

ORACLE®

ORACLE使用手册

中 册

美国ORACLE(中国)有限公司

总 目 录

上 册

1. IBM PC / MS-DOS 下的 ORACLE 安装及用户指南
(IBM for PC / MS-DOS Installation and User's Guide)
2. IBM PC / MS-DOS 下 ORACLE 系统发行公报
(ORACLE FOR IBM PC / MS-DOS System Release Bulletin)
3. ORACLE RDBMS 发行注解
(ORACLE RDBMS Release Notes)
4. ORACLE 数据库管理员指南
(ORACLE Database Administrator's Guide)
5. ORACLE 实用程序用户指南
(ORACLE Utilities User's Guide)
6. ORACLE 错误信息及其代码手册
(Error Messages and Codes Manual)
- 附录: 常用词语英汉对照表

中 册

1. SQL 入门
(SQL Primer)
2. SQL * Plus 发行注解
(SQL * Plus Release Notes)
3. SQL * Plus 快速指南
(SQL * Plus Quick Tour)
4. SQL * Plus 用户指南
(SQL * Plus User's Guide)
5. SQL * Plus 参考指南
(SQL * Plus Reference Guide)
6. Pro * C 用户指南
(Pro * C User's Guide)
7. SQL * Calc 发行注解
(SQL * Calc Release Notes)

- 8.SQL * Calc 入门
(Introduction to SQL * Calc)
- 9.SQL * Calc 快速指南
(SQL * Calc Quick Tour)
- 10.SQL * Calc 用户指南
(SQL * Calc User's Guide)

下 册

- 1.SQL * Forms 发行注解
(SQL * Forms Release Notes)
- 2.SQL * Forms 入门
(Introduction to SQL * Forms)
- 3.SQL * Forms 快速指南
(SQL * Forms Quick Tour)
- 4.SQL * Forms 操作指南
(SQL * Forms Operator's Guide)
- 5.SQL * Forms 设计自学手册
(SQL * Forms Designer's Tutorial)
- 6.SQL * Forms 设计参考手册
(SQL * Forms Designer's Reference)
- 7.SQL * Report 发行注解
(SQL * Report Release Notes)
- 8.SQL * Report 用户指南
(SQL * Report User's Guide)

中 册 目 录

1.SQL 入门	(1)
(SQL Primer)	
2.SQL * Plus 发行注解	(46)
(SQL * Plus Release Notes)	
3.SQL * Plus 快速指南	(64)
(SQL * Plus Quick Tour)	
4.SQL * Plus 用户指南	(85)
(SQL * Plus User's Guide)	
5.SQL * Plus 参考指南	(297)
(SQL * Plus Reference Guide)	
6.Pro * C 用户指南	(393)
(Pro * C User's Guide)	
7.SQL * Calc 发行注解	(562)
(SQL * Calc Release Notes)	
8.SQL * Calc 入门	(573)
(Introduction to SQL * Calc)	
9.SQL * Calc 快速指南	(628)
(SQL * Calc Quick Tour)	
10.SQL * Calc 用户指南	(655)
(SQL * Calc User's Guide)	

SQL 入 门
(SQL Primer)
1985 年 10 月

目 录

一、	引言：样本数据库	(3)
二、	从一个表中选取数据	(5)
三、	建立一个表	(7)
四、	往表中插入新行	(9)
五、	从表中选取特定的行和列	(10)
六、	对查询结果行排序	(14)
七、	避免选取重复的行	(16)
八、	查询多个表	(17)
九、	把查询结果格式化为报表	(20)
十、	算术表达式及字符串函数	(22)
十一、	日期格式化和日期算术运算	(24)
十二、	按行组计算的函数	(26)
十三、	子查询	(28)
十四、	修改已存入的数据	(30)
十五、	动态修改数据库的描述	(32)
十六、	数据视图	(38)
十七、	数据独立性	(41)
十八、	数据共享和保密	(44)

一、 引言：样本数据库

1. 要点:

这本入门小册子解释基本的关系数据库概念，并提供一系列实例来展示 SQL 关系数据语言的能力。你将学会如何使用 SQL 来完成:

- SELECT (选择) 表中的行
- CREATE (建立) 一个表
- INSERT (插入) 新行到表中

首先，我们仔细观察一下数据库的结构并且定义一些术语。

2. 数据库和表

ORACLE 数据库由表构成。这本小册子中最常用的表是 DEPT 表和 EMP 表，这两个样本表中分别包含一个公司内的部门 (Department) 及雇员 (Employees) 信息。

样本数据库:

DEPT 表

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

EMP 表

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800.00		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1,600.00	300.00	30
7521	WARD	SALESMAN	7698	22-FEB-81	1,250.00	500.00	30
7566	JONES	MANAGER	7839	02-APR-81	2,975.00		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1,250.00	1,400.00	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2,850.00		30
7782	CLARK	MANAGER	7839	09-JUN-81	2,450.00		10
7788	SCOTT	ANALYST	7566	09-NOV-81	3,000.00		20
7839	KING	PRESIDENT		17-NOV-81	5,000.00		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1,500.00	0.00	30
7876	ADAMS	CLERK	7788	23-SEP-81	1,100.00		20
7900	JAMES	CLERK	7698	03-DEC-81	950.00		30
7902	FORD	ANALYST	7566	03-DEC-81	3,000.00		20
7934	MILLER	CLERK	7782	23-JAN-82	1,300.00		10

3. 列和行

每个表由（垂直的）列（column）及（水平的）行（row）构成。表 DEPT 有 3 列（列名分别为 DEPTNO, DNAME, 及 LOC）及 4 行（部门号 10, 20, 30, 40, 各占一行）

4. 字段和值

行由字段（field）构成。在行与列交点处的每个字段包含一个数据值（value）。

例如，在 DEPT 表的第一行中，值 10 贮存在它的 DEPTNO 字段中，值 ACCOUNTING 贮存在它的 DNAME 字段中，值 NEW YORK 贮存在它的 LOC 字段中。

（注意：除非另加说明，为了易于阅读，我们对以下三列 SAL, COMM, 及 BUDGET 在显示时总是采用 SQL 格式 999,999.99.）

二、 从一个表中选取数据

1. SELECT 命令:

由数据库中检索数据是最常用的 SQL 操作。一次数据库检索称之为一次查询 (query), 用 SELECT 命令来发出一次查询。基本的 SELECT 命令包括两个部分:

```
SELECT      某些数据 (列名)
FROM        一个或多个表 (表名)
```

每一部分 (SELECT 及 FROM) 都是一个子句 (clause)。SELECT 子句一定要首先键入, FROM 子句紧跟其后。

现在我们对 DEPT 和 EMP 表发出一些简单的 SQL 查询。首先, 取出 DEPT 表中所有的列和所有的行:

```
用户打入: SELECT DEPTNO, DNAME, LOC
           FROM DEPT;
           (其中分号; 是命令结束符)
```

则显示输出为:

DEPTNO	DNAME	LOC
-----	-----	-----
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

2. SELECT *

在上例中, 在 SELECT 子句中依次列出 DEPT 表中的每个列——DEPTNO, DNAME, LOC。通常, 为了简化起见, 你可用星号 (*) 来代表要选择表内所有的列, 而无须列出所有的列名。如下列:

```
SELECT      *
FROM        EMP;
```

则显示结果如下:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800.00		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1,600.00	300.00	30
7521	WARD	SALESMAN	7698	22-FEB-81	2,250.00	500.00	30
7566	JONES	MANAGER	7839	02-APR-81	2,975.00		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1,250.00	1,400.00	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2,850.00		30
7782	CLARK	MANAGER	7839	09-JUN-81	2,450.00		10
7788	SCOTT	ANALYST	7566	09-NOV-81	3,000.00		20
7839	KING	PRESIDENT		17-NOV-81	5,000.00		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1,500.00	0.00	30
7876	ADAMS	CLERK	7788	23-SEP-81	1,100.00		20
7900	JAMES	CLERK	7698	03-DEC-81	950.00		30
7902	FORD	ANALYST	7566	03-DEC-81	3,000.00		20
7934	MILLER	CLERK	7782	23-JAN-82	1,300.00		10

为便于阅读, 我们把一个 SQL 命令分为多行键入, 而且全使用大写字母。实际上, 你可以把整个命令全在一行打入, 而且随便使用大小写字母。但是, 命令结束符 (分号“;”)一定要在最后一行的最后一个字符位置上打入。ORACLE 一看到这个分号, 就去执行你的命令。

三、 建立一个表

1. CREATE TABLE 命令

在由你的数据库中选取数据之前，你必须把数据插入到你的数据库中，而在把数据插入到你的数据库之前，你必须建立用以存放那些数据的表。我们发出如下的 SQL 命令来建立 DEPT 表：

```
CREATE TABLE DEPT (DEPTNO NUMBER (2),  
                     DNAME CHAR (14),  
                     LOC CHAR (13));
```

显示输出：

Table created

在此 CREATE TABLE 命令中，首先你告诉 ORACLE 要建立的表名 (DEPT)。接下去，你说明表中各列的名字 (DEPTNO, DNAME, LOC) 以及每列包含的数据的类型。比如，在建立 DEPT 表时，我们指定了 DEPTNO 列只包含数值数据 (NUMBER)，DNAME 和 LOC 列为字符数据 (CHAR)——可为字母，数字或标点符号等。最后，还指定了可贮存在各列中的数值的最大长度。比如，在上述 CREATE TABLE 命令中的 LOC CHAR (13) 表示地名 (LOC) 不能长于 13 个字符。

我们用如下命令来建立 EMP 表：

```
CREATE TABLE EMP (EMPNO NUMBER (4) NOT NULL,  
                   ENAME CHAR (10),  
                   JOB CHAR (9),  
                   MGR NUMBER (4),  
                   HIREDATE DATE,  
                   SAL NUMBER (7, 2),  
                   COMM NUMBER (7, 2),  
                   DEPTNO NUMBER (2));
```

2. 不允许空值

在上述 CREATE TABLE 命令，我们定义表 EMP 中的列 EMPNO 为 NOT NULL。NOT NULL 的意思是，在 EMP 表中的每一行都必须在它的 EMPNO 字段中包含一个值 (ORACLE 将禁止你插入不包含 EMPNO 字段值的行)。换句话说，不允许对这列给缺省值 (称为空值——NULL)。

3. 自动进行数据验证

在上例中因指明 NOT NULL，ORACLE 就能在你往数据库中送入数据时对数据数值自动作一些校验，从而保证它们遵从你所指定的数据完整性断言 (integrity assertions)。在后面还会看到，如何指定在一列中的值应当是 UNIQUE (唯一的)。指定 UNIQUE 后，ORACLE 就能自动防止两行在同一列上有相同值。

4. 日期数据类型

名为 HIREDATE 的列，定义为含有日期 (DATE) 数据值。

5. 十进制数字:

工资 (SAL) 和佣金 (COMM) 列都定义为最大长度为 7 的数值，其中有 2 位在十进小数点的右侧——NUMBER (7, 2)。

6. 高级数据类型

在保持与 IBM 的 SQL/DS 及 DB2 兼容的同时，ORACLE 对 SQL 进行了扩充，以使它对用户更实用更友好。这种扩充之一就表现在高级数据类型方面。ORACLE 使你按数据实际的样子——CHARACTER, NUMBER, DATE 或 TIME 来定义数据。可是，IBM 的 SQL 要求你按数据贮存在计算机中的内部形式——INTEGER, FLOATING POINT, DECIMAL 等来说明数据。使用 ORACLE 时，你无需关心数据在内部被贮存的形式。

7. 数据压缩

ORACLE 并不浪费空间去按你为每个字段所指定的最大长度来保留空间。比如，虽然你指定表 EMP 中的字段 ENAME 最大长度为 14，但 ORACLE 为值 WARD 仅使用 4 字符存贮空间。又例如，未接收佣金的雇员的 COMM 字段取空值 (NULL)，则根本不占用存贮空间。

四、 往表中插入新行

INSERT 命令：完成建表之后，你就可以开始使用 INSERT 命令来向该表送入数据行。现在发出如下的命令以送入 DEPT 表中的第一个数据行：

```
INSERT INTO DEPT  
VALUES (30, 'SALES', 'CHICAGO');
```

显示输入出：

```
1 record created
```

在 INSERT 命令中，你指明要插入数据行的表名，然后，列出应送入每列的数据值 (30, 'SALES', 'CHICAGO')。后面我们还会看到另一种插入一行或多行的方法。

现在，我们再回到查询和 SELECT 命令。

五、 从表中选取特定的行和列

1. 选取特定的列:

在前两个查询实例中都是选取一个表的全部列和全部行。如果你并不打算查看该表的所有列，那么在你的 SELECT 子句之后就只需列出你所要求的各列的名字。

```
SELECT      DNAME, DEPTNO
FROM        DEPT;
```

DNAME	DEPTNO
ACCOUNTING	10
RESEARCH	20
SALES	30
OPERATIONS	40

2. 控制列的顺序

如同你已看到的，任何查询结果本身也是个由列和行组成的表。在你的查询子句 SELECT 中列出的各列的次序控制着你的结果表中各列从左到右的顺序。如果你使用了 SELECT *，那么你的查询结果的自左向右的顺序取决于你在 CREATE TABLE 命令中定义各列时的次序。

3. 选取特定的行

上例表明，如何用 SELECT 子句来检索表中特定的列，但如何从你的表中检索特定的行呢？为此，必须要在你的 SELECT 命令中增加一个 WHERE 子句。

```
SELECT      *
FROM        EMP
WHERE       DEPTNO = 30;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	20-FEB-81	1,600.00	300.00	30
7521	WARD	SALESMAN	7698	22-FEB-81	1,250.00	500.00	30
7654	MARTIN	SALESMAN	7698	28-SEP-81	1,250.00	1,400.00	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2,850.00		30
7844	TURNER	SALESMAN	7698	08-SEP-81	1,500.00	0.00	30
7900	JAMES	CLERK	7698	03-DEC-81	950.00		30

WHERE 子句的作用是让 ORACLE 在你的表中搜索数据并且只检索出满足你的搜索条件 (search-condition) 的各行。在上例中，ORACLE 取回的仅是其中在 30 号部门

(WHERE DEPTNO = 30) 工作的雇员。

4. 多个搜索条件

有时需要在一个 WHERE 子句中指定一个以上的搜索条件。例如，你打算列出所有的，其收入高于 \$ 2,800 的经理。

```
SELECT      ENAME, JOB, SAL
FROM        EMP
WHERE       JOB = 'MANAGER'
AND         SAL > 2800;
```

ENAME	JOB	SAL
JONES	MANAGER	2,975.00
BLAKS	MANAGER	2,850.00

多个搜索条件要用 AND 来联系 (JOB = 'MANAGER' AND SAL > 2800)。AND 的含义是，在 ORACLE 取出一行之前，你的数据必须满足所有的搜索条件。在一个 WHERE 子句中，你可以 AND (联合) 任意多个搜索条件。

5. 二者取一的搜索条件

除了能选取同时满足所有的搜索条件的行外，你也可选取满足多个搜索条件中任何一个的行。

```
SELECT      ENAME, JOB, SAL
FROM        EMP
WHERE       JOB = 'MANAGER'
OR          SAL > 2800;
```

ENAME	JOB	SAL
JONES	MANAGER	2,975.00
BLAKE	MANAGER	2,850.00
CLARK	MANAGER	2,450.00
SCOTT	ANALYST	3,000.00
KING	PRESIDENT	5,000.00
FORD	ANALYST	3,000.00

6. 否定的搜索条件

你也可选取不满足某些搜索条件的行。例如，取出所有不在 30 号部门工作的经理。

```

SELECT      ENAME, JOB, DEPTNO
FROM        EMP
WHERE       JOB='MANAGER'
AND         DEPTNO! = 30;

```

ENAME	JOB	DEPTNO
JONES	MANAGER	20
CLARK	MANAGER	10

其中! = 表示不等于。

如有必要，你可在同一查询中用 AND, OR 和 NOT 连接符来结合多个搜索条件以取出你所要求的行。

7. 按一定范围搜索

算符 BETWEEN 可使你选取出，某些字段取值在你所指定范围内的各行。比如，选取工资在 1200 与 1400 之间所有雇员。

```

SELECT      ENAME, SAL
FROM        EMP
WHERE       SAL BETWEEN 1200 AND 1400;

```

ENAME	SAL
WARD	1,250.00
MARTIN	1,250.00
MILLER	1,300.00

8. 按一组（离散）值来搜索

算符 IN 可使你选取出，某些字段取值为你所指定的一组（离散）值之一的各行。比如，选取部门号为 10 或 30 的各部门。

```

SELECT      *
FROM        DEPT
WHERE       DEPTNO IN (10, 30);

```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
30	SALE	CHICAGO

注意：要把指定的一组值括在括号内，如 (10, 30)。

这种查询也可用 OR 连接符方便地完成，即 WHERE DEPTNO=10 OR DEPTNO=30，其查询结果是相同的。

9. 匹配一个字符图式 (pattern)

你可以选取与一个字符或数字图式匹配的各行。比如，选取其名字的第 3 个字母为 R 的全体雇员。(LIKE 算符)

```
SELECT      ENAME
FROM        EMP
WHERE       ENAME LIKE '___R%';
```

```
ENAME
-----
WARD
MARTIN
TURNER
FORD
```

在这个例子中，我们使用 SQL 算符 LIKE，指示 ORACLE 从 EMP 表中取出这样的行，它们的 ENAME 字段取值与我们指定的一个图式 ('___R%') 匹配（或相似—LIKE）。

下画线 (___) 表示一个字符位置，百分号 (%) 表示为零个或多个字符的串。

在比较算符 BETWEEN, IN 和 LIKE 之前都可前缀以单词 NOT 还可与连接算符 AND 或 OR 一起拼成一个复杂的 WHERE 子句，以能检索出你所需要的行。