第四章：检索数据

1. 检索单个列：select prod\_name from products;
2. 检索多个列：select prod\_name,prod\_id from products;
3. 检索所有列：select \* from products;
4. 只显示不同的列：select distinct prod\_id from products;
5. 显示前5行的列：select prod\_name from products limit 5;
6. 显示第3行开始的4行的列：select prod\_name from products limit 3,4;

第五章：检索排序数据

1. 排序子句：select prod\_name from products order by prod\_name;
2. 多个列排序：select prod\_id,prod\_name,prod\_price from products order by prod\_id,prod\_name;
3. 默认为升序排序，降序排序实现：select prod\_id,prod\_price from products order by prod\_price desc,prod\_id;(desc只对前面一列生效)；
4. 选出最高价格：select prod\_price from products order by prod\_price limit 1;

第六章：过滤数据

1. 单值元素过滤 select prod\_name from products where prod\_name=’Safe’;
2. 不等过滤 select prod\_price from products where prod\_price != 2.50;
3. 大于小于过滤：select prod\_price from products where prod\_price < 10;
4. 范围过滤: select prod\_price from products where prod\_price between 5 and 10;
5. 空值过滤：select cust\_id from customers where cust\_email is NULL;

第七章：过滤数据

1. And操作符:select prod\_id,prod\_name,prod\_price from products where vend\_id=1003 and prod\_price<=10;
2. OR操作符: select prod\_id,prod\_name,prod\_price from products where vend\_id=1003 or vend\_id=1002;
3. 组合过滤: select prod\_id,prod\_name,prod\_price from products where prod\_price<=10 and (vend\_id=1003 or vend\_id=1002);
4. IN操作符: select prod\_id,prod\_name,prod\_price from products where vend\_id in (1002,1003);
5. NOT操作符: select prod\_id,prod\_name,prod\_price from products where vend\_id not in (1002,1003);

第八章：用通配符进行过滤

1. %匹配任意多个字符，查找含有名称anvil的产品: select prod\_id,prod\_name,prod\_price from products where prod\_name like ‘%anvil%’;
2. \_匹配单个字符：select prod\_id,prod\_name,prod\_price from products where prod\_name like ‘\_ ton anvil’;

第九章：用正则表达式搜索

1. Regexp代替like，不用通配符即可查找: select prod\_id,prod\_name,prod\_price from products where prod\_name regexp ‘1000’;
2. “.”匹配一个字符: select prod\_id,prod\_name,prod\_price from products where prod\_name regexp ‘.000’;
3. |检索多个串: select prod\_id,prod\_name,prod\_price from products where prod\_name regexp ‘1000|2000’;
4. []匹配多个字符: select prod\_id,prod\_name,prod\_price from products where prod\_name regexp ‘[123] ton’;
5. [^]显示不匹配的行: select prod\_id,prod\_name,prod\_price from products where prod\_name regexp ‘[^123] ton’;
6. 范围匹配: select prod\_id,prod\_name,prod\_price from products where prod\_name regexp ‘[1-5] ton’;
7. 匹配特殊字符使用\\：select prod\_id,prod\_name,prod\_price from products where prod\_name regexp ‘\\.’;

第十章：创建计算字段

1. 计算字段并不存在于数据表中，而是通过select语句查询时创建的。
2. 拼接字段使用Concat()函数:select Concat(vend\_name,’(‘,vend\_country,’)’) from vendors;
3. 创建别名使用AS: select Concat(vend\_name,’(‘,vend\_country,’)’) as vend\_title from vendors;
4. 进行计算: select prod\_id,quantity,item\_price,quantity\*item\_price as total\_price from orderitems where order\_num=20005;

第十一章：使用数据处理函数

1. 字符串处理函数:select vend\_name,Upper(vend\_name) as upper\_name from vendors;
2. 时间处理函数: select cust\_id,order\_num from orders where Date(order\_date)='2005-09-01';
3. 数值处理函数:

第十二章：汇总数据

1. 聚集函数：avg(),count(),max(),min(),sum()。
2. Avg函数：select avg(prod\_price) avg\_price from products where vend\_id=1003;
3. Count函数统计所有行: select count(\*) as cust\_num from customers;
4. Count函数统计指定列满足条件的行: select count(cust\_email) as num\_cust from customers where cust\_email like '%was%';
5. Max/min返回指定列的最大值/最小值: select max(prod\_price) as max\_price from products;
6. Sum计算某列数据和: select sum(quantity) as total\_quantity from orderitems where order\_num=20005;

第十三章:分组数据

1. Group by创建分组:select vend\_id count(\*) from products group by vend\_id;
2. Having子句过滤分组: select vend\_id,count(\*) from products group by vend\_id having count(\*)>2;
3. Where和having结合: select vend\_id,count(\*) from products group by vend\_id having count(\*)>=2;
4. 子句顺序:select->from->where->group by->having->order by->limit;

第十四章：使用子查询

1. 嵌套子查询: select cust\_id from orders where order\_num in( select order\_num from orderitems where prod\_id='TNT2');

第十五章：联结表

1. 等值联结:select vend\_name,prod\_name,prod\_price from vendors,products where vendors.vend\_id=products.vend\_id;
2. 内联结: select vend\_name,prod\_name,prod\_price from vendors inner join products on vendors.vend\_id=products.vend\_id;

第十六章：创建高级联结

1. 通过给表定义别名实现自联结: select p1.prod\_id,p1.prod\_name from products as p1,products as p2 where p1.vend\_id=p2.vend\_id and p2 .prod\_id='dtntr';
2. 外联结分为左外联结和右外联结: select customers.cust\_id,orders.order\_num from customers left outer join orders on customers.cust\_id=orders.cust\_id;

第十七章:组合查询

1. 组合查询将查询相同列的多个查询结果组合。
2. Union实现组合查询: select vend\_id,prod\_id,prod\_price from products where prod\_price<=5 union select vend\_id,prod\_id,prod\_price from products where vend\_id in(1001,1002);
3. 要排序时只需要在最后一个select中添加order by子句;

第十八章:全文本搜索

第十九章:插入数据

1. Insert into插入一行: insert into orders values('20010','2020-8-28 00:00:00','10001');
2. 指定列名插入: insert into orders(cust\_id,order\_num,order\_date) values('10002','20011','2020-8-28 00:00:00');
3. 插入多行：直接多条inert into语句

第二十章：更新和删除数据

1. Update更新数据: update orders set cust\_id=10003 where order\_num=20011;
2. Delete 删除数据: delete from orders where order\_num in (20010,20011);

第二十一章：创建和操纵表

1. 创建表:Create table orders2(order\_num int not null default 1,order\_date data not null,cust\_id int not null,primary key(order\_num))engine=innodb;
2. 更新表:alter table orders2 add order\_txt char(20);
3. 删除表:drop table orders2;

第二十二章：使用视图

1. 创建视图: create view productcustomers as select cust\_name,cust\_contact,prod\_id from customers,orders,orderitems where customers.cust\_id=orders.cust\_id and orderitems.order\_num=orders.order\_num;
2. 使用视图: select cust\_name,cust\_contact from productcustomers where prod\_id='tnt2';
3. 删除视图: drop view productcustomers;