**第10章 使用数据库**

**1. SQL是什么？**

**常见的关系型数据库管理系统：MySQL，Oracle、DB2、PostgreSQL、Sybase、Microsoft SQL Server等。**

**MySQL与标准的SQL之间，存在细微的差别。**

**sql组成：**

**DDL：数据库模式定义语言，关键字：create**

**DML：数据操纵语言，关键字：insert、delete、update**

**DCL：数据库控制语言 ，关键字：grant、remove**

**DQL：数据库查询语言，关键字：select**

**2. 在数据库中插入数据**

**Book-O-Rama书店的数据库模式：**

CUSTOMERS(**CustomerID**,Name,Address,City)

ORDERS(**OrderID**,***CustomerID***,Amount,Date)

BOOKS(**ISBN**,Author,Title，Price）

ORDER\_ITEMS(**OrderID**，**ISBN**,Quantity）

BOOK\_REVIEWS(**ISBN**,Review）

**在Book-O-Rama的customers表中插入一条记录：**



**数字和日期，不需要引号。字符串，应该包含在单引号或双引号中。**

**给部分列插入数据：**



**也可以这样插入：**



**book\_insert.sql——操作Book-O-Rama数据库表的SQL脚本**

**————————————————————————————————**

use books;

insert into customers values

(3, "Julie Smith", "25 Oak Street", "Airport West"),

(4, "Alan Wong", "1/47 Haines Avenue", "Box Hill"),

(5, "Michelle Arthur", "357 North Road", "Yarraville");

insert into orders values

(NULL, 3, 69.98, "2007-04-02"),

(NULL, 1, 49.99, "2007-04-15"),

(NULL, 2, 74.98, "2007-04-19"),

(NULL, 3, 24.99, "2007-05-01");

insert into books values

("0-672-31697-8", "Michael Morgan", "Java 2 for Professional Developers", 34.99),

("0-672-31745-1", "Thomas Down", "Installing Debian GNU/Linux", 24.99),

("0-672-31509-2", "Pruitt, et al.", "Teach Yourself GIMP in 24 Hours", 24.99),

("0-672-31769-9", "Thomas Schenk", "Caldera OpenLinux System Administration Unleashed", 49.99);

insert into order\_items values

(1, "0-672-31697-8", 2),

(2, "0-672-31769-9", 1),

(3, "0-672-31769-9", 1),

(3, "0-672-31509-2", 1),

(4, "0-672-31745-1", 3);

insert into book\_reviews values

("0-672-31697-8", "Morgan's book is clearly written and goes well beyond most of the basic Java books out there.");

**————————————————————————————————>mysql -h host -u root -p books < book\_insert.sql**

**通过命令运行该脚本。**

**3. 从数据库中获取数据**





**3.1 获取满足特定条件的数据**



**WHERE子句可包括各种条件运算符**

**1 ,比较运算符(大小比较)**

**>、>=、=、<、<=、<>或!= 、IS NOT NULL 、IS NULL**

**2,范围运算符(表达式值是否在指定的范围内**

**between、 IN 、 NOT IN 、 LIKE、 NOT LIKE 、 REGEXP（正则表达式）**



**3.2 从多个表中获取数据**













**要查询Julie的订单，得先在customers表中查找 name属性值为 Julie的customerid，然后从orders表中查该customerid所对应的订单。**

**关联，是SQL中最复杂的一部分。**

**A. 简单的双表关联**

**select orders.orderid,orders.amount,orders.date**

**from customers,orders**

**where customers.name = 'Julie Smith'**

**and customers.customerid = orders.customerid;**



**表名称之间的逗号，等价于 inner join 或 cross join 称为 完全关联。**

**customers.customerid = orders.customerid; 是关联条件（等价关联）。**

**B. 关联多个表**

**要查出 哪些顾客已经订购了关于Java的图书**

**select customers.name**

**from customers, orders, order\_items, books**

**where customers.customerid = orders.customerid**

**and orders.orderid = order\_items.orderid**

**and order\_items.isbn = books.isbn**

**and books.title like '%Java%' ;**



**C. 查找不匹配行**

**例如，查询: 从来没有订单的顾客，或从没有被订购过一次的图书。**

**左连接：如果右表中没有匹配行，结果中就会增加一行，该行右边的列值为NULL. 例：**

**select customers.customerid, customers.name, orders.orderid**

**from customers left join orders**

**on customers.customerid = orders.customerid;**



**如果我们只需要查看没有订购任何商品的顾客：**

**select customers.customerid, customers.name, orders.orderid**

**from customers left join orders**

**using(customerid)**

**where orders.orderid is null;**



**using语法，不需要指定连接属性所来自的表，但是，两个表中的列，必须名称相同。**

**D. 使用表的别名： Aliases**

B.中的例子

select customers.name

from customers, orders, order\_items, books

where customers.customerid = orders.customerid

and orders.orderid = order\_items.orderid

and order\_items.isbn = books.isbn

and books.title like '%Java%' ;

**改写为：**

**select c.name**

**from customers as c, orders as o, order\_items as oi , books as b**

**where c.customerid = o.customerid**

**and o.orderid = oi.orderid**

**and oi.isbn = b.isbn**

**and b.title like '%Java%' ;**



**这种用法，如果要查找同一个表中值相同的行，它就很有意义。例如，要查找住在同一个城市的顾客，可以给同一个表起两个不同的别名。**

**select c1.name, c2.name, c1.city**

**from customers as c1, customers as c2**

**where c1.city = c2.city**

**and c1.name != c2.name ;**



**E. 关联的总结**

**笛卡尔乘积、完全关联、交叉关联、内部关联、等价关联、左关联**

**3.3 以特定的顺序获取数据**

**select name, address**

**from customers**

**order by name;**

**默认的顺序是升序，等价于**

**select name, address**

**from customers**

**order by name asc;**



**降序：**

**select name, address**

**from customers**

**order by name desc;**



**3.4 分组与统计**



**count() 统计记录条数，如 select count(\*) from stu;**

**sum() 统计记录字段的和，如select sum(salary) from emp;**

**avg() 统计记录字段的平均值，如select avg(salary) from emp;**

**max() 查询字段中的最大值，如select max(age) from stu;**

**min() 查询字段中的最小值，如select min(age) from stu;**

**例子：计算订单总金额的平均值：**

**select avg(amount) from orders;**

**按顾客数分组浏览，从而知道哪些顾客的订单总金额最大：**

**select customerid,avg(amount)**

**from orders**

**group by customerid;**

**想知道哪些顾客的平均订单总金额超过$50，可以如下：**

**select customerid,avg(amount)**

**from orders**

**group by customerid**

**having avg(amount) >50 ;**

**3.5 选择要返回的行**

**limit 起始行号，返回行数 （注意：行号是以0开始索引的。）**

**select name**

**from customers**

**limit 2,3;**

**3.6 使用子查询**

**子查询，是一个嵌套在另一个查询内部的查询。**

**1. 基本的子查询**

**用一个查询的结果作为另一个查询的比较条件。**

**例如：希望找到一个金额最大的订单**

**select customerid ,amount**

**from orders**

**where amount = (select max(amount) from orders);**

**子查询返回一个单一值，最大金额，然后用作输出查询的比较条件。**



**select customerid ,amount**

**from orders**

**order by amount desc**

**limit 1;**

**也能得到相同结果。**

**2. 子查询和操作符**



**3. 关联子查询**

**在内部查询中，使用外部查询的结果**

**select isbn,title**

**from books**

**where not exists**

**(select \* from order\_items where order\_items.isbn = books.isbn);**

**检索任何还没有被订购的图书。**

**5. 使用子查询作为临时表**

**select \* from**

**(select customerid,name from customers where city = 'Box Hill')**

**as box\_hill;**

**必须为子查询的结果定义一个别名。**



**4. 更新数据库记录**

**update books**

**set price = price\*1.1 ;**

**update customers**

**set address = '250 Road'**

**where customerid = 4;**

**5. 创建表后 修改表**

**alter table customers**

**modify name char(70) not null;**

**alter table orders**

**add tax float(6,2) after amount ;**

**alter table orders**

**drop tax;**

**6. 删除数据库中的记录**

**delete from table;**

**所有行都被删除，因此要非常小心！**

**delete from customers**

**where customerid = 5;**

**7. 表的删除**

**drop table *table*;**

**8. 删除整个数据库**

**drop database *database*;**