表看作一个组。因为该表被看作是单个组, 所以最多可以有一个结果行。如果 **HAVING** 条件对整个表为真, 则返回选择的结果 (该结果必须整个由列函数组成); 否则不返回任何行。

第5章 表达式和子查询

DB2 在表示查询方面提供了灵活性。本章描述一些可用于表示复杂查询的重要方法。

本章给出下列各项的综合说明:

- 标量全查询
- 转换数据类型
- 条件表达式
- 表表达式
- 相关名

标量全查询

全查询是可在 SQL 语句中使用的查询格式。标量全查询是一个全查询，它返回一行（该行只包含一个值）。标量全查询对从数据库中检索数据值供表达式使用是很有用的。

- 以下示例列出了工资超过全部雇员平均工资的雇员的姓名。查询中的标量全查询是用括号括起来的选择语句。

```
SELECT LASTNAME, FIRSTNAME
       FROM EMPLOYEE
      WHERE SALARY > (SELECT AVG(SALARY)
                     FROM EMPLOYEE)
```

- 此示例查找 STAFF 表中雇员的平均工资和 EMPLOYEE 表中雇员的平均工资。

```
SELECT AVG(SALARY) AS "Average_Employee",
       (SELECT AVG(SALARY) AS "Average_Staff" FROM STAFF)
       FROM EMPLOYEE
```

转换数据类型

有时可能需要将一些值从一种数据类型转换成另一种数据类型，例如，从数字值转换成字符串。要将一个值转换成另一个不同的类型，使用 CAST 说明。

转换说明的另一个可能用途是截断很长的字符串。在 EMP_RESUME 表中，RESUME 列是 CLOB(5K)。您可能只想显示包含应聘者个人信息的前 370 个字符。要从 EMP_RESUME 表中显示简历的 ASCII 格式的前 370 个字符，发出下列请求：

```
SELECT EMPNO, CAST(RESUME AS VARCHAR(370))
FROM EMP_RESUME
WHERE RESUME_FORMAT = 'ascii'
```

会发出一个警告，通知您超过 370 个字符的值被截断。

可将空值转换为更便于在查询中进行处理的其他数据类型。第36页的『公共表表达式』是一个为此使用转换的示例。

条件表达式

可在 SQL 语句中使用 CASE 表达式以便于处理表的数据表示。这提供了一种功能强大的条件表达式能力，在概念上与某些程序设计语言中的 CASE 语句类似。

- 要从 ORG 表中的 DEPTNAME 列将部门号更改为有意义的字，输入下列查询：

```
SELECT DEPTNAME,
CASE DEPTNUMB
WHEN 10 THEN 'Marketing'
WHEN 15 THEN 'Research'
WHEN 20 THEN 'Development'
WHEN 38 THEN 'Accounting'
ELSE 'Sales'
END AS FUNCTION
FROM ORG
```

结果为：

DEPTNAME	FUNCTION
Head Office	Marketing
New England	Research
Mid Atlantic	Development
South Atlantic	Accounting
Great Lakes	Sales
Plains	Sales
Pacific	Sales
Mountain	Sales

- 可使用 CASE 表达式来防止出现异常情况，如被零除等。在下列示例中，如果雇员没有奖金或佣金报酬，则语句条件通过避免除法运算来防止出错：

```

SELECT LASTNAME, WORKDEPT FROM EMPLOYEE
WHERE(CASE
      WHEN BONUS+COMM=0 THEN NULL
      ELSE SALARY/(BONUS+COMM)
      END ) > 10

```

- 可使用 CASE 表达式来产生一个比率，该比率为一系列中值的子集的和对该列的所有值的和的比率。此比率可包含在使用 CASE 表达式的单一语句中；此表达式只需要传送数据一次。在没有 CASE 表达式的情况下，执行同样的计算至少需要传送两次。

下列示例使用 CASE 表达式计算部门 20 的工资之和与全部工资总额的比率：

```

SELECT CAST(CAST (SUM(CASE
                  WHEN DEPT = 20 THEN SALARY
                  ELSE 0
                  END) AS DECIMAL(7,2))/
          SUM(SALARY) AS DECIMAL (3,2))
FROM STAFF

```

结果为 0.11。注意，CAST 函数确保结果的精度得到保持。

- 可使用 CASE 表达式来计算简单的函数，而不必调用函数本身，调用函数将需要额外开销。例如：

```

CASE
  WHEN X<0 THEN -1
  WHEN X=0 THEN 0
  WHEN X>0 THEN 1
END

```

此表达式与 SYSFUN 模式中 SIGN 用户定义函数有相同的结果。

表表达式

如果只需要单个查询的视图定义，可使用表表达式。

表表达式是临时的，只在 SQL 语句的有效期内有效；表表达式不能象视图那样共享，但它们比视图有更大灵活性。

本节描述如何在查询中使用公共表表达式和嵌套表表达式。

嵌套表表达式

嵌套表表达式是一个临时视图，其中的定义被嵌套（直接定义）在主查询的 FROM 子句中。

下列查询使用嵌套表表达式来查找那些教育级别超过 16 的雇员的平均总收入、教育级别以及雇用年份：

```

SELECT EDLEVEL, HIREYEAR, DECIMAL(AVG(TOTAL_PAY),7,2)
FROM (SELECT EDLEVEL, YEAR(HIREDATE) AS HIREYEAR,
SALARY+BONUS+COMM AS TOTAL_PAY
FROM EMPLOYEE
WHERE EDLEVEL > 16) AS PAY_LEVEL
GROUP BY EDLEVEL, HIREYEAR
ORDER BY EDLEVEL, HIREYEAR

```

结果如下:

EDLEVEL	HIREYEAR	3
17	1967	28850.00
17	1973	23547.00
17	1977	24430.00
17	1979	25896.50
18	1965	57970.00
18	1968	32827.00
18	1973	45350.00
18	1976	31294.00
19	1958	51120.00
20	1975	42110.00

此查询使用嵌套表表达式，首先从 `HIREDATE` 列中抽取雇用年份，以便可以在后续的 `GROUP BY` 子句中使用该雇用年份。您可能不想将这作为视图来创建（如果您打算用不同的 `EDLEVEL` 值来执行相似查询的话）。

此示例中使用了标量内部函数 `DECIMAL`。`DECIMAL` 返回数字或字符串的十进制表示。有关函数的详情，参考 *SQL Reference*。

公共表表达式

公共表表达式是您创建来供在复杂查询中使用的表表达式。在查询的开头使用 `WITH` 子句定义并命名公共表表达式。对公共表表达式的重复引用使用同一个结果集。相比之下，如果使用嵌套表表达式或视图，则每次都将重新生成结果集，其结果可能各不相同。

以下示例列出公司中教育级别大于 16、平均工资比同年雇用的且有同样教育级别的人低的所有人员。在该查询后面会更详细地描述查询的各个部分。

1

```

WITH
PAYLEVEL AS
(SELECT EMPNO, EDLEVEL, YEAR(HIREDATE) AS HIREYEAR,
SALARY+BONUS+COMM AS TOTAL_PAY
FROM EMPLOYEE
WHERE EDLEVEL > 16),

```

2

```
PAYBYED (EDUC_LEVEL, YEAR_OF_HIRE, AVG_TOTAL_PAY) AS
(SELECT EDLEVEL, HIREFYEAR, AVG(TOTAL_PAY)
FROM PAYLEVEL
GROUP BY EDLEVEL, HIREFYEAR)
```

3

```
SELECT EMPNO, EDLEVEL, YEAR_OF_HIRE, TOTAL_PAY, DECIMAL(AVG_TOTAL_PAY,7,2)
FROM PAYLEVEL, PAYBYED
WHERE EDLEVEL = EDUC_LEVEL
AND HIREFYEAR = YEAR_OF_HIRE
AND TOTAL_PAY < AVG_TOTAL_PAY
```

1

这是名为 PAYLEVEL 的公共表表达式。此结果表包括雇员号、雇用该雇员的年份、该雇员的总收入以及他（或她）的教育级别。只包括雇员的教育级别大于 16 的那些行。

2

这是名为 PAYBYED（或 PAY BY EDation）的公共表表达式。该表达式使用在前一个公共表表达式中创建的 PAYLEVEL 表来确定每个教育级别同一年雇用的雇员的教育级别、雇用年份以及平均收入。此表返回的列被赋予的名称与选择列表中所使用的列名不同（如 EDUC_LEVEL）。这会产生名为 PAYBYED 的结果集，它与“嵌套表表达式”示例中所产生的结果相同。

3

最后，我们获得能产生期望结果的实际查询。连接这两个表（PAYLEVEL, PAYBYED）以确定总收入比同年雇用的人的平均收入低的那些人。注意，PAYBYED 是以 PAYLEVEL 为基础的。所以在完整语句中有效地存取了 PAYLEVEL 两次。两次都使用同一组行来计算查询。

最终结果如下：

EMPNO	EDLEVEL	YEAR_OF_HIRE	TOTAL_PAY	5
000210	17	1979	20132.00	25896.50

相关名

相关名是用于识别一个对象的多种用途的标识符。可在查询的 FROM 子句中和 UPDATE 或 DELETE 语句的第一个子句中定义相关名。相关名可与表、视图或嵌套表表达式关联，但只限于定义相关名的上下文中。

例如，子句 FROM STAFF S、ORG O 分别指定 S 和 O 作为 STAFF 和 ORG 的相关名。

```
SELECT NAME, DEPTNAME
FROM STAFF S, ORG O
WHERE O.MANAGER = S.ID
```

一旦定义了相关名，则只能使用相关名来限定对象。例如，在上例中，如果写成 `ORG.MANAGER=STAFF.ID` 的话，则该语句就会失效。

也可以使用相关名作为表示数据库对象的简称。只输入 `S` 比输入 `STAFF` 更容易。

通过使用相关名，可复制对象。这在需要将表中各项与自己本身作比较时很有用。在下列示例中，`EMPLOYEE` 表与它自己的另一个实例比较以查找所有雇员的经理。该示例显示并非设计人员的雇员的姓名、这些雇员的经理的姓名以及部门号。

```
SELECT E2.FIRSTNME, E2.LASTNAME, E2.JOB, E1.FIRSTNME AS MGR_FIRSTNAME,
       E1.LASTNAME AS MGR_LASTNAME, E1.WORKDEPT
FROM EMPLOYEE E1, EMPLOYEE E2
WHERE E1.WORKDEPT = E2.WORKDEPT
      AND E1.JOB = 'MANAGER'
      AND E2.JOB <> 'MANAGER'
      AND E2.JOB <> 'DESIGNER'
```

此语句产生下列结果:

FIRSTNME	LASTNAME	JOB	MGR_FIRSTNAME	MGR_LASTNAME	WORKDEPT
DOLORES	QUINTANA	ANALYST	SALLY	KWAN	C01
HEATHER	NICHOLLS	ANALYST	SALLY	KWAN	C01
JAMES	JEFFERSON	CLERK	EVA	PULASKI	D21
SALVATORE	MARINO	CLERK	EVA	PULASKI	D21
DANIEL	SMITH	CLERK	EVA	PULASKI	D21
SYBIL	JOHNSON	CLERK	EVA	PULASKI	D21
MARIA	PEREZ	CLERK	EVA	PULASKI	D21
ETHEL	SCHNEIDER	OPERATOR	EILEEN	HENDERSON	E11
JOHN	PARKER	OPERATOR	EILEEN	HENDERSON	E11
PHILIP	SMITH	OPERATOR	EILEEN	HENDERSON	E11
MAUDE	SETRIGHT	OPERATOR	EILEEN	HENDERSON	E11
RAMLAL	MEHTA	FIELDREP	THEODORE	SPENSER	E21
WING	LEE	FIELDREP	THEODORE	SPENSER	E21
JASON	GOUNOT	FIELDREP	THEODORE	SPENSER	E21

相关子查询

允许引用先前提到的任何表的子查询称为*相关子查询*。我们也说该子查询具有对主查询中表的*相关引用*。

以下示例使用一个*不相关子查询*来列出部门 'A00' 中工资超过该部门平均工资的雇员的雇员号和姓名:

```
SELECT EMPNO, LASTNAME
FROM EMPLOYEE
WHERE WORKDEPT = 'A00'
```

```

AND SALARY > (SELECT AVG(SALARY)
              FROM EMPLOYEE
              WHERE WORKDEPT = 'A00')

```

此语句产生下列结果:

```

EMPNO  LASTNAME
-----
000010 HAAS
000110 LUCCHESSI

```

如果想要知道每个部门的平均工资，则需要对每个部门计算一次子查询。通过对在外层查询中标识的表的每一行执行一次 SQL 的相关功能（该功能允许您编写重复执行的子查询），就可做到这一点。

以下示例使用相关子查询来列出其工资高于部门平均工资的所有雇员:

```

SELECT E1.EMPNO, E1.LASTNAME, E1.WORKDEPT
FROM EMPLOYEE E1
WHERE SALARY > (SELECT AVG(SALARY)
               FROM EMPLOYEE E2
               WHERE E2.WORKDEPT = E1.WORKDEPT)
ORDER BY E1.WORKDEPT

```

在此查询中，对每个部门计算一次子查询。结果为:

EMPNO	LASTNAME	WORKDEPT
000010	HAAS	A00
000110	LUCCHESSI	A00
000030	KWAN	C01
000060	STERN	D11
000150	ADAMSON	D11
000170	YOSHIMURA	D11
000200	BROWN	D11
000220	LUTZ	D11
000070	PULASKI	D21
000240	MARINO	D21
000270	PEREZ	D21
000090	HENDERSON	E11
000280	SCHNEIDER	E11
000100	SPENSER	E21
000330	LEE	E21
000340	GOUNOT	E21

要编写带有相关子查询的查询，使用与带有子查询的普通外部查询相同的基本格式。然而，在外部查询的 FROM 子句中，只是在表名后面放置一个相关名。于是子查询可能包含由该相关名限定的列引用。例如，如果 E1 是相关名，则 E1.WORKDEPT 表示外部查询中表的当前行的工作部门值。在外部查询中对表的每一行（概念上）重新计算子查询。

通过使用相关子查询，可以使系统为您工作并减少需要在应用程序中编写的代码量。

DB2 中允许非限定相关引用。例如，表 EMPLOYEE 有一个命名为 LASTNAME 的列，但表 SALES 有一个命名为 SALES_PERSON 的列，且没有命名为 LASTNAME 的列。

```
SELECT LASTNAME, FIRSTNAME, COMM
FROM EMPLOYEE
WHERE 3 > (SELECT AVG(SALES)
           FROM SALES
           WHERE LASTNAME = SALES_PERSON)
```

在此示例中，系统检查最内层的 FROM 子句，以获取 LASTNAME 列。如果未找到 LASTNAME 列，则系统检查次最内层的 FROM 子句（此情况下为外部 FROM 子句）。虽然不总是必要的，还是建议限定相关引用以改进查询的可读性并确保获取想要的结果。

实现相关子查询

想何时使用相关子查询？列函数的使用有时是一条线索。

假定您想要列出教育级别高于部门平均值的雇员。

首先，您必须确定选择列表项。问题为『List the employees』。这意味着 EMPLOYEE 表中的 LASTNAME 应能足以唯一标识雇员。该问题也将 EDLEVEL 和雇员的部门 WORKDEPT 说明为条件。当问题未明确要求显示列时，在选择列表中包括这些列将会有助于说明解法。现在可构造查询的一部分：

```
SELECT LASTNAME, WORKDEPT, EDLEVEL
FROM EMPLOYEE
```

接着需要搜索条件（WHERE子句）。问题语句为『...whose level of education is higher than the average for that employee's department』。这意味着对于表中每个雇员，必须计算该雇员所在部门的平均教育级别。此语句适合相关子查询的说明。正在对每一行计算一些未知特性（当前雇员所在部门的平均教育级别）。EMPLOYEE 表需要相关名：

```
SELECT LASTNAME, WORKDEPT, EDLEVEL
FROM EMPLOYEE E1
```

需要的子查询较简单。该子查询计算每个部门的平均教育级别。完整的 SQL 语句为：


```

SELECT LASTNAME, WORKDEPT, EDLEVEL
  FROM EMPLOYEE E1
 WHERE EDLEVEL > (SELECT AVG(EDLEVEL)
                  FROM EMPLOYEE E2
                  WHERE E2.WORKDEPT = E1.WORKDEPT)

```

结果为:

LASTNAME	WORKDEPT	EDLEVEL
HAAS	A00	18
KWAN	C01	20
PULASKI	D21	16
HENDERSON	E11	16
LUCCHESI	A00	19
PIANKA	D11	17
SCOUTTEN	D11	17
JONES	D11	17
LUTZ	D11	18
MARINO	D21	17
JOHNSON	D21	16
SCHNEIDER	E11	17
MEHTA	E21	16
GOUNOT	E21	16

假定不列出雇员的部门号，则应列出部门名称。需要的信息 (DEPTNAME) 在独立表 (DEPARTMENT) 中。定义相关变量的外层查询也可以是连接查询（参见第25页的『从多个表中选择数据』以了解详情）。

当在外层查询中使用连接时，列出要在 FROM 子句中连接的表，并将相关名放在适当表名的旁边。

要修改查询以列出部门名称而不是部门号，在选择列表中用 DEPTNAME 替换 WORKDEPT。FROM 子句现在也必须包括 DEPARTMENT 表，并且 WHERE 子句必须表示适当的连接条件。

以下是修改的查询:

```

SELECT LASTNAME, DEPTNAME, EDLEVEL
  FROM EMPLOYEE E1, DEPARTMENT
 WHERE E1.WORKDEPT = DEPARTMENT.DEPTNO
 AND EDLEVEL > (SELECT AVG(EDLEVEL)
                FROM EMPLOYEE E2
                WHERE E2.WORKDEPT = E1.WORKDEPT)

```

此语句产生下列结果:

LASTNAME	DEPTNAME	EDLEVEL
HAAS	SPIFFY COMPUTER SERVICE DIV.	18
LUCCHESI	SPIFFY COMPUTER SERVICE DIV.	19
KWAN	INFORMATION CENTER	20

PIANKA	MANUFACTURING SYSTEMS	17
SCOUTTEN	MANUFACTURING SYSTEMS	17
JONES	MANUFACTURING SYSTEMS	17
LUTZ	MANUFACTURING SYSTEMS	18
PULASKI	ADMINISTRATION SYSTEMS	16
MARINO	ADMINISTRATION SYSTEMS	17
JOHNSON	ADMINISTRATION SYSTEMS	16
HENDERSON	OPERATIONS	16
SCHNEIDER	OPERATIONS	17
MEHTA	SOFTWARE SUPPORT	16
GOUNOT	SOFTWARE SUPPORT	16

上例显示，必须在包含相关子查询的某个查询的 **FROM** 子句中定义用于子查询中的相关名。然而，这种包含可能涉及若干层嵌套。

假定某些部门只有几个雇员，因此这些部门的平均教育级别可能是错误的。为了使平均教育级别在用于比较雇员时是有意义的数字，可以决定一个部门中必须至少有 5 个雇员。因此现在必须列出教育级别高于雇员所在部门平均值的雇员，并只考虑至少有 5 个雇员的部门。

该问题暗含另一个子查询，因为对于外层查询中每个雇员来说，必须计算该雇员所在部门的雇员总数：

```
SELECT COUNT(*)
  FROM EMPLOYEE E3
 WHERE E3.WORKDEPT = E1.WORKDEPT
```

仅当计数大于或等于 5 时才计算平均值：

```
SELECT AVG(EDLEVEL)
  FROM EMPLOYEE E2
 WHERE E2.WORKDEPT = E1.WORKDEPT
 AND 5 <= (SELECT COUNT(*)
           FROM EMPLOYEE E3
           WHERE E3.WORKDEPT = E1.WORKDEPT)
```

最后，只包括其教育级别高于部门平均值的那些雇员：

```
SELECT LASTNAME, DEPTNAME, EDLEVEL
  FROM EMPLOYEE E1, DEPARTMENT
 WHERE E1.WORKDEPT = DEPARTMENT.DEPTNO
 AND EDLEVEL >
 (SELECT AVG(EDLEVEL)
  FROM EMPLOYEE E2
 WHERE E2.WORKDEPT = E1.WORKDEPT
 AND 5 <=
 (SELECT COUNT(*)
  FROM EMPLOYEE E3
 WHERE E3.WORKDEPT = E1.WORKDEPT))
```

此语句产生下列结果：

LASTNAME	DEPTNAME	EDLEVEL
PIANKA	MANUFACTURING SYSTEMS	17
SCOUTTEN	MANUFACTURING SYSTEMS	17
JONES	MANUFACTURING SYSTEMS	17
LUTZ	MANUFACTURING SYSTEMS	18
PULASKI	ADMINISTRATION SYSTEMS	16
MARINO	ADMINISTRATION SYSTEMS	17
JOHNSON	ADMINISTRATION SYSTEMS	16
HENDERSON	OPERATIONS	16
SCHNEIDER	OPERATIONS	17

第6章 在查询中使用运算符和谓词

在 DB2 通用数据库中，可用不同的集合运算符组合查询，并可用定量谓词构造复杂的条件语句。

本章说明如何：

- 用 UNION、EXCEPT 以及 INTERSECT 集合运算符组合不同的表
- 用定量谓词构造用于查询的复杂条件。基本谓词已在第20页的『选择行』中简要讨论过。

用集合运算符组合查询

UNION、EXCEPT 以及 INTERSECT 集合运算符使您能够将两个或更多外层查询组合成单个查询。执行由这些集合运算符连接的每个查询，并将各个结果结合起来。每个运算符产生不同的结果。

UNION 运算符

UNION 运算符通过组合其他两个结果表（例如 TABLE1 和 TABLE2）并消去表中任何重复行而派生出一个结果表。当 ALL 随 UNION 一起使用时（即 UNION ALL），不消除重复行。两种情况下，派生表的每一行不是来自 TABLE1 就是来自 TABLE2。

在下列 UNION 运算符的示例中，查询返回工资高于 \$21,000、有管理责任且工龄少于 8 年的人员的姓名：

1

```
SELECT ID, NAME FROM STAFF WHERE SALARY > 21000  
UNION
```

2

```
SELECT ID, NAME FROM STAFF WHERE JOB='Mgr' AND YEARS < 8  
ORDER BY ID
```

各个查询的结果如下：

1

ID	NAME
140	Fraye
160	Molinare
260	Jones

2

ID	NAME
10	Sanders
30	Marenghi
100	Plotz
140	Fraye
160	Molinare
240	Daniels

数据库管理程序组合这两个查询的结果，消除重复行，并按升序返回最终结果。

ID	NAME
10	Sanders
30	Marenghi
100	Plotz
140	Fraye
160	Molinare
240	Daniels
260	Jones

如果在带有任何集合运算符的查询中使用 **ORDER BY** 子句，则必须在最后一个查询之后写该子句。系统对组合的回答集进行排序。

如果两个表中的列名不同，则组合的结果表没有相应列的名称。替代地，将这些列按其出现的顺序编号。因此，如果想要对结果表进行排序，则必须在 **ORDER BY** 子句中指定列号。

EXCEPT 运算符

EXCEPT 运算符通过包括所有在 **TABLE1** 中但不在 **TABLE2** 中的行并消除所有重复行而派生出一个结果表。当 **ALL** 随 **EXCEPT** 一起使用时 (**EXCEPT ALL**)，不消除重复行。

在以下 **EXCEPT** 运算符的示例中，查询返回收入超过 \$21,000，但没有经理职位而工龄为 8 年或更长的所有人员的姓名。

```

SELECT ID, NAME FROM STAFF WHERE SALARY > 21000
EXCEPT
SELECT ID, NAME FROM STAFF WHERE JOB='Mgr' AND YEARS < 8

```

各个查询的结果列出在关于 UNION 的一节中。上面的语句产生下列结果:

```

ID      NAME
-----
260 Jones

```

INTERSECT 运算符

INTERSECT 运算符通过只包括 TABLE1 和 TABLE2 中都有的行并消除所有重复行而派生出一个结果表。当 ALL 随 INTERSECT 一起使用时 (INTERSECT ALL), 不消除重复行。

在以下 INTERSECT 运算符的示例中, 查询返回收入超过 \$21,000, 有管理职位且工龄少于 8 年的雇员的姓名和 ID。

```

SELECT ID, NAME FROM STAFF WHERE SALARY > 21000
INTERSECT
SELECT ID, NAME FROM STAFF WHERE JOB='Mgr' AND YEARS < 8

```

各个查询的结果在关于 UNION 的一节中列出。这两个使用 INTERSECT 的查询的结果为:

```

ID      NAME
-----
140 Fraye
160 Molinare

```

当使用 UNION、EXCEPT 以及 INTERSECT 运算符时, 记住下列事项:

- 运算符的查询选择列表中的所有对应项必须是相容的。有关详情, 参见 *SQL Reference* 中的数据类型相容性表。
- ORDER BY 子句 (如果使用该子句的话) 必须放在最后一个带有集合运算符的查询后面。对于每个运算符来说, 仅当列名与查询的选择列表中的对应项完全相同时, 该列名才能在 ORDER BY 子句中使用。
- 在具有相同数据类型和相同长度的列之间进行的运算会产生一个具有该类型和长度的列。针对 UNION、EXCEPT 以及 INTERSECT 集合运算符的结果, 参见 *SQL Reference* 中结果数据类型的规则。

谓词

谓词允许您构造条件, 以便只处理满足这些条件的那些行。第20页的『选择行』中讨论了基本谓词。本节讨论 IN、BETWEEN、LIKE、EXISTS 以及定量谓词。

使用 IN 谓词

使用 IN 谓词将一个值与其他几个值进行比较。例如:

```
SELECT NAME
FROM STAFF
WHERE DEPT IN (20, 15)
```

此示例相当于:

```
SELECT NAME
FROM STAFF
WHERE DEPT = 20 OR DEPT = 15
```

当子查询返回一组值时, 可使用 IN 和 NOT IN 运算符。例如, 以下查询列出了负责项目 MA2100 和 OP2012 的雇员的姓:

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE EMPNO IN
  (SELECT RESPEMP
   FROM PROJECT
   WHERE PROJNO = 'MA2100'
   OR PROJNO = 'OP2012')
```

计算一次子查询, 并将结果列表直接代入外层查询。例如, 如果上面的子查询选择雇员号 10 和 330, 则对外层查询进行计算, 就好象 WHERE 子句为:

```
WHERE EMPNO IN (10, 330)
```

子查询返回的值列表可包含零个、一个或多个值。

使用 BETWEEN 谓词

BETWEEN 谓词将单一值与一个范围内的值 (在 BETWEEN 谓词中命名) 作比较。

以下示例查找收入在 \$10,000 和 \$20,000 之间的雇员的姓名:

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE SALARY BETWEEN 10000 AND 20000
```

这相当于:

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE SALARY >= 10000 AND SALARY <= 20000
```

下一个示例查找收入少于 \$10,000 或超过 \$20,000 的雇员的姓名:

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE SALARY NOT BETWEEN 10000 AND 20000
```


使用 LIKE 谓词

使用 LIKE 谓词搜索具有某些模式的字符串。通过百分号和下划线指定模式。

- 下划线字符(_)表示任何单个字符。
- 百分号(%)表示零或多个字符的字符串。
- 任何其他表示本身的字符。

以下示例选择长为 7 个字母且以字母 'S' 开头的雇员姓名:

```
SELECT NAME
FROM STAFF
WHERE NAME LIKE 'S _ _ _ _ _ _ _'
```

下一个示例选择不以字母'S'开头的雇员名:

```
SELECT NAME
FROM STAFF
WHERE NAME NOT LIKE 'S%'
```

使用 EXISTS 谓词

可使用子查询来测试满足某个条件的行的存在性。在此情况下,谓词 EXISTS 或 NOT EXISTS 将子查询链接到外层查询。

当用 EXISTS 谓词将子查询链接到外层查询时,该子查询不返回值。相反,如果子查询的回答集包含一个或更多个行,则 EXISTS 谓词为真;如果回答集不包含任何行,则 EXISTS 谓词为假。

通常将 EXISTS 谓词与相关子查询一起使用。下面的示例列出了当前在项目 (PROJECT) 表中没有项的部门:

```
SELECT DEPTNO, DEPTNAME
FROM DEPARTMENT X
WHERE NOT EXISTS
      (SELECT *
       FROM PROJECT
        WHERE DEPTNO = X.DEPTNO)
ORDER BY DEPTNO
```

可通过在外层查询的 WHERE 子句中使用 AND 和 OR 将 EXISTS 和 NOT EXISTS 谓词与其他谓词连接起来。

定量谓词

定量谓词将一个值和值的集合进行比较。如果全查询返回多个值，则必须通过附加后缀 **ALL**、**ANY** 或 **SOME** 来修改谓词中的比较运算符。这些后缀确定如何在外层谓词中处理返回的这组值。使用 **>** 比较运算符作为示例（下面的注释也适用于其他运算符）：

表达式 **> ALL**（全查询）

如果该表达式大于由全查询返回的每个单值，则该谓词为真。如果全查询未返回值，则该谓词为真。如果指定的关系至少对一个值为假，则结果为假。注意，**<>ALL** 定量谓词相当于 **NOT IN** 谓词。

以下示例使用子查询和 **> ALL** 比较来查找收入超过所有经理的所有雇员的姓名和职位：

```
SELECT LASTNAME, JOB
FROM EMPLOYEE
WHERE SALARY > ALL
(SELECT SALARY
 FROM EMPLOYEE
 WHERE JOB='MANAGER')
```

表达式 **> ANY**（全查询）

如果表达式至少大于由全查询返回的值之一，则该谓词为真。如果全查询未返回值，则该谓词为假。注意，**=ANY** 定量运算符相当于 **IN** 谓词。

表达式 **> SOME**（全查询）

SOME 与 **ANY** 同义。

有关谓词和运算符的详情，参考 *SQL Reference*。

第7章 高级 SQL

本章包括 DB2 通用数据库的几个功能，这些功能允许您在定制查询以满足您的要求的同时更有效地设计查询。本章中的主题假设您对这之前的材料已理解透彻。

本章包括：

- 用约束和触发器实施商业规则
- 连接
- ROLLUP 和 CUBE 查询和递归查询
- OLAP 函数

用约束和触发器实施商业规则

在商界，需要确保始终实施某些规则。例如，参与某个项目的雇员必须在工资单上。或者，可能我们想要某些事件自动地发生。例如，如果销售人员售出一些商品，则应增加其佣金。

为此，DB2 通用数据库提供了一套有用的方法：

- 唯一约束禁止在表的一列或多列中出现重复值。
- 参考完整性约束确保在各个指定的表中的数据的一致性。
- 表检查约束是限制列所允许的值的规则。如果正在对列指定的值不满足该列的检查约束，则插入和更新失败。
- 触发器定义一组操作，这组操作是由对指定的表进行删除、插入或更新操作来执行或触发的。触发器可用于写入其他表、修改输入值以及发布警报信息。

第一节提供关键字的概念性概述。稍后，通过示例和图解来研究参考完整性、约束和触发器。

关键字

关键字是可用来标识或存取特定行的一组列。

由不止一列组成的关键字称为组合关键字。在具有组合关键字的表中，组合关键字中各列的排序不一定与这些列在表中的排序相对应。

唯一关键字

唯一关键字定义为其中没有相同的值的列（或一组列）。唯一关键字的列不能包含空值。在执行 `INSERT` 和 `UPDATE` 语句期间，数据库管理程序强制执行该约束。一个表可多个唯一关键字。唯一关键字是可选的，并且可在 `CREATE TABLE` 或 `ALTER TABLE` 语句中定义。

主关键字

主关键字是一种唯一关键字，表定义的一部分。一个表不能多个主关键字，并且主关键字的列不能包含空值。主关键字是可选的，并且可在 `CREATE TABLE` 或 `ALTER TABLE` 语句中定义。

外部关键字

外部关键字在参考约束的定义中指定。一个表可以有零个或多个外部关键字。如果组合外部关键字的值的任何部分为空，则该值为空。外部关键字是可选的，并且可在 `CREATE TABLE` 语句或 `ALTER TABLE` 语句中定义。

唯一约束

唯一约束确保关键字的值在表中是唯一的。唯一约束是可选的，并且可以通过使用指定 `PRIMARY KEY` 或 `UNIQUE` 子句的 `CREATE TABLE` 或 `ALTER TABLE` 语句来定义唯一约束。例如，可在一个表的雇员号列上定义一个唯一约束，以确保每个雇员有唯一的号码。

参考完整性约束

通过定义唯一约束和外部关键字，可以定义表与表之间的关系，从而实施某些商业规则。唯一关键字和外部关键字约束的组合通常称为参考完整性约束。外部关键字所引用的唯一约束称为父关键字。外部关键字表示特定的父关键字，或与特定的父关键字相关。例如，某规则可能规定每个雇员（`EMPLOYEE` 表）必须属于某现存的部门（`DEPARTMENT` 表）。因此，将 `EMPLOYEE` 表中的“部门号”定义为外部关键字，而将 `DEPARTMENT` 表中的“部门号”定义为主关键字。下列图表提供参考完整性约束的直观说明。

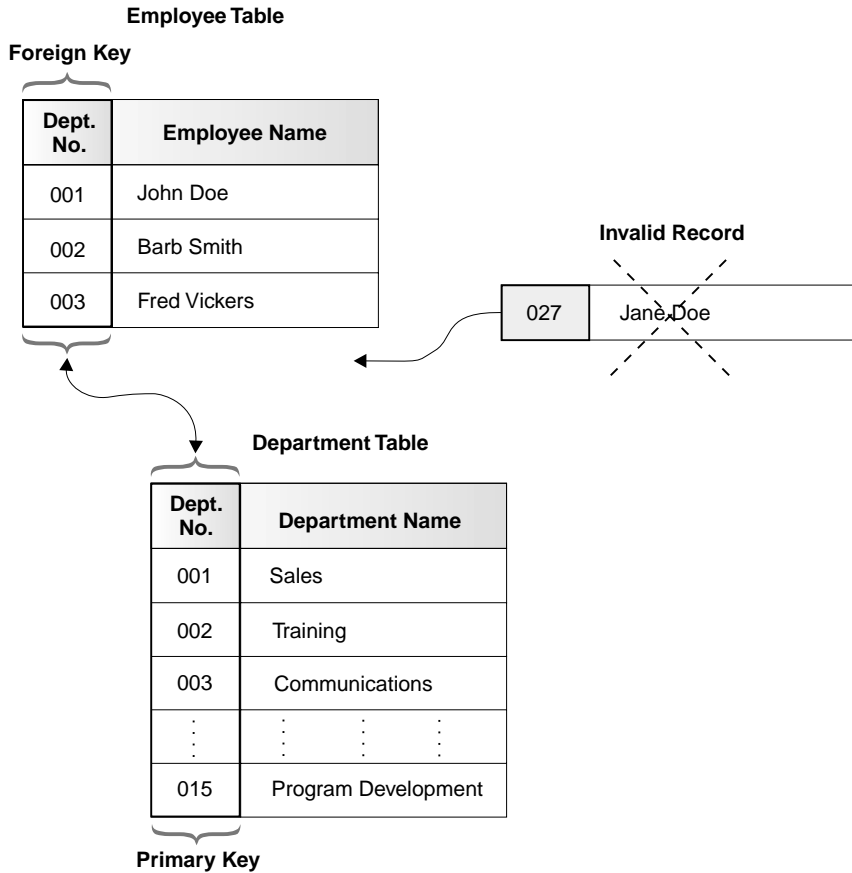


图 4. 外部约束和主约束定义关系并保护数据

表检查约束

表检查约束指定对于表的每行都要进行判定的条件。可对个别列指定检查约束。可使用 `CREATE` 或 `ALTER TABLE` 语句添加检查约束。

下列语句创建具有下列约束的表:

- 部门号的值必须在范围 10 至 100 内
- 雇员的职务只能为下列之一: “Sales”、“Mgr” 或 “Clerk”
- 1986 年之前雇用的每个雇员的工资必须超过 \$40,500。

```
CREATE TABLE EMP
  (ID          SMALLINT NOT NULL,
   NAME       VARCHAR(9),
   DEPT       SMALLINT CHECK (DEPT BETWEEN 10 AND 100),
   JOB        CHAR(5) CHECK (JOB IN ('Sales', 'Mgr', 'Clerk')),
```

```

HIREDATE    DATE,
SALARY      DECIMAL(7,2),
COMM        DECIMAL(7,2),
PRIMARY KEY (ID),
CONSTRAINT YEARSAL CHECK
            (YEAR(HIREDATE) >= 1986 OR SALARY > 40500) )

```

仅当条件判定为假时才会违反约束。例如，如果插入行的 DEPT 为空值，则插入继续进行而不出错，尽管 DEPT 的值应该象约束中定义的那样在 10 和 100 之间。

下列语句将一个约束添加至名为 COMP 的 EMPLOYEE 表中，该约束为雇员的总报酬必须超过 \$15,000:

```

ALTER TABLE EMP
ADD CONSTRAINT COMP CHECK (SALARY + COMM > 15000)

```

将检查表中现存的行以确保这些行不违反新约束。可通过使用如下的 SET CONSTRAINTS 语句将此检查延期:

```

SET CONSTRAINTS FOR EMP OFF
ALTER TABLE EMP ADD CONSTRAINT COMP CHECK (SALARY + COMM > 15000)
SET CONSTRAINTS FOR EMP IMMEDIATE CHECKED

```

首先使用 SET CONSTRAINTS 语句以延期对表的约束检查。然后可将一个或多个约束添加至表而不检查这些约束。接着再次发出 SET CONSTRAINTS 语句，反过来将约束检查打开并执行任何延期的约束检查。

触发器

触发器定义一组操作。可由修改指定基表中的数据的操作来激活触发器。

触发器的一些用途:

- 执行输入数据的验证
- 自动生成新插入的行的值
- 为交叉引用而从其它表读取
- 为审查记录而写入其它表
- 通过电子邮件信息支持警报

使用触发器将导致应用程序开发和商业规则的全面实施更快速，以及使得应用程序和数据的维护更容易。

DB2 通用数据库支持几种类型的触发器。可定义触发器在 DELETE、INSERT 或 UPDATE 操作之前或之后激活。每个触发器包括一组称为触发操作的 SQL 语句，这组语句可包括一个可选的搜索条件。

可进一步定义后触发器以对每一行都执行触发操作，或对语句执行一次触发操作。前触发器总是对每一行都执行触发操作。

在 INSERT、UPDATE 或 DELETE 语句之前使用触发器，以便在执行触发操作之前检查某些条件，或在将输入值存储在表中之前更改输入值。

使用后触发器，以便在必要时传播值或执行其他任务，如发送信息等，这些任务可能是触发器操作所要求的。

以下示例说明了前触发器和后触发器的一个用途。考虑一个记录并跟踪股票价格波动的应用程序。该数据库包含两个表，CURRENTQUOTE 和 QUOTEHISTORY，定义如下：

```
CREATE TABLE CURRENTQUOTE
(SYMBOL VARCHAR(10),
QUOTE DECIMAL(5,2),
STATUS VARCHAR(9))

CREATE TABLE QUOTEHISTORY
(SYMBOL VARCHAR(10),
QUOTE DECIMAL(5,2),
TIMESTAMP TIMESTAMP)
```

当使用如下语句更新 CURRENTQUOTE 的 QUOTE 列时：

```
UPDATE CURRENTQUOTE
SET QUOTE = 68.5
WHERE SYMBOL = 'IBM'
```

应更新 CURRENTQUOTE 的 STATUS 列以反映股票是否：

- 在升值
- 处于本年度的新高
- 在下跌
- 处于本年度的新低
- 价位稳定

这通过使用下列前触发器来实现：

1

```
CREATE TRIGGER STOCK_STATUS
NO CASCADE BEFORE UPDATE OF QUOTE ON CURRENTQUOTE
REFERENCING NEW AS NEWQUOTE OLD AS OLDQUOTE
FOR EACH ROW MODE DB2SQL
```

2

```
SET NEWQUOTE.STATUS =
```

3

```
CASE
```

4

```
WHEN NEWQUOTE.QUOTE >=
    (SELECT MAX(QUOTE)
     FROM QUOTEHISTORY
     WHERE SYMBOL = NEWQUOTE.SYMBOL
     AND YEAR(TIMESTAMP) = YEAR(CURRENT DATE) )
THEN 'High'
```

5

```
WHEN NEWQUOTE.QUOTE <=
    (SELECT MIN(QUOTE)
     FROM QUOTEHISTORY
     WHERE SYMBOL = NEWQUOTE.SYMBOL
     AND YEAR(TIMESTAMP) = YEAR(CURRENT DATE) )
THEN 'Low'
```

6

```
WHEN NEWQUOTE.QUOTE > OLDQUOTE.QUOTE
    THEN 'Rising'
WHEN NEWQUOTE.QUOTE < OLDQUOTE.QUOTE
    THEN 'Dropping'
WHEN NEWQUOTE.QUOTE = OLDQUOTE.QUOTE
    THEN 'Steady'
```

```
END
```

1

此代码块将 `STOCK_STATUS` 定义为一个应该在更新 `CURRENTQUOTE` 表的 `QUOTE` 列之前激活的触发器。第二行指定，在将 `CURRENTQUOTE` 表的实际更新所引起的任何更改应用于数据库之前，要应用触发操作。`NO CASCADE` 子句意味着触发操作将不会导致激活任何其他触发器。第三行指定一些名称，必须将这些名称作为列名的限定符用于新值 (`NEWQUOTE`) 和旧值 (`OLDQUOTE`)。用这些相关名 (`NEWQUOTE` 和 `OLDQUOTE`) 限定的列名称为转换变量。第四行表示应对每一行都执行触发操作。

2

这标记此触发器的触发操作中第一个也是唯一的一个 `SQL` 语句的开始。`SET` 转换变量语句在一个触发器中用来将值赋给表的行中的列，该表正在由激活该触发器的语句进行更新。此语句正在给 `CURRENTQUOTE` 表的 `STATUS` 列赋值。

3

该赋值语句右边使用的表达式为 `CASE` 表达式。`CASE` 表达式扩充为 `END` 关键字。

- 4 第一种情况检查新报价 (NEWQUOTE.QUOTE) 是否超过当前日历年度中股票符号的最高价。子查询正在使用由跟在后面的后触发器更新的 QUOTEHISTORY 表。
- 5 第二种情况检查新报价 (NEWQUOTE.QUOTE) 是否小于当前日历年度中股票符号的最低价。子查询正在使用由跟在后面的后触发器更新的 QUOTEHISTORY 表。
- 6 最后三种情况将新报价 (NEWQUOTE.QUOTE) 与表 (OLDQUOTE.QUOTE) 中的报价比较, 以确定新报价是大于、小于还是等于旧报价。SET 转换变量语句在此处结束。

除了更新 CURRENTQUOTE 表中的项之外, 还需要通过将新报价连同时间戳记一起复制到 QUOTEHISTORY 表中来创建一个审查记录。这通过使用下列后触发器来实现:

1

```
CREATE TRIGGER RECORD_HISTORY  
AFTER UPDATE OF QUOTE ON CURRENTQUOTE  
REFERENCING NEW AS NEWQUOTE  
FOR EACH ROW MODE DB2SQL  
BEGIN ATOMIC
```

2

```
INSERT INTO QUOTEHISTORY  
VALUES (NEWQUOTE.SYMBOL, NEWQUOTE.QUOTE, CURRENT TIMESTAMP);  
END
```

1

此代码块将命名为 RECORD_HISTORY 的触发器定义为应该在更新 CURRENTQUOTE 表的 QUOTE 列之后激活的触发器。第三行指定应作为列名的限定符用于新值 (NEWQUOTE) 的名称。第四行表示应对每一行都执行触发操作。

2

此触发器的触发操作包括单个 SQL 语句, 该语句使用已更新的行中的数据 (NEWQUOTE.SYMBOL 和 NEWQUOTE.QUOTE) 和当前的时间戳记将该行插入 QUOTEHISTORY 表。

CURRENT TIMESTAMP 是包含时间戳记的专用寄存器。第68页的『专用寄存器』中提供了列表和解释。

连接

从两个或更多个表中组合数据的过程称为连接表。数据库管理程序从指定的表中形成行的所有组合。对于每个组合, 它都测试连接条件。连接条件是带有一些约束的搜索条件。有关约束的列表, 参考 *SQL Reference*。

注意，连接条件涉及的列的数据类型不必相同；然而，这些数据类型必须相容。计算连接条件的方式与计算其他搜索条件的方式相同，并且使用相同的比较规则。

如果未指定连接条件，则返回在 **FROM** 子句中列出的表中行的所有组合，即使这些行可能完全不相关。该结果称为这两个表的交叉积。

本节中的示例基于下面两个表。这两个表只是样本数据库中表的简化形式，在样本数据库中并不存在。这两个表一般用来概述关于连接的重点。 **SAMP_STAFF** 列出未作为合同工雇用的雇员的姓名以及这些雇员的职务说明，而 **SAMP_PROJECT** 则列出雇员（合同工和全职人员）的姓名以及这些雇员所参与的项目。

这些表如下：

NAME	PROJ
Haas	AD3100
Thompson	PL2100
Walker	MA2112
Lutz	MA2111

图 5. **SAMP_PROJECT** 表

NAME	JOB
Haas	PRES
Thompson	MANAGER
Lucchessi	SALESREP
Nicholls	ANALYST

图 6. **SAMP_STAFF** 表

以下示例产生两个表的交叉积。未指定连接条件，因而存在每一个行组合：

```
SELECT SAMP_PROJECT.NAME,  
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB  
FROM SAMP_PROJECT, SAMP_STAFF
```

此语句产生下列结果：

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
Thompson	PL2100	Haas	PRES
Walker	MA2112	Haas	PRES
Lutz	MA2111	Haas	PRES
Haas	AD3100	Thompson	MANAGER
Thompson	PL2100	Thompson	MANAGER
Walker	MA2112	Thompson	MANAGER
Lutz	MA2111	Thompson	MANAGER
Haas	AD3100	Lucchessi	SALESREP
Thompson	PL2100	Lucchessi	SALESREP
Walker	MA2112	Lucchessi	SALESREP
Lutz	MA2111	Lucchessi	SALESREP
Haas	AD3100	Nicholls	ANALYST
Thompson	PL2100	Nicholls	ANALYST
Walker	MA2112	Nicholls	ANALYST
Lutz	MA2111	Nicholls	ANALYST

两个主要的连接类型是内连接和外连接。到目前为止，所有示例中使用的都是内连接。内连接只保留交叉积中满足连接条件的那些行。如果某行在一个表中存在，但在另一个表中不存在，则结果表中不包括该信息。

下列示例产生两个表的内连接。该内连接列出分配给某个项目的全职雇员：

```

SELECT SAMP_PROJECT.NAME,
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT, SAMP_STAFF
WHERE SAMP_STAFF.NAME = SAMP_PROJECT.NAME

```

或者，也可以指定如下内连接：

```

SELECT SAMP_PROJECT.NAME,
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT INNER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME

```

结果是：

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
Thompson	PL2100	Thompson	MANAGER

注意，该内连接的结果由与右表和左表中 NAME 列的值匹配的行组成 - 'Haas' 和 'Thompson' 都包括在 SAMP_STAFF 表（该表列出所有全职雇员）中和 SAMP_PROJECT 表（该表列出分配给某个项目的全职雇员和合同雇员）中。

外连接是内连接和左表和 / 或右表中不在内连接中的那些行的并置。当对两个表执行外连接时，可任意将一个表指定为左表而将另一个表指定为右表。外连接有三种类型：

1. **左外连接**包括内连接和左表中未包括在内连接中的那些行。
2. **右外连接**包括内连接和右表中未包括在内连接中的那些行。
3. **全外连接**包括内连接以及左表和右表中未包括在内连接中的行。

使用 `SELECT` 语句来指定要显示的列。在 `FROM` 子句中，列出后跟关键字 `LEFT OUTER JOIN`、`RIGHT OUTER JOIN` 或 `FULL OUTER JOIN` 的第一个表的名称。接着需要指定后跟 `ON` 关键字的第二个表。在 `ON` 关键字后面，指定表示要连接的表之间关系的连接条件。

在下列示例中，将 `SAMP_STAFF` 指定为右表，而 `SAMP_PROJECT` 则被指定为左表。通过使用 `LEFT OUTER JOIN`，列出所有全职雇员和合同雇员（在 `SAMP_PROJECT` 中列出）的姓名和项目号，如果是全职雇员（在 `SAMP_STAFF` 中列出），还列出这些雇员的职位：

```
SELECT SAMP_PROJECT.NAME, SAMP_PROJECT.PROJ,
       SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT LEFT OUTER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

此语句产生下列结果：

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
Lutz	MA2111	-	-
Thompson	PL2100	Thompson	MANAGER
Walker	MA2112	-	-

所有列中都具有值的那些行是该内连接的结果。这些都是满足连接条件的行：'Haas'和'Thompson'既在 `SAMP_PROJECT`（左表）中列出又在 `SAMP_STAFF`（右表）中列出。对于不满足连接条件的行，右表的列上出现空值：'Lutz'和'Walker'都是在 `SAMP_PROJECT` 表中列出的合同雇员，因而未在 `SAMP_STAFF` 表中列出。注意，左表中的所有行都包括在结果集中。

在下一个示例中，将 `SAMP_STAFF` 指定为右表而 `SAMP_PROJECT` 则被指定为左表。通过使用 `RIGHT OUTER JOIN` 列出所有全职雇员（在 `SAMP_STAFF` 中列出）的姓名和工作职位，如果将这些雇员分配给了某个项目（在 `SAMP_PROJECT` 中列出），还列出他们的项目编号：

```
SELECT SAMP_PROJECT.NAME,
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT RIGHT OUTER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

结果为：

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
-	-	Lucchessi	SALESREP
-	-	Nicholls	ANALYST
Thompson	PL2100	Thompson	MANAGER

象在左外连接中一样，所有列中都具有值的那些行是内连接的结果。这些都是满足连接条件的行：'Haas'和'Thompson'既在 SAMP_PROJECT（左表）中列出又在 SAMP_STAFF（右表）中列出。对于不满足连接条件的行，右表的列上出现空值：'Lucchessi'和'Nicholls'都是未分配给某项目的全职雇员。虽然他们在 SAMP_STAFF 中列出，但未在 SAMP_PROJECT 中列出。注意，右表中的所有行都包括在结果集中。

下一个示例对 SAMP_PROJECT 表和 SAMP_STAFF 表使用 FULL OUTER JOIN。该示例列出所有全职雇员（包括未分配给某项目的雇员）和合同雇员的姓名：

```
SELECT SAMP_PROJECT.NAME, SAMP_PROJECT.PROJ,
       SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT FULL OUTER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

结果为：

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
-	-	Lucchessi	SALESREP
-	-	Nicholls	ANALYST
Thompson	PL2100	Thompson	MANAGER
Lutz	MA2111	-	-
Walker	MA2112	-	-

此结果包括左外连接、右外连接以及内连接。列出所有全职雇员和合同雇员。正如左外连接和右外连接一样，对于不满足连接条件的值，各个列中会出现空值。SAMP_STAFF 和 SAMP_PROJECT 中的每一行都包括在结果集中。

复杂查询

DB2 通用数据库允许您通过使用 ROLLUP 和 CUBE 分组、合并及查看单个结果集中的多列。这种新型而强大的功能增强并简化了基于数据分析的 SQL。

有很多方法可从数据库中抽取有用信息。可执行递归查询从现存数据集中产生结果表。

ROLLUP 和 CUBE 查询

在查询的 GROUP BY 子句中指定 ROLLUP 和 CUBE 运算。

ROLLUP 分组产生包含常规分组行和小计行的结果集。 CUBE 分组产生包含来自 ROLLUP 和交叉制表行中的行的结果集。

所以对于 ROLLUP，可获取每人每月的销售额及每月的销售总额，以及全年销售总额。对于 CUBE，将包括每人销售总额的附加行。

参见 *SQL Reference* 以了解详情。

递归查询

递归查询是迭代使用结果数据来确定进一步结果的查询。可以把这想象成在一棵树上或一幅图中来回移动。

实际示例包括保留系统、网络规划和调度。

递归查询是使用包括引用自己名称的的公共表表达式来编写的。

参见 *SQL Reference* 以获取递归查询的示例。

OLAP 函数

联机分析处理 (OLAP) 函数对一窗口数据执行列函数操作。此窗口可指定分区内的行分区、行排序或聚合组。聚合组允许用户指定参与计算的行（相对于当前行）。使用这样的窗口允许象计算累计总和及滑动平均值这样的操作。

除允许用户为现存列函数（如 SUM 和 AVG）指定窗口外，OLAP 函数还可执行排序（RANK 和 DENSE_RANK）操作，并提供行号编制（ROW_NUMBER），给出特定的行分区和行排序。

以下示例查询根据工资给出了部门内的雇员级别，并显示部门（对于第 15 部门和第 38 部门）内的工资的累计总和：

```
SELECT NAME, DEPT,
       RANK () OVER (PARTITION BY DEPT ORDER BY SALARY DESC) AS RANK,
       SUM (SALARY) OVER (PARTITION BY DEPT
                          ORDER BY SALARY DESC
                          ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
       AS CUMULATIVE_SUM
FROM STAFF
WHERE DEPT IN (15,38)
ORDER BY DEPT, RANK
```

此语句产生下列结果：

NAME	DEPT	RANK	CUMULATIVE_SUM
Hanes	15	1	20659.80
Rothman	15	2	37162.63
Ngan	15	3	49670.83
Kermisch	15	4	61929.33
O'Brien	38	1	18006.00
Marenghi	38	2	35512.75
Quigley	38	3	52321.05
Naughton	38	4	65275.80
Abrahams	38	5	77285.55

第8章 定制和增强数据操作

本章简要介绍了 DB2 通用数据库中的面向对象的扩充。使用面向对象的扩充有很多优点。用户定义类型 (UDT) 增加了可用于应用程序的数据类型集，而用户定义函数 (UDF) 允许创建应用程序特定的函数。UDF 通过提供类型的一致行为和封装来充当 UDT 的方法。

然后讨论专用寄存器和系统目录。专用寄存器是数据库管理程序定义的存储区；它们用来存储 SQL 语句可引用的信息。专用寄存器是在连接时建立的，且专门用于该应用程序的处理。系统目录包含关于数据库对象的逻辑结构和物理结构的信息。

本章包括：

- 用户定义类型
- 用户定义函数
- 大对象 (LOB)
- 专用寄存器
- 目录视图的介绍

上面主题的详细讨论超出了本书的范围，但在 *SQL Reference* 和管理指南中给出了详细讨论。

用户定义类型

单值类型是与现存类型（它的“源”类型）共享其内部表示的用户定义数据类型，但对于大多数运算来说，认为单值类型是独立和不兼容的。例如，您可能想定义年龄类型、重量类型以及高度类型，所有这些类型都有相当不同的语义，但都使用内部数据类型 INTEGER 作为它们的内部表示。

下列示例说明了命名为 PAY 的单值类型的创建：

```
CREATE DISTINCT TYPE PAY AS DECIMAL(9,2) WITH COMPARISONS
```

虽然 PAY 有与内部数据类型 DECIMAL(9,2) 相同的表示，但还是认为它是与 DECIMAL(9,2) 或任何其他类型不可比的独立类型。它只能与相同的单值类型比较。并且，会影响到按 DECIMAL 使用的运算符和函数将在此不适用。例如，具有 PAY 数据类型的值不能与具有 INTEGER 数据类型的值相乘。因此，您必须编写只应用于 PAY 数据类型的函数。

使用单值数据类型可限制偶然错误。例如，如果将表 EMPLOYEE 的 SALARY 列定义为 PAY 数据类型，则不能将该列添加至 COMM，即使它们的源类型相同。

单值数据类型支持类型转换。源类型可以转换为单值数据类型，单值数据类型也可以转换为源类型。例如，如果将表 EMPLOYEE 的 SALARY 列定义为 PAY 数据类型，则下列示例将不会在比较运算符处出错。

```
SELECT * FROM EMPLOYEE
WHERE DECIMAL(SALARY) = 41250
```

DECIMAL(SALARY) 返回一个十进制数据类型。相反地，数字数据类型可以转换为 PAY 类型。例如，可以使用 PAY(41250) 来转换数字 41250 的类型。

用户定义函数

如第27页的『使用函数』中所述，DB2 通用数据库提供内部函数和用户定义函数 (UDF)。然而，此函数集从不会满足所有需求。您常常需要为特别的任务创建定制函数。用户定义函数允许您创建定制函数。

有四种类型的用户定义函数：有源（或模板）、外部标量，外部表和 OLE DB 外部表。

本节涉及有源类型和外部标量类型。有关外部表类型和 OLE DB 表类型的详情，参见 *SQL Reference*。

源用户定义函数允许用户定义类型有选择地引用另一个已为数据库所知的内部函数或用户定义函数。您可以既使用标量函数又使用列函数。

在下一个示例中，用户定义函数（称为 MAX）是根据内部 MAX 列函数创建的，该内部 MAX 列函数采用 DECIMAL 数据类型作为输入。MAX UDF 采用 PAY 类型作为输入且返回一个 PAY 类型作为输出。

```
CREATE FUNCTION MAX(PAY) RETURNS PAY
SOURCE MAX(DECIMAL)
```

外部用户定义函数由用户用程序设计语言编写。有外部标量函数和外部表函数，在 *SQL Reference* 中讨论了这两个函数。

又例如，假定您已编写了一个计算字符串中字数的函数，则您可以使用 CREATE FUNCTION 语句以名称 WORDCOUNT 向数据库注册该函数。然后就可在 SQL 语句中使用此函数。

以下语句返回雇员号和他们简历的 ASCII 格式的字数。WORDCOUNT 是用户已向数据库注册并且现正在语句中使用的外部标量函数。

```
SELECT EMPNO, WORDCOUNT(RESUME)
FROM EMP_RESUME
WHERE RESUME_FORMAT = 'ascii'
```

有关编写用户定义函数的详情，参考 *Application Development Guide*。

大对象 (LOB)

术语大对象及其缩写词 *LOB* 用于表示三种数据类型：BLOB、CLOB 或 DBCLOB。这些类型可以包含诸如音频、图片以及文档等对象的大量数据。

二进制大对象 (*BLOB*) 是变长字符串，以字节进行量度，最大长度可达 2G 字节。BLOB 主要用来保存非传统数据，如图片、声音以及混合媒体等。

字符大对象 (*CLOB*) 是变长字符串，以字节进行量度，最大长度可达 2G 字节。CLOB 用于存储大的单字节字符集数据，如文档等。CLOB 被认为是字符串。

双字节字符大对象 (*DBCLOB*) 是最大长度可达 2G 字节的双字节字符变长字符串（1 073 741 823 个双字节字符）。DBCLOB 用于存储大的双字节字符集数据，如文档等。DBCLOB 被认为是图形字符串。

操作大对象 (LOB)

由于 LOB 值可以很大，所以将它们从数据库服务器传送至客户机应用程序可能要花费一些时间。然而，一般一次处理 LOB 值的一部分，而不是将它们作为一个整体处理。对于应用程序不需要（或不想要）将整个 LOB 值存储在应用程序内存中的那些情况，应用程序可以通过大对象定位器变量引用此值。

然后后续语句可以使用定位器对数据执行操作，而不必检索整个大对象。定位器变量用来减少应用程序的存储器需求，并通过减少客户机与服务器之间的数据流而改进性能。

另一个机制是文件引用变量。它们用来直接对文件检索大对象或直接从文件来更新表中的大对象。文件引用变量用来减少应用程序的存储器需求，因为这些变量不必存储大对象数据。有关详情，参考 *Application Development Guide* 和 *SQL Reference*。

专用寄存器

专用寄存器是由数据库管理程序为连接定义的存储区，用于存储可以用 SQL 语句引用的信息。下列是几个较常用的专用寄存器的示例。有关所有专用寄存器的列表及详情，参考 *SQL Reference*。

- **CURRENT DATE:** 保存对应于执行 SQL 语句时的日时钟的日期。
- **CURRENT FUNCTION PATH:** 保存一个指定用于分解函数和数据类型引用的函数路径的值。
- **CURRENT SERVER:** 指定当前应用程序服务器。
- **CURRENT TIME:** 保存对应于执行 SQL 语句时的日时钟的时间。
- **CURRENT TIMESTAMP:** 指定对应于执行 SQL 语句时的日时钟的时间戳记。
- **CURRENT TIMEZONE:** 指定世界时与应用程序服务器本地时间之间的差别。
- **USER:** 指定运行权限 ID。

您可以用 VALUES 语句显示专用寄存器的内容。例如：

```
VALUES (CURRENT TIMESTAMP)
```

也可以使用：

```
SELECT CURRENT TIMESTAMP FROM ORG
```

这将为表中的每一行项返回 **TIMESTAMP**。

目录视图的介绍

DB2 为每个数据库创建并维护一个系统目录表扩充集。这些表包含关于数据库对象（如表、视图、程序包、参考完整性关系、函数、单值类型以及触发器）的逻辑结构和物理结构的信息。这些表是在创建数据库时创建的，并在正常操作过程中得到更新。不能显式创建或卸下它们，但可以查询和查看其内容。

有关详情，参考 *SQL Reference*。

从系统目录中选择行

目录视图象任何其他数据库视图一样。可以使用 SQL 语句来查看数据，确切地说，就是使用与查看系统中任何其他视图的相同方式来查看数据。

可以在 SYSCAT.TABLES 目录中找到关于表的非常有用的信息。要查找已创建的现存表的名称，发出一个类似以下语句的语句：

```

SELECT TABNAME, TYPE, CREATE_TIME
FROM SYSCAT.TABLES
WHERE DEFINER = USER

```

此语句产生下列结果:

TABNAME	TYPE	CREATE_TIME
ORG	T	1999-07-21-13.42.55.128005
STAFF	T	1999-07-21-13.42.55.609001
DEPARTMENT	T	1999-07-21-13.42.56.069001
EMPLOYEE	T	1999-07-21-13.42.56.310001
EMP_ACT	T	1999-07-21-13.42.56.710001
PROJECT	T	1999-07-21-13.42.57.051001
EMP_PHOTO	T	1999-07-21-13.42.57.361001
EMP_RESUME	T	1999-07-21-13.42.59.154001
SALES	T	1999-07-21-13.42.59.855001
CL_SCHED	T	1999-07-21-13.43.00.025002
IN_TRAY	T	1999-07-21-13.43.00.055001

下列列表包括与本书讨论的与主题有关的目录视图。还有很多其他目录视图，在 *SQL Reference* 和 *管理指南手册* 中详细列出了它们。

说明	目录视图
检查约束	SYSCAT.CHECKS
列	SYSCAT.COLUMNS
检查约束引用的列	SYSCAT.COLCHECKS
关键字中使用的列	SYSCAT.KEYCOLUSE
数据类型	SYSCAT.DATATYPES
函数参数或函数结果	SYSCAT.FUNCPARMS
参考约束	SYSCAT.REFERENCES
模式	SYSCAT.SCHEMATA
表约束	SYSCAT.TABCONST
表	SYSCAT.TABLES
触发器	SYSCAT.TRIGGERS
用户定义函数	SYSCAT.FUNCTIONS
视图	SYSCAT.VIEWS

附录A. 样本数据库表

本附录显示包含在样本数据库 SAMPLE 的样本表中的信息，以及如何安装和除去样本表。

其他样本数据库与 DB2 通用数据库 一同提供，显示商业智能函数以及在商业智能教程中使用。但是，在本附录中只描述样本数据库 SAMPLE 的内容。参考 *数据仓库中心管理指南* 以得到有关商业智能样本数据库的详尽信息。

样本表在此手册和此资料库中的其他手册中出现的示例中使用。另外，显示样本文件包含的数据类型为 BLOB 和 CLOB 的数据。

下列各节包含在本附录中：

- 第72页的『样本数据库』
- 第72页的『创建样本数据库』
- 第72页的『要擦除样本数据库』
- 第72页的『CL_SCHED 表』
- 第73页的『DEPARTMENT 表』
- 第73页的『EMPLOYEE 表』
- 第76页的『EMP_ACT 表』
- 第78页的『EMP_PHOTO 表』
- 第78页的『EMP_RESUME 表』
- 第79页的『IN_TRAY 表』
- 第79页的『ORG 表』
- 第79页的『PROJECT 表』
- 第81页的『SALES 表』
- 第82页的『STAFF 表』
- 第83页的『STAFFG 表』
- 第84页的『数据类型为 BLOB 和 CLOB 的样本文件』
- 第84页的『Quintana 的相片』
- 第84页的『Quintana 的简历』
- 第85页的『Nicholls 的相片』
- 第86页的『Nicholls 的简历』
- 第87页的『Adamson 的相片』
- 第87页的『Adamson 的简历』
- 第88页的『Walker 的相片』
- 第89页的『Walker 的简历』。

样本数据库表

在样本表中，破折号 (-) 表示空值。

样本数据库

此书中的示例使用样本数据库。要使用这些示例，必须创建 **SAMPLE** 数据库。要使用该数据库，必须安装数据库管理程序。

创建样本数据库

由可执行文件创建样本数据库。² 要创建一个数据库，必须有 **SYSADM** 权限。

- 当使用基于 **UNIX** 的平台时

若正在使用操作系统命令提示符，则输入：

```
sqllib/bin/db2samp1 <path>
```

其中，*path* 是一个可选参数，它指定将用来创建样本数据库的路径。按 **Enter** 键。³ **DB2SAMPL** 的模式是 **CURRENT SCHEMA** 专用寄存器值。

- 当使用 **OS/2** 或 **Windows** 平台时

若正在使用操作系统命令提示符，则输入：

```
db2samp1 e
```

其中，*e* 是一个可选参数，它指定将创建数据库的驱动器。按 **Enter** 键。⁴

若未通过用户简要表管理注册至工作站，则将提示您这样做。

要擦除样本数据库

若您不需要存取样本数据库，您可使用 **DROP DATABASE** 命令来擦除它：

```
db2 drop database sample
```

CL_SCHED 表

Name:	CLASS_CODE	DAY	STARTING	ENDING
Type:	char(7)	smallint	time	time

2. 有关与此命令相关的信息，参见 *Command Reference* 中的 **DB2SAMPL** 命令。

3. 若未指定路径参数，则样本数据库创建在由数据库管理程序配置文件中的 **DFTDBPATH** 参数指定的缺省路径中。

4. 若未指定驱动器参数，则将样本表创建在 **DB2** 所在的驱动器上。

Name:	CLASS_CODE	DAY	STARTING	ENDING
Desc:	Class (room:teacher)	Code Day # of 4 day schedule	Class Start Time	Class End Time

DEPARTMENT 表

Name:	DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
Type:	char(3) not null	varchar(29) not null	char(6)	char(3) not null	char(16)
Desc:	Department number	Name describing general activities of department	Employee number (EMPNO) of department manager	Department (DEPTNO) to which this department reports	Name of the remote location
Values:	A00	SPIFFY COMPUTER SERVICE DIV.	000010	A00	-
	B01	PLANNING	000020	A00	-
	C01	INFORMATION CENTER	000030	A00	-
	D01	DEVELOPMENT CENTER	-	A00	-
	D11	MANUFACTURING SYSTEMS	000060	D01	-
	D21	ADMINISTRATION SYSTEMS	000070	D01	-
	E01	SUPPORT SERVICES	000050	A00	-
	E11	OPERATIONS	000090	E01	-
	E21	SOFTWARE SUPPORT	000100	E01	-

EMPLOYEE 表

Names:	EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE
Type:	char(6) not null	varchar(12) not null	char(1) not null	varchar(15) not null	char(3)	char(4)	date
Desc:	Employee number	First name	Middle initial	Last name	Department (DEPTNO) in which the employee works	Phone number	Date of hire
JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM	
char(8)	smallint not null	char(1)	date	dec(9,2)	dec(9,2)	dec(9,2)	
Job	Number of years of formal education	Sex (M male, F female)	Date of birth	Yearly salary	Yearly bonus	Yearly commission	

在以下各页中查看 EMPLOYEE 表中的值。

EMPNO	FIRSTNAME	MID INIT	LASTNAME	WORK DEPT	PHONE NO	HIREDATE	JOB	ED LEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
char(6) not null	varchar(12) not null	char(1) not null	varchar(15) not null	char(3)	char(4)	date	char(8)	smallint not null	char(1)	date	dec(9,2)	dec(9,2)	dec(9,2)
000010	CHRISTINE	I	HAAS	A00	3978	1965-01-01	PRES	18	F	1933-08-24	52750	1000	4220
000020	MICHAEL	L	THOMPSON	B01	3476	1973-10-10	MANAGER	18	M	1948-02-02	41250	800	3300
000030	SALLY	A	KWAN	C01	4738	1975-04-05	MANAGER	20	F	1941-05-11	38250	800	3060
000050	JOHN	B	GEYER	E01	6789	1949-08-17	MANAGER	16	M	1925-09-15	40175	800	3214
000060	IRVING	F	STERN	D11	6423	1973-09-14	MANAGER	16	M	1945-07-07	32250	500	2580
000070	EVA	D	PULASKI	D21	7831	1980-09-30	MANAGER	16	F	1953-05-26	36170	700	2893
000090	EILEEN	W	HENDERSON	E11	5498	1970-08-15	MANAGER	16	F	1941-05-15	29750	600	2380
000100	THEODORE	Q	SPENSER	E21	0972	1980-06-19	MANAGER	14	M	1956-12-18	26150	500	2092
000110	VINCENZO	G	LUCCHESI	A00	3490	1958-05-16	SALESREP	19	M	1929-11-05	46500	900	3720
000120	SEAN		O'CONNELL	A00	2167	1963-12-05	CLERK	14	M	1942-10-18	29250	600	2340
000130	DOLORES	M	QUINTANA	C01	4578	1971-07-28	ANALYST	16	F	1925-09-15	23800	500	1904
000140	HEATHER	A	NICHOLLS	C01	1793	1976-12-15	ANALYST	18	F	1946-01-19	28420	600	2274
000150	BRUCE		ADAMSON	D11	4510	1972-02-12	DESIGNER	16	M	1947-05-17	25280	500	2022
000160	ELIZABETH	R	PIANKA	D11	3782	1977-10-11	DESIGNER	17	F	1955-04-12	22250	400	1780
000170	MASATOSHI	J	YOSHIMURA	D11	2890	1978-09-15	DESIGNER	16	M	1951-01-05	24680	500	1974
000180	MARILYN	S	SCOUTTEN	D11	1682	1973-07-07	DESIGNER	17	F	1949-02-21	21340	500	1707
000190	JAMES	H	WALKER	D11	2986	1974-07-26	DESIGNER	16	M	1952-06-25	20450	400	1636
000200	DAVID		BROWN	D11	4501	1966-03-03	DESIGNER	16	M	1941-05-29	27740	600	2217
000210	WILLIAM	T	JONES	D11	0942	1979-04-11	DESIGNER	17	M	1953-02-23	18270	400	1462
000220	JENNIFER	K	LUTZ	D11	0672	1968-08-29	DESIGNER	18	F	1948-03-19	29840	600	2387
000230	JAMES	J	JEFFERSON	D21	2094	1966-11-21	CLERK	14	M	1935-05-30	22180	400	1774
000240	SALVATORE	M	MARINO	D21	3780	1979-12-05	CLERK	17	M	1954-03-31	28760	600	2301
000250	DANIEL	S	SMITH	D21	0961	1969-10-30	CLERK	15	M	1939-11-12	19180	400	1534
000260	SYBIL	P	JOHNSON	D21	8953	1975-09-11	CLERK	16	F	1936-10-05	17250	300	1380
000270	MARIA	L	PEREZ	D21	9001	1980-09-30	CLERK	15	F	1953-05-26	27380	500	2190
000280	ETHEL	R	SCHNEIDER	E11	8997	1967-03-24	OPERATOR	17	F	1936-03-28	26250	500	2100
000290	JOHN	R	PARKER	E11	4502	1980-05-30	OPERATOR	12	M	1946-07-09	15340	300	1227
000300	PHILIP	X	SMITH	E11	2095	1972-06-19	OPERATOR	14	M	1936-10-27	17750	400	1420
000310	MAUDE	F	SETRIGHT	E11	3332	1964-09-12	OPERATOR	12	F	1931-04-21	15900	300	1272

EMPNO	FIRSTNME	MID INIT	LASTNAME	WORK DEPT	PHONE NO	HIREDATE	JOB	ED LEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
000320	RAMLAL	V	MEHTA	E21	9990	1965-07-07	FIELDREP	16	M	1932-08-11	19950	400	1596
000330	WING		LEE	E21	2103	1976-02-23	FIELDREP	14	M	1941-07-18	25370	500	2030
000340	JASON	R	GOUNOT	E21	5698	1947-05-05	FIELDREP	16	M	1926-05-17	23840	500	1907

样本数据库表

EMP_ACT 表

Name:	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
Type:	char(6) not null	char(6) not null	smallint not null	dec(5,2)	date	date
Desc:	Employee number	Project number	Activity number	Proportion of employee's time spent on project	Date activity starts	Date activity ends
Values:	000010	AD3100	10	.50	1982-01-01	1982-07-01
	000070	AD3110	10	1.00	1982-01-01	1983-02-01
	000230	AD3111	60	1.00	1982-01-01	1982-03-15
	000230	AD3111	60	.50	1982-03-15	1982-04-15
	000230	AD3111	70	.50	1982-03-15	1982-10-15
	000230	AD3111	80	.50	1982-04-15	1982-10-15
	000230	AD3111	180	1.00	1982-10-15	1983-01-01
	000240	AD3111	70	1.00	1982-02-15	1982-09-15
	000240	AD3111	80	1.00	1982-09-15	1983-01-01
	000250	AD3112	60	1.00	1982-01-01	1982-02-01
	000250	AD3112	60	.50	1982-02-01	1982-03-15
	000250	AD3112	60	.50	1982-12-01	1983-01-01
	000250	AD3112	60	1.00	1983-01-01	1983-02-01
	000250	AD3112	70	.50	1982-02-01	1982-03-15
	000250	AD3112	70	1.00	1982-03-15	1982-08-15
	000250	AD3112	70	.25	1982-08-15	1982-10-15
	000250	AD3112	80	.25	1982-08-15	1982-10-15
	000250	AD3112	80	.50	1982-10-15	1982-12-01
	000250	AD3112	180	.50	1982-08-15	1983-01-01
	000260	AD3113	70	.50	1982-06-15	1982-07-01
	000260	AD3113	70	1.00	1982-07-01	1983-02-01
	000260	AD3113	80	1.00	1982-01-01	1982-03-01
	000260	AD3113	80	.50	1982-03-01	1982-04-15
	000260	AD3113	180	.50	1982-03-01	1982-04-15
	000260	AD3113	180	1.00	1982-04-15	1982-06-01
	000260	AD3113	180	.50	1982-06-01	1982-07-01
	000270	AD3113	60	.50	1982-03-01	1982-04-01
	000270	AD3113	60	1.00	1982-04-01	1982-09-01
	000270	AD3113	60	.25	1982-09-01	1982-10-15
	000270	AD3113	70	.75	1982-09-01	1982-10-15
	000270	AD3113	70	1.00	1982-10-15	1983-02-01

Name:	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
	000270	AD3113	80	1.00	1982-01-01	1982-03-01
	000270	AD3113	80	.50	1982-03-01	1982-04-01
	000030	IF1000	10	.50	1982-06-01	1983-01-01
	000130	IF1000	90	1.00	1982-01-01	1982-10-01
	000130	IF1000	100	.50	1982-10-01	1983-01-01
	000140	IF1000	90	.50	1982-10-01	1983-01-01
	000030	IF2000	10	.50	1982-01-01	1983-01-01
	000140	IF2000	100	1.00	1982-01-01	1982-03-01
	000140	IF2000	100	.50	1982-03-01	1982-07-01
	000140	IF2000	110	.50	1982-03-01	1982-07-01
	000140	IF2000	110	.50	1982-10-01	1983-01-01
	000010	MA2100	10	.50	1982-01-01	1982-11-01
	000110	MA2100	20	1.00	1982-01-01	1982-03-01
	000010	MA2110	10	1.00	1982-01-01	1983-02-01
	000200	MA2111	50	1.00	1982-01-01	1982-06-15
	000200	MA2111	60	1.00	1982-06-15	1983-02-01
	000220	MA2111	40	1.00	1982-01-01	1983-02-01
	000150	MA2112	60	1.00	1982-01-01	1982-07-15
	000150	MA2112	180	1.00	1982-07-15	1983-02-01
	000170	MA2112	60	1.00	1982-01-01	1983-06-01
	000170	MA2112	70	1.00	1982-06-01	1983-02-01
	000190	MA2112	70	1.00	1982-02-01	1982-10-01
	000190	MA2112	80	1.00	1982-10-01	1983-10-01
	000160	MA2113	60	1.00	1982-07-15	1983-02-01
	000170	MA2113	80	1.00	1982-01-01	1983-02-01
	000180	MA2113	70	1.00	1982-04-01	1982-06-15
	000210	MA2113	80	.50	1982-10-01	1983-02-01
	000210	MA2113	180	.50	1982-10-01	1983-02-01
	000050	OP1000	10	.25	1982-01-01	1983-02-01
	000090	OP1010	10	1.00	1982-01-01	1983-02-01
	000280	OP1010	130	1.00	1982-01-01	1983-02-01
	000290	OP1010	130	1.00	1982-01-01	1983-02-01
	000300	OP1010	130	1.00	1982-01-01	1983-02-01
	000310	OP1010	130	1.00	1982-01-01	1983-02-01
	000050	OP2010	10	.75	1982-01-01	1983-02-01
	000100	OP2010	10	1.00	1982-01-01	1983-02-01
	000320	OP2011	140	.75	1982-01-01	1983-02-01

样本数据库表

Name:	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
	000320	OP2011	150	.25	1982-01-01	1983-02-01
	000330	OP2012	140	.25	1982-01-01	1983-02-01
	000330	OP2012	160	.75	1982-01-01	1983-02-01
	000340	OP2013	140	.50	1982-01-01	1983-02-01
	000340	OP2013	170	.50	1982-01-01	1983-02-01
	000020	PL2100	30	1.00	1982-01-01	1982-09-15

EMP_PHOTO 表

Name:	EMPNO	PHOTO_FORMAT	PICTURE
Type:	char(6) not null	varchar(10) not null	blob(100k)
Desc:	Employee number	Photo format	Photo of employee
Values:	000130	bitmap	db200130.bmp
	000130	gif	db200130.gif
	000130	xwd	db200130.xwd
	000140	bitmap	db200140.bmp
	000140	gif	db200140.gif
	000140	xwd	db200140.xwd
	000150	bitmap	db200150.bmp
	000150	gif	db200150.gif
	000150	xwd	db200150.xwd
	000190	bitmap	db200190.bmp
	000190	gif	db200190.gif
	000190	xwd	db200190.xwd

- 第84页的『Quintana 的相片』显示雇员 Delores Quintana 的相片。
- 第85页的『Nicholls 的相片』显示雇员 Heather Nicholls 的相片。
- 第87页的『Adamson 的相片』显示雇员 Bruce Adamson 的相片。
- 第88页的『Walker 的相片』显示雇员 James Walker 的相片。

EMP_RESUME 表

Name:	EMPNO	RESUME_FORMAT	RESUME
Type:	char(6) not null	varchar(10) not null	clob(5k)
Desc:	Employee number	Resume Format	Resume of employee
Values:	000130	ascii	db200130.asc
	000130	script	db200130.scr
	000140	ascii	db200140.asc
	000140	script	db200140.scr

Name:	EMPNO	RESUME_FORMAT	RESUME
	000150	ascii	db200150.asc
	000150	script	db200150.scr
	000190	ascii	db200190.asc
	000190	script	db200190.scr

- 第84页的『Quintana 的简历』显示雇员 Delores Quintana 的简历。
- 第86页的『Nicholls 的简历』显示雇员 Heather Nicholls 的简历。
- 第87页的『Adamson 的简历』显示雇员 Bruce Adamson 的简历。
- 第89页的『Walker 的简历』显示雇员 James Walker 的简历。

IN_TRAY 表

Name:	RECEIVED	SOURCE	SUBJECT	NOTE_TEXT
Type:	timestamp	char(8)	char(64)	varchar(3000)
Desc:	Date and Time received	User id of person	Brief description	The note
		sending note		

ORG 表

Name:	DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
Type:	smallint not null	varchar(14)	smallint	varchar(10)	varchar(13)
Desc:	Department number	Department name	Manager number	Division of Corporation	City
Values:	10	Head Office	160	Corporate	New York
	15	New England	50	Eastern	Boston
	20	Mid Atlantic	10	Eastern	Washington
	38	South Atlantic	30	Eastern	Atlanta
	42	Great Lakes	100	Midwest	Chicago
	51	Plains	140	Midwest	Dallas
	66	Pacific	270	Western	San Francisco
	84	Mountain	290	Western	Denver

PROJECT 表

Name:	PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE	MAJPROJ
Type:	char(6) not null	varchar(24) not null	char(3) not null	char(6) not null	dec(5,2)	date	date	char(6)
Desc:	Project number	Project name	Department responsible	Employee responsible	Estimated mean staffing	Estimated start date	Estimated end date	Major project, for a subproject

样本数据库表

Name:	PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE	MAJPROJ
Values:	AD3100	ADMIN SERVICES	D01	000010	6.5	1982-01-01	1983-02-01	-
	AD3110	GENERAL ADMIN SYSTEMS	D21	000070	6	1982-01-01	1983-02-01	AD3100
	AD3111	PAYROLL PROGRAMMING	D21	000230	2	1982-01-01	1983-02-01	AD3110
	AD3112	PERSONNEL PROGRAMMING	D21	000250	1	1982-01-01	1983-02-01	AD3110
	AD3113	ACCOUNT PROGRAMMING	D21	000270	2	1982-01-01	1983-02-01	AD3110
	IF1000	QUERY SERVICES	C01	000030	2	1982-01-01	1983-02-01	-
	IF2000	USER EDUCATION	C01	000030	1	1982-01-01	1983-02-01	-
	MA2100	WELD LINE AUTOMATION	D01	000010	12	1982-01-01	1983-02-01	-
	MA2110	W L PROGRAMMING	D11	000060	9	1982-01-01	1983-02-01	MA2100
	MA2111	W L PROGRAM DESIGN	D11	000220	2	1982-01-01	1982-12-01	MA2110
	MA2112	W L ROBOT DESIGN	D11	000150	3	1982-01-01	1982-12-01	MA2110
	MA2113	W L PROD CONT PROGS	D11	000160	3	1982-02-15	1982-12-01	MA2110
	OP1000	OPERATION SUPPORT	E01	000050	6	1982-01-01	1983-02-01	-
	OP1010	OPERATION	E11	000090	5	1982-01-01	1983-02-01	OP1000
	OP2000	GEN SYSTEMS SERVICES	E01	000050	5	1982-01-01	1983-02-01	-
	OP2010	SYSTEMS SUPPORT	E21	000100	4	1982-01-01	1983-02-01	OP2000
	OP2011	SCP SYSTEMS SUPPORT	E21	000320	1	1982-01-01	1983-02-01	OP2010
	OP2012	APPLICATIONS SUPPORT	E21	000330	1	1982-01-01	1983-02-01	OP2010
	OP2013	DB/DC SUPPORT	E21	000340	1	1982-01-01	1983-02-01	OP2010
	PL2100	WELD LINE PLANNING	B01	000020	1	1982-01-01	1982-09-15	MA2100

SALES 表

Name:	SALES_DATE	SALES_PERSON	REGION	SALES
Type:	date	varchar(15)	varchar(15)	int
Desc:	Date of sales	Employee's last name	Region of sales	Number of sales
Values:	12/31/1995	LUCCHESSI	Ontario-South	1
	12/31/1995	LEE	Ontario-South	3
	12/31/1995	LEE	Quebec	1
	12/31/1995	LEE	Manitoba	2
	12/31/1995	GOUNOT	Quebec	1
	03/29/1996	LUCCHESSI	Ontario-South	3
	03/29/1996	LUCCHESSI	Quebec	1
	03/29/1996	LEE	Ontario-South	2
	03/29/1996	LEE	Ontario-North	2
	03/29/1996	LEE	Quebec	3
	03/29/1996	LEE	Manitoba	5
	03/29/1996	GOUNOT	Ontario-South	3
	03/29/1996	GOUNOT	Quebec	1
	03/29/1996	GOUNOT	Manitoba	7
	03/30/1996	LUCCHESSI	Ontario-South	1
	03/30/1996	LUCCHESSI	Quebec	2
	03/30/1996	LUCCHESSI	Manitoba	1
	03/30/1996	LEE	Ontario-South	7
	03/30/1996	LEE	Ontario-North	3
	03/30/1996	LEE	Quebec	7
	03/30/1996	LEE	Manitoba	4
	03/30/1996	GOUNOT	Ontario-South	2
	03/30/1996	GOUNOT	Quebec	18
	03/30/1996	GOUNOT	Manitoba	1
	03/31/1996	LUCCHESSI	Manitoba	1
	03/31/1996	LEE	Ontario-South	14
	03/31/1996	LEE	Ontario-North	3
	03/31/1996	LEE	Quebec	7
	03/31/1996	LEE	Manitoba	3
	03/31/1996	GOUNOT	Ontario-South	2
	03/31/1996	GOUNOT	Quebec	1
	04/01/1996	LUCCHESSI	Ontario-South	3
	04/01/1996	LUCCHESSI	Manitoba	1
	04/01/1996	LEE	Ontario-South	8
	04/01/1996	LEE	Ontario-North	-
	04/01/1996	LEE	Quebec	8
	04/01/1996	LEE	Manitoba	9
	04/01/1996	GOUNOT	Ontario-South	3

样本数据库表

Name:	SALES_DATE	SALES_PERSON	REGION	SALES
	04/01/1996	GOUNOT	Ontario-North	1
	04/01/1996	GOUNOT	Quebec	3
	04/01/1996	GOUNOT	Manitoba	7

STAFF 表

Name:	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
Type:	smallint not null	varchar(9)	smallint	char(5)	smallint	dec(7,2)	dec(7,2)
Desc:	Employee number	Employee name	Department number	Job type	Years of service	Current salary	Commission
Values:	10	Sanders	20	Mgr	7	18357.50	-
	20	Pernal	20	Sales	8	18171.25	612.45
	30	Marengi	38	Mgr	5	17506.75	-
	40	O'Brien	38	Sales	6	18006.00	846.55
	50	Hanes	15	Mgr	10	20659.80	-
	60	Quigley	38	Sales	-	16808.30	650.25
	70	Rothman	15	Sales	7	16502.83	1152.00
	80	James	20	Clerk	-	13504.60	128.20
	90	Koonitz	42	Sales	6	18001.75	1386.70
	100	Plotz	42	Mgr	7	18352.80	-
	110	Ngan	15	Clerk	5	12508.20	206.60
	120	Naughton	38	Clerk	-	12954.75	180.00
	130	Yamaguchi	42	Clerk	6	10505.90	75.60
	140	Fraye	51	Mgr	6	21150.00	-
	150	Williams	51	Sales	6	19456.50	637.65
	160	Molinare	10	Mgr	7	22959.20	-
	170	Kermisch	15	Clerk	4	12258.50	110.10
	180	Abrahams	38	Clerk	3	12009.75	236.50
	190	Sneider	20	Clerk	8	14252.75	126.50
	200	Scoutten	42	Clerk	-	11508.60	84.20
	210	Lu	10	Mgr	10	20010.00	-
	220	Smith	51	Sales	7	17654.50	992.80
	230	Lundquist	51	Clerk	3	13369.80	189.65
	240	Daniels	10	Mgr	5	19260.25	-
	250	Wheeler	51	Clerk	6	14460.00	513.30
	260	Jones	10	Mgr	12	21234.00	-
	270	Lea	66	Mgr	9	18555.50	-
	280	Wilson	66	Sales	9	18674.50	811.50

Name:	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
	290	Quill	84	Mgr	10	19818.00	-
	300	Davis	84	Sales	5	15454.50	806.10
	310	Graham	66	Sales	13	21000.00	200.30
	320	Gonzales	66	Sales	4	16858.20	844.00
	330	Burke	66	Clerk	1	10988.00	55.50
	340	Edwards	84	Sales	7	17844.00	1285.00
	350	Gafney	84	Clerk	5	13030.50	188.00

STAFFG 表

注: 仅为双字节代码页创建 STAFFG。

Name:	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
Type:	smallint not null	varchar(9)	smallint	graphic(5)	smallint	dec(9,0)	dec(9,0)
Desc:	Employee number	Employee name	Department number	Job type	Years of service	Current salary	Commission
Values:	10	Sanders	20	Mgr	7	18357.50	-
	20	Pernal	20	Sales	8	18171.25	612.45
	30	Marenghi	38	Mgr	5	17506.75	-
	40	O'Brien	38	Sales	6	18006.00	846.55
	50	Hanes	15	Mgr	10	20659.80	-
	60	Quigley	38	Sales	-	16808.30	650.25
	70	Rothman	15	Sales	7	16502.83	1152.00
	80	James	20	Clerk	-	13504.60	128.20
	90	Koonitz	42	Sales	6	18001.75	1386.70
	100	Plotz	42	Mgr	7	18352.80	-
	110	Ngan	15	Clerk	5	12508.20	206.60
	120	Naughton	38	Clerk	-	12954.75	180.00
	130	Yamaguchi	42	Clerk	6	10505.90	75.60
	140	Fraye	51	Mgr	6	21150.00	-
	150	Williams	51	Sales	6	19456.50	637.65
	160	Molinare	10	Mgr	7	22959.20	-
	170	Kermisch	15	Clerk	4	12258.50	110.10
	180	Abrahams	38	Clerk	3	12009.75	236.50
	190	Sneider	20	Clerk	8	14252.75	126.50
	200	Scoutten	42	Clerk	-	11508.60	84.20
	210	Lu	10	Mgr	10	20010.00	-
	220	Smith	51	Sales	7	17654.50	992.80

样本数据库表

Name:	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
	230	Lundquist	51	Clerk	3	13369.80	189.65
	240	Daniels	10	Mgr	5	19260.25	-
	250	Wheeler	51	Clerk	6	14460.00	513.30
	260	Jones	10	Mgr	12	21234.00	-
	270	Lea	66	Mgr	9	18555.50	-
	280	Wilson	66	Sales	9	18674.50	811.50
	290	Quill	84	Mgr	10	19818.00	-
	300	Davis	84	Sales	5	15454.50	806.10
	310	Graham	66	Sales	13	21000.00	200.30
	320	Gonzales	66	Sales	4	16858.20	844.00
	330	Burke	66	Clerk	1	10988.00	55.50
	340	Edwards	84	Sales	7	17844.00	1285.00
	350	Gafney	84	Clerk	5	13030.50	188.00

数据类型为 BLOB 和 CLOB 的样本文件

此节显示 EMP_PHOTO 文件（雇员的相片）以及 EMP_RESUME 文件（雇员的简历）中的数据。

Quintana 的相片



图 7. Delores M. Quintana

Quintana 的简历

db200130.asc 和 db200130.scr 文件中有下列文本。

Resume: Delores M. Quintana

Personal Information

Address: 1150 Eglinton Ave Mellonville, Idaho 83725
Phone: (208) 555-9933
Birthdate: September 15, 1925
Sex: Female
Marital Status: Married
Height: 5'2"
Weight: 120 lbs.

Department Information

Employee Number: 000130
Dept Number: C01
Manager: Sally Kwan
Position: Analyst
Phone: (208) 555-4578
Hire Date: 1971-07-28

Education

1965 Math and English, B.A. Adelphi University
1960 Dental Technician Florida Institute of Technology

Work History

10/91 - present Advisory Systems Analyst Producing documentation tools for engineering department.
12/85 - 9/91 Technical Writer Writer, text programmer, and planner.
1/79 - 11/85 COBOL Payroll Programmer Writing payroll programs for a diesel fuel company.

Interests

- Cooking
- Reading
- Sewing
- Remodeling

Nicholls 的相片



图8. Heather A. Nicholls

Nicholls 的简历

db200140.asc 和 db200140.scr 文件中有下列文本。

Resume: Heather A. Nicholls

Personal Information

Address: 844 Don Mills Ave Mellonville, Idaho 83734
Phone: (208) 555-2310
Birthdate: January 19, 1946
Sex: Female
Marital Status: Single
Height: 5'8"
Weight: 130 lbs.

Department Information

Employee Number: 000140
Dept Number: C01
Manager: Sally Kwan
Position: Analyst
Phone: (208) 555-1793
Hire Date: 1976-12-15

Education

1972 Computer Engineering, Ph.D. University of Washington
1969 Music and Physics, M.A. Vassar College

Work History

2/83 - present

Architect, OCR Development Designing the architecture of OCR products.

12/76 - 1/83

Text Programmer Optical character recognition (OCR) programming in PL/I.

9/72 - 11/76

Punch Card Quality Analyst Checking punch cards met quality specifications.

Interests

- Model railroading
- Interior decorating
- Embroidery
- Knitting

Adamson 的相片

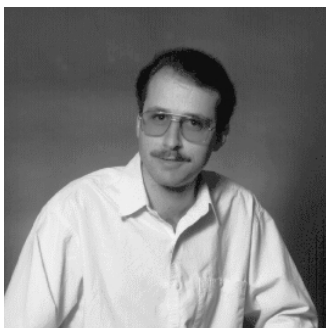


图 9. Bruce Adamson

Adamson 的简历

db200150.asc 和 db200150.scr 文件中有下列文本。

Resume: Bruce Adamson

Personal Information

Address: 3600 Steeles Ave Mellonville, Idaho 83757
Phone: (208) 555-4489
Birthdate: May 17, 1947
Sex: Male

样本数据库表

Marital Status: Married
Height: 6'0"
Weight: 175 lbs.

Department Information

Employee Number: 000150
Dept Number: D11
Manager: Irving Stern
Position: Designer
Phone: (208) 555-4510
Hire Date: 1972-02-12

Education

1971 Environmental Engineering, M.Sc. Johns Hopkins University
1968 American History, B.A. Northwestern University

Work History

8/79 - present Neural Network Design Developing neural networks for machine intelligence products.
2/72 - 7/79 Robot Vision Development Developing rule-based systems to emulate sight.
9/71 - 1/72 Numerical Integration Specialist Helping bank systems communicate with each other.

Interests

- Racing motorcycles
- Building loudspeakers
- Assembling personal computers
- Sketching

Walker 的相片

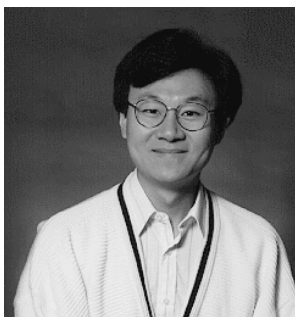


图 10. James H. Walker

Walker 的简历

db200190.asc 和 db200190.scr 文件中有下列文本。

Resume: James H. Walker

Personal Information

Address: 3500 Steeles Ave Mellonville, Idaho 83757
Phone: (208) 555-7325
Birthdate: June 25, 1952
Sex: Male
Marital Status: Single
Height: 5'11"
Weight: 166 lbs.

Department Information

Employee Number: 000190
Dept Number: D11
Manager: Irving Stern
Position: Designer
Phone: (208) 555-2986
Hire Date: 1974-07-26

Education

1974 Computer Studies, B.Sc. University of Massachusetts
1972 Linguistic Anthropology, B.A. University of Toronto

Work History

样本数据库表

6/87 - present

Microcode Design Optimizing algorithms for mathematical functions.

4/77 - 5/87

Printer Technical Support Installing and supporting laser printers.

9/74 - 3/77

Maintenance Programming Patching assembly language compiler for mainframes.

Interests

- Wine tasting
- Skiing
- Swimming
- Dancing

附录B. 使用 DB2 资料库

DB2 通用数据库由联机帮助、书籍（PDF 和 HTML）和 HTML 格式的样本程序组成。本节描述所提供的信息以及如何访问这些信息。

要访问联机产品信息，可以使用“信息中心”。有关详情，参见第104页的『用“信息中心”访问“信息”』。可以查看任务信息、DB2 书籍、疑难解答信息、样本程序和 Web 上的 DB2 信息。

DB2 PDF 文件和打印的书籍

DB2 信息

下表将 DB2 书籍分为四个类别：

DB2 指南和参考信息

这些书籍包含所有平台的公共 DB2 信息。

DB2 安装和配置信息

这些书籍是针对特定平台上的 DB2 的。例如，有分别针对 OS/2 平台、Windows 平台和基于 UNIX 的平台上 DB2 的快速入门书籍。

HTML 格式的跨平台样本程序

这些样本是与“应用程序开发客户机”一起安装的样本程序的 HTML 版本。样本仅供参考，并不替代实际程序。

发行说明

这些文件包含 DB2 书籍中未能包括的最新信息。

HTML 格式的安装手册、发行说明和教程可直接在产品 CD-ROM 上看到。大部分书籍在产品 CD-ROM 上都有 HTML 格式以便查看，而在 DB2 出版物 CD-ROM 上则有 Adobe Acrobat (PDF) 格式以便查看和打印。还可从 IBM 订购打印的副本；参见第100页的『订购打印书籍』。下表列示了可订购的书籍。

在 OS/2 和 Windows 平台上，可在 `sqllib\doc\html` 目录下安装 HTML 文件。DB2 信息被翻译成各种语言；但是，并非所有的信息都有每一种语言的翻译版本。每当信息不能以某种特定语言表示出来时，就会提供英语信息

在 UNIX 平台上，可在 `doc/%L/html`（其中 `%L` 表示本国语言环境）目录下安装多种语言版本的 HTML 文件。有关详情，参考适当的快速入门书籍。

您可使用以下各种方法来获取 DB2 书籍并访问其中的信息:

- 第103页的『查看联机信息』
- 第107页的『搜索联机信息』
- 第100页的『订购打印书籍』
- 第99页的『打印 PDF 书籍』

表 1. DB2 信息

名称	说明	书号	HTML 目录
		PDF 文件名	
DB2 指南和参考信息			
管理指南	<p>管理指南: 计划提供数据库概念的概述、有关设计问题 (如逻辑和物理数据库设计) 的信息, 以及高可用性的讨论。</p> <p>管理指南: 实现提供有关实现问题 (如实现设计、存取数据库、审核、备份和恢复) 的信息。</p> <p>管理指南: 性能提供有关数据库环境以及应用程序性能评估和调整的信息。</p> <p>在北美, 可使用书号 SBOF-8934 来订购三卷英文版的管理指南。</p>	<p>SB84-0219 db2d1x70</p> <p>SB84-0218 db2d2x70</p> <p>SB84-0243 db2d3x70</p>	db2d0
<i>Administrative API Reference</i>	描述 DB2 应用程序设计接口 (API) 以及您可以用来管理数据库的数据结构。此书还说明如何在应用程序中调用 API。	SC09-2947 db2b0x70	db2b0
应用程序构建指南	提供环境设置信息和关于如何在 Windows、OS/2 和基于 UNIX 的平台上编译、链接和运行 DB2 应用程序的逐步指导。	SB84-0220 db2axx70	db2ax
<i>APPC, CPI-C, and SNA Sense Codes</i>	提供关于使用DB2 通用数据库产品时可能遇到的 APPC、CPI-C 和 SNA 检测码的一般信息。	无书号 db2apx70	db2ap
仅有 HTML 格式的版本。			

表 1. DB2 信息 (续)

名称	说明	书号	HTML 目录
		PDF 文件名	
<i>Application Development Guide</i>	说明如何开发使用嵌入式 SQL 或 Java (JDBC 和 SQLJ) 来存取 DB2 数据库的应用程序。讨论主题包括在分区环境或联合体系统中编写存储过程、编写用户定义函数、创建用户定义类型、使用触发器和开发应用程序。	SC09-2949 db2a0x70	db2a0
<i>CLI Guide and Reference</i>	说明如何开发使用“DB2 调用层接口”(一个与 Microsoft ODBC 规范兼容的可调用 SQL 接口)来存取 DB2 数据库的应用程序。	SC09-2950 db2l0x70	db2l0
<i>Command Reference</i>	说明如何使用“命令行处理器”，并描述可用来管理数据库的 DB2 命令。	SC09-2951 db2n0x70	db2n0
<i>Connectivity Supplement</i>	提供有关以下各项的设置和参考信息：如何将作为 DRDA 应用程序请求器的 DB2 AS/400 版、DB2 OS/390 版、DB2 MVS 版、DB2 VM 版与 DB2 通用数据库服务器配合使用。此书还详述了如何将 DRDA 应用服务器与 DB2 Connect 应用程序请求器配合使用。 仅有 HTML 和 PDF 格式。	无书号 db2h1x70	db2h1
<i>Data Movement Utilities Guide and Reference</i>	说明如何使用 DB2 实用程序(如调入、调出、装入、自动装入程序和 DPROP)来使数据移动易于进行。	SC09-2955 db2dmx70	db2dm
数据仓库中心管理指南	提供有关如何使用“数据仓库中心”构建和维护数据仓库的信息。	SB84-0226 db2ddx70	db2dd
<i>Data Warehouse Center Application Integration Guide</i>	提供帮助程序员将应用程序与“数据仓库中心”和“信息目录管理程序”集成的信息。	SC26-9994 db2adx70	db2ad
<i>DB2 Connect 用户指南</i>	提供 DB2 Connect 产品的概念、程序设计以及一般用法信息。	SB84-0221 db2c0x70	db2c0
<i>DB2 Query Patroller Administration Guide</i>	提供 DB2 Query Patroller 系统的操作概述、特定操作和管理信息以及管理图形用户界面实用程序的任务信息。	SC09-2958 db2dwx70	db2dw

表 1. DB2 信息 (续)

名称	说明	书号	HTML 目录
		PDF 文件名	
<i>DB2 Query Patroller</i> 用户指南	描述如何使用 DB2 Query Patroller 的工具和功能。	SB84-0222	db2ww
		db2wwx70	
词汇表	提供 DB2 及其部件中使用的术语的定义。 有 HTML 格式可用且在 <i>SQL Reference</i> 中。	无书号	db2t0
		db2t0x70	
<i>Image, Audio 和 Video Extender</i> 管理和程序设计	提供有关 DB2 Extender 的一般信息, 有关 Image、Audio and Video (IAV) Extender 的管理和配置的信息, 以及有关使用 IAV Extender 进行程序设计的信息。它包括参考信息、诊断资料 (带有信息) 和样本。	SB84-0247	dmbu7
		dmbu7x70	
<i>Information Catalog Manager Administration Guide</i>	提供有关管理信息目录的指南。	SC26-9995	db2di
		db2dix70	
<i>Information Catalog Manager Programming Guide and Reference</i>	提供“信息目录管理程序”的体系结构接口的定义。	SC26-9997	db2bi
		db2bix70	
信息目录管理程序用户指南	提供有关使用“信息目录管理程序”用户界面的信息。	SB84-0227	db2ai
		db2aix70	
安装和配置补遗	指导您了解计划、安装和设置特定于平台的 DB2 客户机。此补遗还包含关于联编、设置客户机和服务器通信、DB2 GUI 工具、DRDA AS、分布式安装、配置分布式请求和存取多机种数据源的信息。	GB84-0127	db2iy
		db2iyx70	
信息参考	列出由 DB2、信息目录管理程序和数据仓库中心发出的信息和代码, 并描述应执行的操作。 在北美, 您可订购两卷英文版的信息参考 (使用书号 SBOF-8932)。	第 1 卷 GB84-0216	db2m0
		db2m1x70	
		第 2 卷 GB84-0217	
		db2m2x70	
<i>OLAP Integration Server Administration Guide</i>	说明如何使用“OLAP 集成服务器”的“管理程序”部件。	SC27-0787	n/a
		db2dpx70	

表 1. DB2 信息 (续)

名称	说明	书号	HTML 目录
		PDF 文件名	
<i>OLAP Integration Server Metaoutline User's Guide</i>	说明如何使用标准“OLAP 元轮廓”接口（而非通过使用“元轮廓辅助程序”）创建和填充 OLAP 元轮廓。	SC27-0784 db2upx70	n/a
<i>OLAP Integration Server Model User's Guide</i>	说明如何使用标准“OLAP 模型接口”（而非使用“模型辅助程序”）来创建 OLAP 模型。	SC27-0783 db2lpx70	n/a
<i>OLAP Setup and User's Guide</i>	提供 OLAP Starter Kit 的配置和设置信息。	SC27-0702 db2ipx70	db2ip
<i>OLAP Spreadsheet Add-in 用户指南 Excel 版</i>	描述如何使用 Excel 电子表格程序来分析 OLAP 数据。	SA40-1756 db2epx70	db2ep
<i>OLAP Spreadsheet Add-in 用户指南 Lotus 1-2-3 版</i>	描述如何使用 Lotus 1-2-3 电子表格程序来分析 OLAP 数据。	SA40-1757 db2tpx70	db2tp
<i>Replication Guide and Reference</i>	提供随 DB2 提供的“IBM 复制”工具的计划、配置、管理和用法信息。	SC26-9920 db2e0x70	db2e0
<i>Spatial Extender 用户指南和参考</i>	提供关于空间扩展程序的安装、配置、管理、程序设计和疑难解答的信息。还提供空间数据概念的重要说明，并提供空间扩展程序特定的参考资料（信息和 SQL）。	SB84-0249 db2sbx70	db2sb
<i>SQL 入门</i>	介绍 SQL 概念，并提供许多构造和任务的示例。	SB84-0223 db2y0x70	db2y0
<i>SQL Reference, 第 1 卷和第 2 卷</i>	描述 SQL 语法、语义和语言规则。此书还包括关于发行版间的不兼容性、产品限制和目录视图的信息。 在北美，可使用书号 SBOF-8933 来订购两卷英文版的 <i>SQL Reference</i> 。	第 1 卷 SC09-2974 db2s1x70 第 2 卷 SC09-2975 db2s2x70	db2s0
<i>System Monitor Guide and Reference</i>	描述如何收集关于数据库和数据库管理程序的各种信息。此书说明如何利用信息来了解数据库活动、提高性能和确定问题的原因。	SC09-2956 db2f0x70	db2f0

表 1. DB2 信息 (续)

名称	说明	书号	HTML 目录
		PDF 文件名	
<i>Text Extender</i> 管理和程序设计	提供有关 DB2 Extender 的一般信息, 有关 Text Extender 的管理和配置的信息, 以及有关使用 Text Extender 进行程序设计的信息。它包括参考信息、诊断资料 (带有信息) 和样本。	SB84-0248	desu9
		desu9x70	
<i>Troubleshooting Guide</i>	帮助您确定错误源、从问题中恢复并向 “DB2 客户服务” 咨询以使用诊断工具。	GC09-2850	db2p0
		db2p0x70	
新增内容	描述 DB2 通用数据库 (版本 7) 中的新特性、函数和增强功能。	SB84-0224	db2q0
		db2q0x70	
DB2 安装和配置信息			
<i>DB2 Connect Enterprise Edition for OS/2 and Windows Quick Beginnings</i>	提供 OS/2 和 Windows 32 位操作系统上的 DB2 Connect 企业版的计划、迁移、安装和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GC09-2953	db2c6
		db2c6x70	
<i>DB2 Connect Enterprise Edition for UNIX Quick Beginnings</i>	提供基于 UNIX 的平台上的 DB2 Connect 企业版的计划、迁移、安装、配置和任务信息。此书还包含许多受支持的客户机的安装和设置信息。	GC09-2952	db2cy
		db2cyx70	
<i>DB2 Connect</i> 个人版快速入门	提供 OS/2 和 Windows 32 位操作系统上的 DB2 Connect 个人版的计划、迁移、安装、配置和任务信息。此书还包含所有受支持的客户机的安装和设置信息。	GB84-0212	db2c1
		db2c1x70	
DB2 Connect Personal Edition Quick Beginnings Linux 版	在进行所有受支持的 Linux 分发时, 提供 “DB2 Connect 个人版” 的计划、安装、迁移和配置信息。	GC09-2962	db2c4
		db2c4x70	
<i>DB2 Data Links</i> 管理程序快速入门	提供 “DB2 Data Links Manager AIX 版” 和 Windows 32 位操作系统的计划、安装、配置和任务信息。	GB84-0211	db2z6
		db2z6x70	
<i>DB2</i> 扩充企业版 UNIX 版快速入门	提供在基于 UNIX 的平台上的 DB2 扩充企业版的计划、安装和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GB84-0209	db2v3
		db2v3x70	

表 1. DB2 信息 (续)

名称	说明	书号	HTML 目录
		PDF 文件名	
<i>DB2 Enterprise - Extended Edition for Windows Quick Beginnings</i>	提供 DB2 扩充企业版 Windows 32 位操作系统版的计划、安装和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GC09-2963 db2v6x70	db2v6
<i>DB2 (OS/2 版) Quick Beginnings</i>	提供 OS/2 操作系统上的 DB2 通用数据库的计划、安装、迁移和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GC09-2968 db2i2x70	db2i2
<i>DB2 (UNIX 版) 快速入门</i>	提供在基于 UNIX 的平台上的 DB2 通用数据库的计划、安装、迁移和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GB84-0214 db2ixx70	db2ix
<i>DB2 Windows 版快速入门</i>	提供 Windows 32 位操作系统上的 DB2 通用数据库的计划、安装、迁移和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GB84-0215 db2i6x70	db2i6
<i>DB2 个人版快速入门</i>	提供 OS/2 和 Windows 32 位操作系统上的“DB2 通用数据库个人版”的计划、安装、迁移和配置信息。	GB84-0213 db2i1x70	db2i1
DB2 Personal Edition Quick Beginnings for Linux	在进行所有受支持的 Linux 分发时，提供“DB2 通用数据库个人版”的计划、安装、迁移和配置信息。	GC09-2972 db2i4x70	db2i4
<i>DB2 Query Patroller 安装指南</i>	提供有关 DB2 Query Patroller 的安装信息。	GB84-0208 db2iwx70	db2iw
DB2 Warehouse Manager 安装指南	提供仓库代理程序、仓库变换器和“信息目录管理程序”的安装信息。	GB84-0122 db2idx70	db2id
HTML 格式的跨平台样本程序			

表 1. DB2 信息 (续)

名称	说明	书号	HTML 目录
		PDF 文件名	
HTML 格式的样本程序	为所有受 DB2 支持的平台上的程序设计语言提供 HTML 格式的样本程序。提供的样本程序仅供参考。并非所有样本都有所有程序设计语言的版本。HTML 样本仅当安装了“DB2 应用程序开发客户机”时才可用。 有关这些程序的详情，参考应用程序构建指南。	无书号	db2hs
发行说明			
<i>DB2 Connect</i> 发行说明	提供 DB2 书籍中未能包括的最新信息。	参见注释 2。	db2cr
<i>DB2</i> 安装注释	提供 DB2 书籍中未能包括的最新安装特定信息。	仅在产品 CD-ROM 上提供。	
<i>DB2</i> 发行说明	提供 DB2 书籍中未能包括的、有关所有 DB2 产品和功能部件的最新信息。	参见注释 2。	db2ir

注:

1. 文件名第六个位置的字符 *x* 指示书籍的语言版本。例如，文件名 db2d0e70 标识英语版本的管理指南，而文件名 db2d0f70 标识同一本书的法语版本。下列字母用在文件名的第六个位置以指示语言版本:

语言	标识符
巴西葡萄牙语	b
保加利亚语	u
捷克语	x
丹麦语	d
荷兰语	q
英语	e
芬兰语	y
法语	f
德语	g
希腊语	a
匈牙利语	h
意大利语	i
日语	j
韩国语	k
挪威语	n
波兰语	p

葡萄牙语	v
俄语	r
简体中文	c
斯洛文尼亚语	l
西班牙语	z
瑞典语	s
繁体中文	t
土耳其语	m

2. DB2 书籍中未能包括的最新信息以 HTML 格式在“发行说明”中提供，或作为 ASCII 文件提供。在“信息中心”中和产品 CD-ROM 上都提供了 HTML 版本。要查看 ASCII 文件：

- 在基于 UNIX 的平台上，参见 Release.Notes 文件。此文件位于 DB2DIR/Readme/%L 目录中，其中 %L 表示本国语言环境名，而 DB2DIR 表示：
 - 在 AIX 上，是 /usr/lpp/db2_07_01
 - 在 HP-UX、PTX、Solaris 和 Silicon Graphics IRIX 上，是 /opt/IBMdb2/V7.1
 - 在 Linux 上，是 /usr/IBMdb2/V7.1。
- 在其它平台上，参见 RELEASE.TXT 文件。此文件在安装产品的目录中。在 OS/2 平台上，还可双击 **IBM DB2** 文件夹，然后双击**发行说明**图符。

打印 PDF 书籍

如果想要书籍的打印副本，则可打印 DB2 出版物 CD-ROM 上的 PDF 文件。使用 Adobe Acrobat Reader，可打印整本书籍或特定范围内的页。有关库中每本书的文件名，参见第92页的表1。

可从 Adobe Web 站点（网址 <http://www.adobe.com>）获取 Adobe Acrobat Reader 的最新版本。

这些 PDF 文件包括在 DB2 出版物 CD-ROM 上，文件扩展名为 PDF。要访问这些 PDF 文件：

1. 插入 DB2 出版物 CD-ROM。在基于 UNIX 的平台上，安装 DB2 出版物 CD-ROM。参考快速入门一书以了解安装过程。
2. 启动 Acrobat Reader。
3. 从下列位置之一打开期望的 PDF 文件：
 - 在 OS/2 和 Windows 平台上：
 - x:\doc\language* 目录，其中 *x* 表示 CD-ROM 驱动器而 *language* 表示两个字符的国家代码，它表示您所用的语言（例如，EN 表示英语）。

- 在基于 UNIX 的平台上:

CD-ROM 上的 `/cdrom/doc/%L` 目录, 其中 `/cdrom` 表示 CD-ROM 的安装点而 `%L` 表示期望的本国语言环境的名称。

还可从 CD-ROM 将 PDF 文件复制至本地或网络驱动器并从该处读取它们。

订购打印书籍

可通过使用销售单 (SBOF) 书号单本地或成套地订购打印的 DB2 书籍 (仅限北美)。要订购书籍, 与 IBM 授权经销商或市场代表联系, 或致电 1-800-879-2755 (美国) 或 1-800-IBM-4YOU (加拿大)。还可从 Publications Web 页 (网址为 <http://www.elink.ibm.link.ibm.com/pbl/pbl>) 订购这些书籍。

有两套书籍。SBOF-8935 提供了“DB2 仓库管理程序”的参考和用法信息。SBOF-8931 提供了所有其他“DB2 通用数据库”产品和功能部件的参考和用法信息。每个 SBOF 的内容列示在下表中:

表 2. 订购打印书籍

SBOF 号	包括的书籍
SBOF-8931	<ul style="list-style-type: none"> • 管理指南: 计划 • 管理指南: 实现 • 管理指南: 性能 • 管理 API 参考 • 应用程序构建指南 • 应用程序开发指南 • CLI Guide and Reference • 命令参考 • Data Movement Utilities Guide and Reference • 数据仓库中心管理指南 • Data Warehouse Center Application Integration Guide • DB2 Connect 用户指南 • 安装和配置补遗 • Image, Audio, and Video Extenders 管理和程序设计 • 信息参考, 第 1 卷和第 2 卷 • OLAP Integration Server Administration Guide • OLAP Integration Server Metaoutline User's Guide • OLAP Integration Server Model User's Guide • OLAP Integration Server User's Guide • OLAP Setup and User's Guide • OLAP Spreadsheet Add-in 用户指南 Excel 版 • OLAP Spreadsheet Add-in 用户指南 Lotus 1-2-3 版 • Replication Guide and Reference • Spatial Extender Administration and Programming Guide • SQL 入门 • SQL Reference, 第 1 卷和第 2 卷 • System Monitor Guide and Reference • Text Extender 管理和程序设计 • Troubleshooting Guide • 新增内容
SBOF-8935	<ul style="list-style-type: none"> • Information Catalog Manager Administration Guide • 信息目录管理程序用户指南 • Information Catalog Manager Programming Guide and Reference • Query Patroller 管理指南 • Query Patroller 用户指南

DB2 联机文档

访问联机帮助

随所有 DB2 部件都附带提供了联机帮助。下表描述了各种类型的联机帮助。

帮助类型	内容	如何访问...
命令帮助	说明命令行处理器中命令的语法。	<p>从命令行处理器，以交互方式输入： <code>? command</code></p> <p>其中 <code>command</code> 表示一个关键字或整个命令。</p> <p>例如，<code>? catalog</code> 显示所有 CATALOG 命令的帮助，而 <code>? catalog database</code> 显示 CATALOG DATABASE 命令的帮助。</p>
客户机配置辅助程序帮助	说明您可在窗口或笔记本中执行的任务。此帮助包括您需要知道的概述和前提条件信息，并描述如何使用窗口或笔记本控件。	从窗口或笔记本，单击帮助按钮或按 F1 键。
命令中心帮助		
控制中心帮助		
数据仓库中心帮助		
事件分析程序帮助		
信息目录管理程序帮助		
卫星管理中心帮助		
脚本中心帮助		
信息帮助	描述信息的起因以及您应该执行的任何操作。	<p>从命令行处理器，以交互方式输入： <code>? XXXnnnnn</code></p> <p>其中 <code>XXXnnnnn</code> 表示有效的信息标识符。</p> <p>例如，<code>? SQL30081</code> 显示关于 SQL30081 信息的帮助。</p> <p>要每次查看一屏信息帮助，可输入： <code>? XXXnnnnn 尚有</code></p> <p>要在文件中保存信息帮助，可输入： <code>? XXXnnnnn > filename.ext</code></p> <p>其中 <code>filename.ext</code> 表示想要保存信息帮助的文件。</p>

帮助类型	内容	如何访问...
SQL 帮助	说明 SQL 语句的语法。	<p>从命令行处理器，以交互方式输入：</p> <pre>help statement</pre> <p>其中，<i>statement</i> 表示 SQL 语句。</p> <p>例如，<code>help SELECT</code> 显示有关 SELECT 语句的帮助。</p> <p>注：在基于 UNIX 的平台上，SQL 帮助不可用。</p>
SQLSTATE 帮助	说明 SQL 状态及类代码。	<p>从命令行处理器，以交互方式输入：</p> <pre>? sqlstate 或 ? class code</pre> <p>其中，<i>sqlstate</i> 表示有效的五位 SQL 状态，而 <i>class code</i> 表示该 SQL 状态的头两位。</p> <p>例如，<code>? 08003</code> 显示 08003 SQL 状态的帮助，而 <code>? 08</code> 显示 08 类代码的帮助。</p>

查看联机信息

此产品中的书籍为超文本标记语言 (HTML) 软拷贝格式。软拷贝格式使您可搜索或浏览信息，并提供访问相关信息的超文本链接。它还使得在站点间共享库更容易。

可使用遵循 HTML 版本 3.2 规范的任何浏览器来查看联机书籍或样本程序。

要查看联机书籍或样本程序：

- 如果正在运行 DB2 管理工具，则使用“信息中心”。
- 从浏览器，单击**文件** → **打开页**。打开的页中包含 DB2 信息的描述和至 DB2 信息的链接：

- 在基于 UNIX 的平台上，打开以下页：

```
INSTHOME/sql1lib/doc/%L/html/index.htm
```

其中 *%L* 表示本国语言环境名称

- 在其它平台上，打开以下页：

```
sql1lib\doc\html\index.htm
```

该路径位于安装了 DB2 的驱动器上。

如果尚未安装“信息中心”，则可通过双击 **DB2 信息** 图符来打开该页。视您正在使用的系统不同，图符在主产品文件夹中或在“Windows 开始”菜单中。

安装 Netscape 浏览器

如果还未安装 Web 浏览器，则可从产品包装箱中的 Netscape CD-ROM 安装 Netscape。要获取如何安装它的详细指导，执行：

1. 插入 Netscape CD-ROM。
2. 安装 CD-ROM（仅限于在基于 UNIX 的平台上）。参考快速入门一书以了解安装过程。
3. 有关安装指导，参考 CDNAV *nn.txt* 文件，其中 *nn* 表示两字符语言标识符。该文件位于 CD-ROM 的根目录下。

用“信息中心”访问“信息”

“信息中心”提供对 DB2 产品信息的快速访问。在所有装有 DB2 管理工具的平台，都提供了“信息中心”。

可通过双击“信息中心”图符来打开“信息中心”。视正在使用的系统的不同，该图符在主产品文件夹的“信息”文件夹中，或在 Windows 的开始菜单中。

还可通过使用工具栏和 DB2 Windows 平台上的帮助菜单来访问“信息中心”。

“信息中心”提供了六种类型的信息。单击适当的标签来查看提供给该类型的主题。

任务	可使用 DB2 执行的关键任务。
参考	DB2 参考信息，如关键字、命令以及 API。
书籍	DB2 书籍。
疑难解答	错误信息类别及其恢复操作。
样本程序	随“DB2 应用程序开发客户机”一起提供的样本程序。如果未安装“DB2 应用程序开发客户机”，则不显示此标签。
Web	万维网（WWW）上的 DB2 信息。要访问此信息，必须从系统连接至 Web。

当选择其中一个列表中的项时，“信息中心”启动一个查看器来显示信息。视所选择的信息种类的不同，查看器可能是系统帮助查看器、编辑器或 Web 浏览器。

“信息中心”提供了查找功能部件，因此您不用浏览这些列表就能查找特定主题。

对于全文本搜索，请遵循“信息中心”中指向搜索 DB2 联机信息搜索表格的超文本链接。

HTML 搜索服务器通常是自动启动的。如果 HTML 信息中的搜索不起作用，则可能必须使用下列其中一个方法来启动搜索服务器：

在 Windows 上

单击**开始**并选择程序 → **IBM DB2** → **信息** → **启动 HTML 搜索服务器**。

在 OS/2 上

双击 **DB2 OS/2** 版文件夹，然后双击**启动 HTML 搜索服务器**图符。

如果在搜索 HTML 信息时遇到任何其它问题，可参考发行说明。

注：搜索功能在 Linux、PTX 和 Silicon Graphics IRIX 环境中不可用。

使用 DB2 向导

向导通过让您一次一步地完成每一个任务来协助您完成特定管理任务。可通过控制中心和客户机配置辅助程序来获取向导。下表列出了这些向导并描述了它们的用途。

注：“创建数据库”、“创建索引”、“配置多站点更新”和“性能配置”向导对分区数据库环境可用。

向导	帮助您...	如何访问...
添加数据库	在客户机工作站上编目数据库。	从“客户机配置辅助程序”单击添加。
备份数据库	确定、创建并调度应急计划。	从“控制中心”，用鼠标右键单击想要备份的数据库并选择 备份 → 数据库 （使用向导）。
配置多站点更新	配置多站点更新、分布式事务或两阶段落实。	从“控制中心”，用鼠标右键单击 数据库 文件夹并选择 多站点更新 。
创建数据库	创建数据库并执行一些基本配置任务。	从“控制中心”，用鼠标右键单击 数据库 文件夹，并选择 创建 → 数据库 （使用向导）。
创建表	选择基本数据类型并创建表的主关键字。	从“控制中心”，用鼠标右键单击 表 图符，并选择 创建 → 表 （使用向导）。
创建表空间	创建新的表空间。	从“控制中心”，用鼠标右键单击 表空间 图符，并选择 创建 → 表空间 （使用向导）。

向导	帮助您...	如何访问...
创建索引	建议对于所有查询要创建和卸下哪些索引。	从“控制中心”，用鼠标右键单击索引图符，并选择 创建 → 索引（使用向导） 。
性能配置	通过更新配置参数来调整数据库性能以满足您的业务需求。	从“控制中心”，用鼠标右键单击想要调整的数据库并选择 使用向导配置性能 。 对于分区数据库环境，从“数据库分区”视图，用鼠标右键单击想要调整的首个数据库分区并选择 使用向导配置性能 。
复原数据库	在故障之后恢复数据库。它帮助您了解要使用的备份及要重放的纪录。	从“控制中心”，用鼠标右键单击想要复原的数据库并选择 复原 → 数据库（使用向导） 。

设置文档服务器

在缺省情况下，DB2 信息安装在本地系统上。这表示需要存取 DB2 信息的每个人都必须安装相同的文件。要将 DB2 信息存储在单个位置中，执行下列步骤：

1. 将所有文件和子目录从本地系统上的 `\sqllib\doc\html` 复制至 Web 服务器。每一本书都有其自己的子目录，该子目录包含构成该书的所有必需的 HTML 和 GIF 文件。确保目录结构仍相同。
2. 配置 Web 服务器以查找新位置中的文件。有关信息，可参考**安装和配置补遗**中的 NetQuestion 附录。
3. 如果正在使用“信息中心”的 Java 版本，可为所有 HTML 文件指定基本的 URL。您应将该 URL 用于书籍列表。
4. 当能够查看书籍文件时，可将经常查看的主题做成书签。您可能想把下列各页做成书签：
 - 书籍列表
 - 经常使用的书籍的目录
 - 经常引用的文章，如 ALTER TABLE 主题
 - 搜索格式

有关如何从中央机器处理 DB2 通用数据库联机文档文件的信息，参考**安装和配置补遗**中的 NetQuestion 附录。

搜索联机信息

要查找 HTML 文件中的信息，使用下列方法之一：

- 在顶部框中单击**搜索**。使用搜索格式来查找特定的主题。此功能在 Linux、PTX 和 Silicon Graphics IRIX 环境中不可用。
- 在顶部框中单击**索引**。使用索引来查找书中的特定主题。
- 显示帮助或 HTML 书籍的目录或索引，然后使用 Web 浏览器的查找功能查找书中的特定主题。
- 使用 Web 浏览器的书签功能来快速返回至特定的主题。
- 使用“信息中心”的搜索功能来查找特定的主题。参见第104页的『用“信息中心”访问“信息”』以获取详情。

附录C. 注意事项

IBM 可能未在所有国家中提供本文档中讨论的产品、服务或功能部件。关于您所在区域目前可用的产品及服务的信息，请向当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并不说明或暗示只能使用 IBM 的产品、程序或服务。凡是同等功能的产品、程序或服务，只要不侵犯 IBM 的知识产权，都可以用来替代 IBM 产品、程序或服务。当然，评估和验证非 IBM 产品、程序或服务均由用户自行负责。

本文档的议题可能涉及 IBM 的某些专利或正在申请中的专利的应用。提供本文档，并不表示允许您使用这些专利。您可以将许可证查询以书面形式发送给：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

关于双字节 (DBCS) 许可证查询的信息，请与您所在国家的 IBM 知识产权部门联系，将查询以书面形式发送至寄往：

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

以下段落不适用于英国与其它当地法律不允许这种供应方式的国家：国际商用机器公司『按原样』出版此书，不做任何明确或暗示的担保，包括但不限于不侵权、可销售或适用于特殊目的暗示性担保。一些地区在某些事务中不允许否认拒绝明确或暗示的担保，因此本条款可能不适合您。

本信息中可能有技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些信息将包含在本书新的版本中。IBM 可以随时对本书中说明的产品和/或程序进行改进和/或改动，而不必通知您。

此信息中对非 IBM Web 站点的任何引用仅是为了方便起见，而不以任何方式为那些 Web 站点作保证。那些 Web 站点的资料并非此 IBM 产品资料的一部分，使用那些 Web 站点的风险由您自己承担。

对于您所提供的任何信息，IBM 有权利以任何她认为适当的方式使用或散发，而不必对您负任何责任。

为了以下目的：(1) 允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换 (2) 允许对已经交换的信息进行相互使用，而希望获取本程序有关信息的合法用户请与下列地址联系：

IBM Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

只要遵守适当的条款和条件，包括某些情形下的一定数量的付款，都可获取这方面的信息。

这些信息中描述的特许程序及其所有可用的特许资料，按 IBM 客户协议、IBM 国际程序许可证协议或任何等价的协议中的条款，由 IBM 提供。

此处包含的所有性能数据都是在受控环境中确定的。因此，在其他操作环境中获得的结果可能与之相差很大。某些测量可能是在开发级的系统上进行的，不能保证这些测量方法在通用系统上同样可用。此外，某些测量方法可能是通过外推法归纳来估计的。实际结果可能会有所不同。此文档的用户应针对他们的特定环境验证数据是否适用。

涉及非 IBM 产品的信息可从这些产品的供应商、其发行公告或其它公众可用源得到。IBM 未测试这些产品，因此不能确认性能的精确度、兼容性或其它对非 IBM 产品的索赔赔偿要求等。有关非 IBM 产品功能方面的问题可向它们的供应商提出。

所有关于 IBM 未来方向或意向的声明都可能随时更改或撤消，而不作任何通知，并且仅代表发展目标。

此信息包含了用于日常商业处理的数据和报表的示例。为了尽可能完整地说明问题，这些示例中包含了个人、公司、品牌和产品的名称。所有这些名称都是虚构的，如与实际商业企业所使用的名称和地址相似，纯属巧合。

版权许可证：

本信息中可能包含用源语言编写的示例应用程序，它们说明了各种不同的操作平台上的程序设计技术。您可以为了开发、使用、市场营销或分发应用程序(这些应用程序遵守编写这些示例程序的操作平台的应用程序接口)的目的，以任何形式复

制、修改和分发这些示例程序，不用向 IBM 付费。这些例子未经所有条件下的完整测试。因此，IBM 不能保证或暗示其可靠性、可用性或这些程序的功能。

这些样本程序或任何派生产品的每个副本或任何部分必须包含如下的版权公告：

©（您的公司名称）（年度）。此代码各部分派生自“IBM 公司样本程序”。© Copyright IBM Corp. _输入年份_。All rights reserved.

注册商标

以星号 (*) 标出的下列术语是 IBM 公司在美国和 / 或其他国家的商标。

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/DS
DB2 Extender	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational	SystemView
Database Architecture	VisualAge
DRDA	VM/ESA
eNetwork	VSE/ESA
Extended Services	VTAM
FFST	WebExplorer
First Failure Support Technology	WIN-OS/2

下列各项是其他公司的商标或注册商标：

Microsoft、Windows、和 Windows NT 是 Microsoft 公司的商标或注册商标。

Java 或所有基于 Java 的商标和标志以及 Solaris 是 Sun Microsystems 公司在美国和 / 或其他国家的商标。

Tivoli 和 NetView 是 Tivoli Systems 公司在美国和 / 或其他国家的商标。

UNIX 是经 X/Open 有限公司唯一许可的在美国和 / 或其它国家的注册商标。

以双星号 (**) 标出的其他公司、产品或服务名, 可能是其他公司的商标或服务标志。

索引

[A]

安装
Netscape 浏览器 104

[B]

保留模式 4
备份数据库向导 105
标量函数 27
ABS 28
DECIMAL 35
HEX 28
LENGTH 28
SIGN 28
YEAR 28
标量全查询
说明 33
表
定义 3
函数 29
基表 3
结果表 3
外部关键字 52
唯一关键字 52
唯一约束 52
限定列名 37
样本数据库 71
主关键字 52
组合数据（连接） 25
表表达式
说明 35
表达式 24
表达式，命名 25
表函数
SQLCACHE_SNAPSHOT 29
表和视图之间的关系 13
表检查约束
定义 51
说明 53
延期约束检查 53

[C]

擦除样本数据库 72
参考完整性约束
定义 51
父关键字 52
说明 52
外部关键字 52
测试存在性 49
测试，存在性 49
查看
联机信息 103
查询，连接 47
除去重复行 23
触发器
定义 51
后触发器 54
前触发器 54
说明 54
转换变量 56
CREATE TRIGGER 54
创建表空间向导 105
创建表向导 105
创建数据库向导 105
创建样本数据库 72
错误信息
信息标识符 18
SQLSCODE 18
SQLSTATE 18

[D]

打印 PDF 书籍 99
大对象位置，定义 67
单值数据类型 65
递归查询，说明 62
定位器 67
多节点关系数据库，定义 1

[E]

二进制整数，说明 4

[F]

发行说明 99
分区关系数据库，定义 1
符号，作为数字属性 4
父关键字，定义 52
复原向导 106

[G]

公共表表达式
说明 36
关键字
定义 51
外部 52
唯一的 52
主 52
组合 51
关系数据库，定义 1
管理指南 v

[H]

函数
标量 27
表 29
联机分析处理 (OLAP) 62
列 27
内部 27
说明 27
用户定义 27
行
定义 3
选择 20
合并查询结果 45

[J]

基表 3
检索数据 19
将行进行排序 22
将列分组，定义 29

交叉积 57
交叉制表行 61
交互式 SQL, 定义 1
结构化查询语言(SQL), 定义 1
结果表 3
精度, 作为数字属性 4

[K]

空值 45
 删除列值 12
空值, 说明 4
快速Quick Beginnings v

[L]

联机帮助 101
联机分析处理 62
联机信息
 查看 103
 搜索 107
连接
 定义 25
 交叉积 57
 连接条件 57
 数据转换 59
 无连接条件 57
 相关子查询 41
连接查询 47
连接条件 57
列
 定义 3
 ASC, 升序排序 22
 DESC, 降序排序 22
列函数 27
 AVG 27
 COUNT 27
 MAX 27
 MIN 27

[M]

命令行处理器 1
模式
 定义 4

[N]

内连接 57

[P]

配置多站点更新向导 105

[Q]

嵌套表表达式, 说明 35
嵌套相关子查询 42
全查询 33
 与 INSERT 语句 10
 子查询 10, 50
 ALL 关键字 50
 ANY 关键字 50
全查询, 定义 10
权限 ID 4

[R]

日期时间值, 说明 4

[S]

设置文档服务器 106
十进制, 说明 4
视图
 说明 3
 限定列名 37
 优点 3
书籍 91, 100
数据结构
 行 3
 列 3
 值 3
数据库管理程序 1
数据类型
 单值 65
 BIGINT 4
 CHAR 4
 DATE 4
 DATETIME 4
 DECIMAL 4
 DOUBLE 4
 FLOAT 4

数据类型 (续)

INTEGER 4
REAL 4
SMALLINT 4
TIME 4
TIMESTAMP 4
VARCHAR 4

数据转换

 集合运算符 47
 连接条件 59
数字, 说明 4
搜索
 联机信息 104, 107
搜索条件 20
算术运算符 24
索引向导 105

[T]

添加数据库向导 105, 106
条件表达式
 说明 34
 SIGN 函数 34
通过视图修改表 14
 WITH CHECK OPTION 14
图形字符串
 变长 4
 定长 4

[W]

外部标量函数 66
外部表函数 66
外部关键字 52
外层查询, 相关 40
外层谓词 50
外连接
 说明 57
 FULL OUTER 连接 57
 LEFT OUTER 连接 57
 RIGHT OUTER 连接 57
唯一关键字 52
 唯一约束 52
唯一约束 52
 定义 51
谓词
 IS NOT NULL 20

谓词 (续)
IS NULL 20

[X]

系统目录 68
限定对象 4, 17
相关
名称 40
使用连接的子查询 41
说明 37
子查询 38
相关名
规则 37
列名的限定引用 37
相关文档 v
相关引用, 说明 38
相关子查询
何时使用 40
说明 38
向导
备份数据库 105
创建表 105
创建表空间 105
创建数据库 105
复原数据库 106
配置多站点更新 105
索引 105
添加数据库 105, 106
完成任务 105
性能配置 106
小计行 61
信息中心 104
性能配置向导 106
选择列表 19

[Y]

样本表 71, 91
样本程序
跨平台 97
HTML 97
样本数据库 71
擦除 72
创建 72
应用程序开发指南 v
用户定义函数 66

用户定义函数 66 (续)
定义 66
外部标量函数 66
外部表函数 66
源函数 66
OLE DB 外部表函数 66
用于子查询的比较运算符 50
语言标识符
书籍 98
源函数 66
约束
参考约束 9
对于集合运算符 47
唯一约束 9
运算次序 24, 27

[Z]

值
定义 3
主关键字 52
专用寄存器 68
CURRENT DATE 68
CURRENT DEGREE 68
CURRENT FUNCTION PATH 68
CURRENT PATH 68
CURRENT SERVER 68
CURRENT TIME 68
CURRENT TIMESTAMP 68
CURRENT TIMEZONE 68
USER 68
转换数据类型
说明 33
子查询
定义 26
字符串
变长 4
定长 4
作为数据类型 4
LOB 67
组合关键字 51
组合, 查询 45
最新信息 99

A

ADD CONSTRAINT 语句 53
ALL, 用于查询 50

ALTER TABLE 语句 53
ANY 关键字 50
AS 子句 25

B

BETWEEN 谓词 48
BIGINT, 数据类型 4
BLOB 数据类型 67
BLOB 字符串 67

C

CHAR, 数据类型 4
CLOB 数据类型 67
CLOB 字符串 67
CL_SCHED 样本表 72
CONNECT 语句 18
显式 18
隐式 18
CREATE DISTINCT TYPE 65
CREATE FUNCTION 66
CREATE TABLE 语句 9
列的 NOT NULL/NOT NULL
WITH DEFAULT 值 9
CREATE TRIGGER 54
CREATE VIEW 语句 13
WITH CHECK OPTION 13
CUBE 61
交叉制表行 61
小计行 61
CURRENT DATE 专用寄存器 68
CURRENT FUNCTION PATH 专用寄
存器 68
CURRENT SERVER 专用寄存器 68
CURRENT TIME 专用寄存器 68
CURRENT TIMESTAMP 专用寄存器
68
CURRENT TIMEZONE 专用寄存器
68

D

DATE, 数据类型 4
DATETIME, 数据类型 4
DB2 资料库
查看联机信息 103

DB2 资料库 (续)
 打印 PDF 书籍 99
 订购打印书籍 100
 结构 91
 联机帮助 101
 设置文档服务器 106
 书籍 91
 书籍的语言标识符 98
 搜索联机信息 107
 向导 105
 信息中心 104
 最新信息 99
DBLOB 数据类型 67
DBLOB 字符串 67
DECIMAL, 数据类型 4
DELETE 语句 12
DEPARTMENT 样本表 73
DISTINCT 关键字 23, 28
DOUBLE, 数据类型 4

E

EMPLOYEE 样本表 73
EMP_ACT 样本表 76
EMP_PHOTO 样本表 78
EMP_RESUME 样本表 78
EXCEPT 运算符 46
 排序结果 47
 涉及的用法约束 47
 数据类型 47
EXCEPT ALL 46
EXISTS 谓词 49

F

FLOAT, 数据类型 4
FROM 子句 19
FULL OUTER 连接 57

G

GROUP BY 24
GROUP BY 子句
 分组列 29
 与 HAVING 子句 30

H

HAVING 24

HAVING 子句
 说明 30
HTML
 样本程序 97

I

IN 谓词 48
INSERT 语句 10
 列的 NOT NULL/NOT NULL
 WITH DEFAULT 值 10
INTEGER, 数据类型 4
INTERSECT 运算符 47
 排序结果 47
 涉及的用法约束 47
 数据类型 47
INTERSECT ALL 47
IN_TRAY 样本表 79

L

LEFT OUTER 连接 57
LIKE 谓词 49
LOB
 定位器, 定义 67
 字符串, 定义 67

N

Netscape 浏览器
 安装 104
NOT BETWEEN 谓词 48
NOT EXISTS 谓词 49
NOT IN 谓词 48
NOT LIKE 谓词 49

O

OLAP 函数 62
 分区行 62
 聚合组 62
 排序行 62
OLE DB 外部表函数 66
ORDER BY 子句 22
 集合运算符 47
ORG 样本表 79

P

PDF 99
PROJECT 样本表 79

R

REAL, 数据类型 4
RIGHT OUTER 连接 57
ROLLUP 61
ROLL-UP
 小计行 61

S

SALES 样本表 81
SELECT 语句 19
SET 子句
 与 UPDATE 语句 12
SET CONSTRAINTS 语句 53
SMALLINT, 数据类型 4
SmartGuide
 向导 105
SOME 关键字 50
SQL 中的值 4
SQL Procedure 语言 v
SQL Reference v
STAFF 样本表 82
STAFFG 样本表 83

T

TIME, 数据类型 4
TIMESTAMP, 数据类型 4

U

UNION 运算符 45, 46
 排序结果 46
 涉及的用法约束 47
 数据类型 47
 说明 45
UNION ALL 45
UPDATE 语句 12
USER 专用寄存器 68

V

VALUES 子句
 与 INSERT 语句 10

VARCHAR, 数据类型 4

W

WHERE 子句 20

 分组考虑事项 30

 在 SELECT 语句中组合表数据
 (连接) 25

WITH 子句 36

WITH CHECK OPTION 14

与 IBM 联系

如果有技术问题，请在与“DB2 客户支持中心”联系之前复查并执行 *Troubleshooting Guide* 所建议的操作。本指南对您可以收集哪些信息以使“DB2 客户支持中心”更好地为您服务提出了建议。

要获取信息或订购任何“DB2 通用数据库”产品，与当地分支机构的 IBM 代表联系，或与任何特许 IBM 软件经销商联系。

您如果住在美国，请致电下列其中一个号码：

- 1-800-237-5511，可获得客户支持
- 1-888-426-4343，可了解所提供的服务项目

产品信息

您如果住在美国，请致电下列其中一个号码：

- 1-800-IBM-CALL (1-800-426-2255) 或 1-800-3IBM-OS2 (1-800-342-6672)，可订购产品或获取一般信息。
- 1-800-879-2755，可订购出版物。

<http://www.ibm.com/software/data/>

DB2 万维网网页提供关于新闻、产品说明、培训计划等等的当前 DB2 信息。

<http://www.ibm.com/software/data/db2/library/>

“DB2 产品和服务技术库”可供您访问常见问题、修订、书籍以及最新的 DB2 技术资料。

注：此资料可能只有英文版。

<http://www.elink.ibm.com/pbl/pbl/>

International Publications Ordering Web 站点提供关于如何订购书籍的信息。

<http://www.ibm.com/education/certify/>

IBM Web 站点中的“专业认证程序”提供各种 IBM 产品（包括 DB2）的认证测试信息。

<ftp.software.ibm.com>

以匿名形式注册。可在目录 /ps/products/db2 中找到有关 DB2 和许多其他产品的演示程序、修订、信息和工具。

comp.databases.ibm-db2, bit.listserv.db2-l

这些 Internet 新闻组可供用户来讨论使用 DB2 产品的经验。

On Compuserve: GO IBMDB2

输入此命令来访问 IBM DB2 系列论坛。这些论坛支持所有的 DB2 产品。

有关如何在美国以外的地区与 IBM 联系的信息，参见 **IBM Software Support Handbook** 的附录 A。要访问此文档，访问以下 Web 页面：<http://www.ibm.com/support/>，然后选择该页面底部附近的 IBM Software Support Handbook 链接。

注：在某些国家，IBM 特许经销商应与他们的经销商支持机构联系，而不是与“IBM 支持中心”联系。



Part Number: CT7YHSC

Printed in China

SB84-0223-00



CT7YHSC

