1、基础查询

- 查所有字段
  - select * from customers;
- 只查指定字段
  - select cust_id, cust_name from customers;
- 去重
  - select distinct cust_id from orders;
  - select distinct cust_id,order_num from orders; # 表示对组合进行去重，两个字段合起来重复的才去重

- 实例
  - 查询所有的用户信息
  - 查询所有的产品信息
  - 查询所有的订单信息
  - 查询下过订单的客户编号？(去哪张表查？是否需要去重？)

```
mysql> select * from orders;
+-----------+---------------------+---------+
| order_num | order_date          | cust_id |
+-----------+---------------------+---------+
|     20005 | 2005-09-01 00:00:00 |   10001 |
|     20006 | 2005-09-12 00:00:00 |   10003 |
|     20007 | 2005-09-30 00:00:00 |   10004 |
|     20008 | 2005-10-03 00:00:00 |   10005 |
|     20009 | 2005-10-08 00:00:00 |   10001 |
|     20010 | 2005-10-08 05:00:00 |   10001 |
|     20011 | 2005-10-08 20:00:00 |   10001 |
+-----------+---------------------+---------+
7 rows in set (0.00 sec)

mysql> select cust_id,order_num from orders;
+---------+-----------+
| cust_id | order_num |
+---------+-----------+
|   10001 |     20005 |
|   10001 |     20009 |
|   10001 |     20010 |
|   10001 |     20011 |
|   10003 |     20006 |
|   10004 |     20007 |
|   10005 |     20008 |
+---------+-----------+
7 rows in set (0.00 sec)

mysql> select distinct cust_id from orders;
+---------+
| cust_id |
+---------+
|   10001 |
|   10003 |
|   10004 |
|   10005 |
+---------+
4 rows in set (0.00 sec)

mysql> select distinct cust_id,order_num from orders;
+---------+-----------+
| cust_id | order_num |
+---------+-----------+
|   10001 |     20005 |
|   10001 |     20009 |
|   10001 |     20010 |
|   10001 |     20011 |
|   10003 |     20006 |
|   10004 |     20007 |
|   10005 |     20008 |
+---------+-----------+
7 rows in set (0.00 sec)

mysql>
```

2、指定返回行数

- limit
  - 用法1:
    - limit m,n 表示从第m行开始，取n行 【备注，mysql是从0开始的】
      - limit后面接2个参数的时候，第一个参数为**偏移量**，第二个参数为**返回的行数**
      - 意思是从行号为m的行后面，取n行
      - mysql的行号是从0开始计算的
      - limit 2,3表示第3行开始往后数3行，也就是第3,4,5行
  - 用法2:
    - limit n 表示返回的行数，就表示返回前几行的意思
      - 从第1行开始计算，可以理解成此时的偏移量是0，比如limit 3 表示返回前3行
    - select * from customers limit 3;
  - 常用于分页操作

```
mysql> select * from orders limit 1,3;
+-----------+---------------------+---------+
| order_num | order_date          | cust_id |
+-----------+---------------------+---------+
|     20006 | 2005-09-12 00:00:00 |   10003 |
|     20007 | 2005-09-30 00:00:00 |   10004 |
|     20008 | 2005-10-03 00:00:00 |   10005 |
+-----------+---------------------+---------+
3 rows in set (0.00 sec)

mysql> select * from orders limit 3;
+-----------+---------------------+---------+
| order_num | order_date          | cust_id |
+-----------+---------------------+---------+
|     20005 | 2005-09-01 00:00:00 |   10001 |
|     20006 | 2005-09-12 00:00:00 |   10003 |
|     20007 | 2005-09-30 00:00:00 |   10004 |
+-----------+---------------------+---------+
3 rows in set (0.00 sec)
```

3、排序

- order by 排序字段 desc，降序
  - select prod_name, prod_price from products order by prod_price desc;
- order by 排序字段 [asc]，默认升序
  - select prod_name, prod_price from products order by prod_price;
  - select prod_name, prod_price from products order by prod_price asc;
- 还可以多列排序
  - order by 排序字段1 [desc|asc],排序字段2[desc|asc],... ; # 先按照
- 实例
  - 查询产品名称和价格，以价格降序排列?
  - 查询产品名称和价格，随机排序返回?
    - select prod_name, prod_price from products order by rand();
    - select prod_name, prod_price from products order by rand() limit 1; # 随机查一个
  - 通过排序结合limit找出产品的最高价?
  - 通过排序结合limit找出产品的最低价?
  - 通过排序找出最低价格产品的名称? x
    - 这个是不行的啊，因为可能有产品价格都最低，价格一样的，如何确定是1行，还是2行还是多行呢
    - 这个通过limit现在搞不定，后续处理

```
mysql> select prod_name, prod_price from products order by prod_price desc;
+---------------+------------+
| prod_name     | prod_price |
+---------------+------------+
| JetPack 2000  |      55.00 |
| Safe          |      50.00 |
| JetPack 1000  |      35.00 |
| 2 ton anvil   |      14.99 |
| Detonator     |      13.00 |
| TNT (5 sticks) |      10.00 |
| Bird seed     |      10.00 |
| 1 ton anvil   |       9.99 |
| Oil can       |       8.99 |
| .5 ton anvil  |       5.99 |
| Sling         |       4.49 |
| Fuses         |       3.42 |
| Carrots       |       2.50 |
| TNT (1 stick) |       2.50 |
+---------------+------------+
14 rows in set (0.00 sec)

mysql> select prod_name, prod_price from products order by prod_price;
+---------------+------------+
| prod_name     | prod_price |
+---------------+------------+
| TNT (1 stick) |       2.50 |
| Carrots       |       2.50 |
| Fuses         |       3.42 |
| Sling         |       4.49 |
| .5 ton anvil  |       5.99 |
| Oil can       |       8.99 |
| 1 ton anvil   |       9.99 |
| TNT (5 sticks) |      10.00 |
| Bird seed     |      10.00 |
| Detonator     |      13.00 |
| 2 ton anvil   |      14.99 |
| JetPack 1000  |      35.00 |
| Safe          |      50.00 |
| JetPack 2000  |      55.00 |
+---------------+------------+
14 rows in set (0.00 sec)

mysql> select prod_name, prod_price from products order by prod_price asc;
+---------------+------------+
| prod_name     | prod_price |
+---------------+------------+
| TNT (1 stick) |       2.50 |
| Carrots       |       2.50 |
| Fuses         |       3.42 |
| Sling         |       4.49 |
| .5 ton anvil  |       5.99 |
| Oil can       |       8.99 |
| 1 ton anvil   |       9.99 |
| TNT (5 sticks) |      10.00 |
| Bird seed     |      10.00 |
| Detonator     |      13.00 |
| 2 ton anvil   |      14.99 |
| JetPack 1000  |      35.00 |
| Safe          |      50.00 |
| JetPack 2000  |      55.00 |
+---------------+------------+
14 rows in set (0.00 sec)

mysql> select prod_name, prod_price from products order by prod_price asc limit 1;
+-----------+------------+
| prod_name | prod_price |
+-----------+------------+
| Carrots   |       2.50 |
+-----------+------------+
1 row in set (0.00 sec)

mysql> select prod_name, prod_price from products order by prod_price desc limit 1;
+-------------+------------+
| prod_name   | prod_price |
+-------------+------------+
| JetPack 2000 |      55.00 |
+-------------+------------+
1 row in set (0.00 sec)
```

4、查询条件过滤

- 语法:
  - select * from 表名 where 过滤 条件
- where 子句支持的操作符
  - =
  - <>
  - !=
  - >
  - <
  - >=
  - <=
  - between ... and ...
  - and
  - or
  - in
  - not
  - is
  - like, % 代表任意字符: _ 代表一个字符

5、别名

- 可以对表、字段、计算的结果(select整个语句)起别名
