

总体评价



王怡彬 未通关
学号: U201914858
截止前完成关卡: 15/19

分班: 计算机2020级卓越班
最新完成关卡: 15/19

完成效率: --

课堂最高完成效率: --

通关时间	计时规则 ?	实训总耗时 ?	评测次数	迟交扣分	最终成绩	总评
--	页面停留时长	5小时 38分钟 30秒	102	--	27.0/38.0	良好

阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	金融应用场景介绍,查询客户主要信息	2022-10-27 20:12	3	3	2022-10-27 20:16	3分钟 35秒	否	100/100	1.00/1	1.00
2	邮箱为null的客户	2022-10-27 20:17	3	2	2022-10-27 20:19	1分钟 59秒	否	100/100	1.00/1	1.00
3	既买了保险又买了基金的客户	2022-10-27 20:19	10	4	2022-10-27 20:26	6分钟 41秒	否	300/300	2.00/2	2.00
4	办理了储蓄卡的客户信息	2022-10-27 20:26	4	2	2022-10-27 20:29	2分钟 35秒	否	300/300	1.00/1	1.00
5	每份金额在30000 ~ 50000之间的理财产品	2022-10-27 20:29	4	1	2022-10-27 20:37	1分钟 37秒	否	200/200	1.00/1	1.00

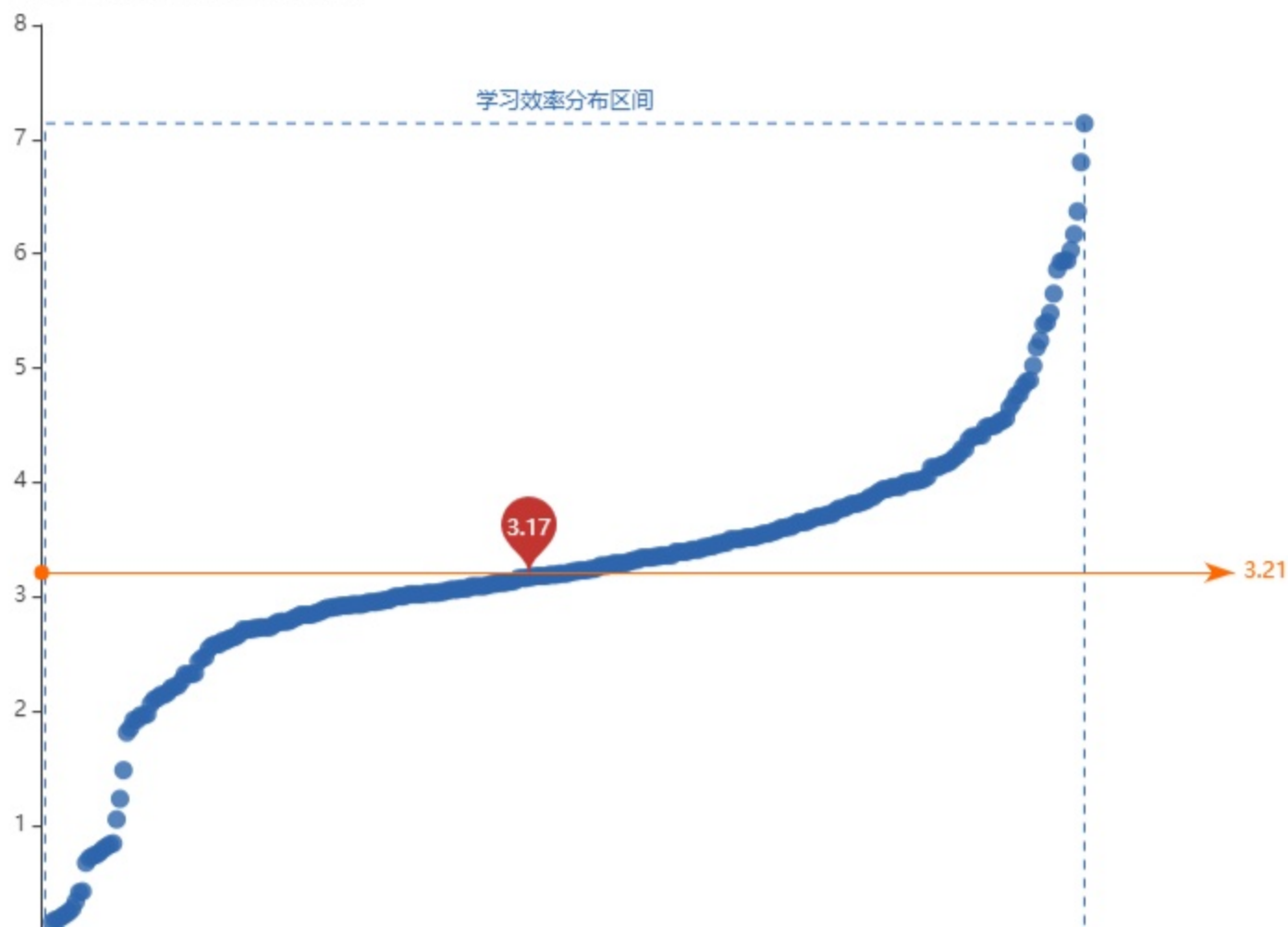
6	商品收益的众数	2022-10-27 20:37	6	8	2022-10-27 21:04	26分钟 58秒	否	700/700	2.00/2	2.00
7	未购买任何理财产品的武汉居民	2022-10-27 21:15	7	2	2022-10-27 21:33	18分钟 43秒	否	300/300	1.00/1	1.00
8	持有两张信用卡的用户	2022-10-27 21:34	11	5	2022-10-27 21:44	10分钟 24秒	否	300/300	1.00/1	1.00
9	购买了货币型基金的客户信息	2022-10-27 22:16	7	2	2022-10-27 22:30	14分钟 31秒	否	500/500	2.00/2	2.00
10	投资总收益前三名的客户	2022-10-27 22:31	10	9	2022-11-03 19:02	29分钟 31秒	否	700/700	2.00/2	2.00
11	黄姓客户持卡数量	2022-11-03 19:03	5	6	2022-11-03 19:13	10分钟 5秒	否	500/500	2.00/2	2.00
12	客户理财、保险与基金投资总额	2022-11-03 19:13	27	45	2022-11-03 21:21	2小时 7分钟 50秒	否	900/900	3.00/3	3.00
13	客户总资产	2022-11-03 21:21	61	10	2022-11-03 21:49	28分钟 5秒	否	1000/1000	3.00/3	3.00
14	第N高问题	2022-11-03 21:49	8	2	2022-11-03 21:51	2分钟 24秒	否	800/800	2.00/2	2.00
15	基金收益两种方式排名	2022-11-03 21:52	16	1	2022-11-24 20:52	53分钟 32秒	否	1000/1000	3.00/3	3.00
16	持有完全相同基金组合的客户	2022-11-24 20:52	--	0	--	--	否	0/1000	0.00/3	0.00

17	购买基金的高峰期	--	--	0	--	--	否	0/1000	0.00/3	0.00
18	至少有一张信用卡余额超过5000元的客户信用卡总余额	--	--	0	--	--	否	0/500	0.00/2	0.00
19	以日历表格式显示每日基金购买总金额	--	--	0	--	--	否	0/900	0.00/3	0.00

图形统计

学习效率

学习效率 = $\log(\text{实训总得分} / \text{实训总耗时})$



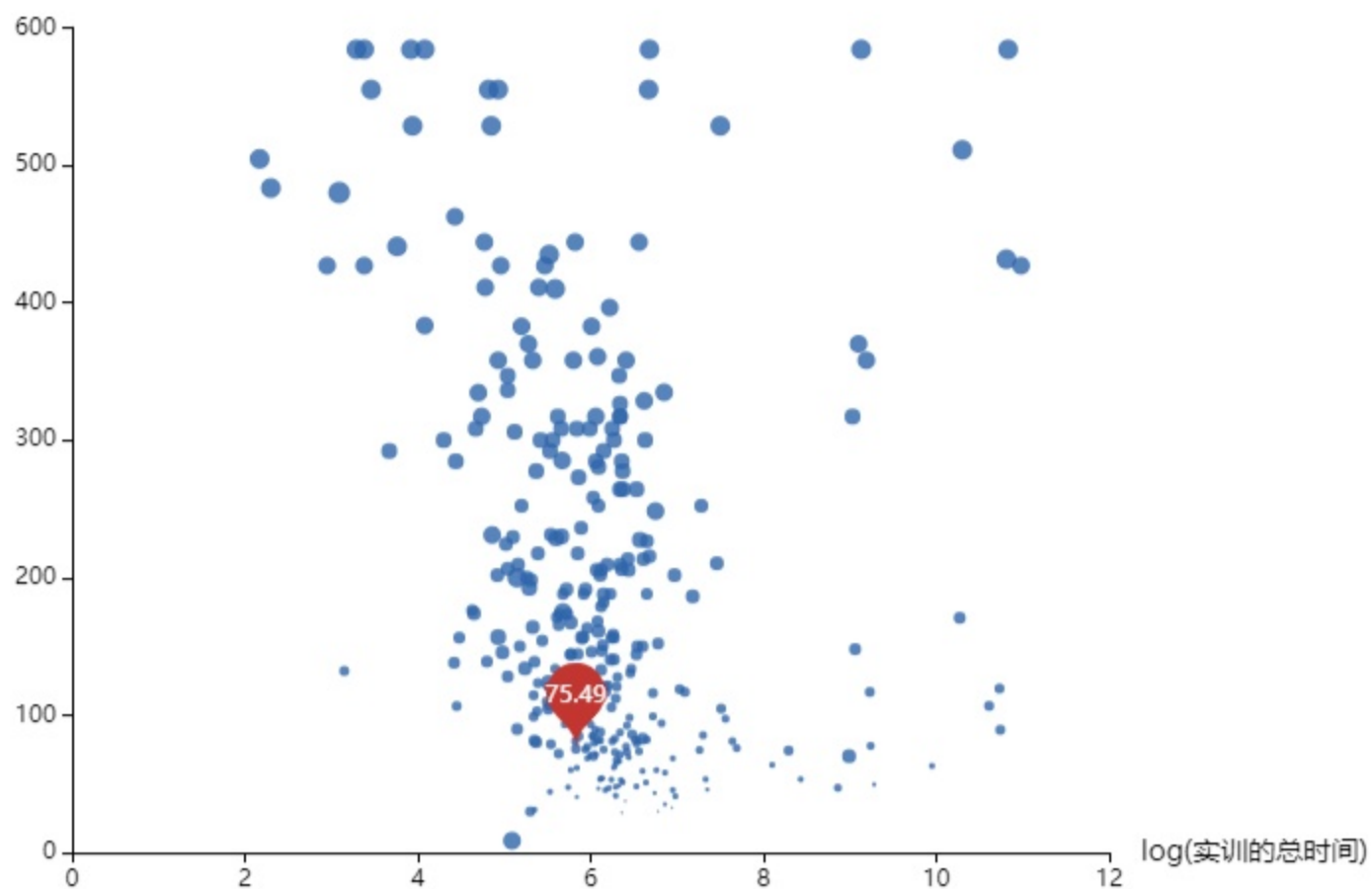
姓名	王怡彬
学号	U201914858
学习效率	3.17
当前排名	163

1个小圆点代表1个学生的学习效率
 红色水滴表示当前学生的学习效率值
 小圆点越高，学习效率越高
 橙色横向箭头：中位值，表示整个课堂学生表现的平均值

能力值

能力值 (实训获得经验值/实训评测次数)

姓名	王怡彬
学号	U201914858
能力	75.49



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大, 评测次数越少

实训详情



第1关 金融应用场景介绍,查询客户主要信息

最后通关代码 通过了本题, 在作业时间内最后一次提交的通关代码

代码修改行数 3

src/test1/test1.sql

```
1  -- 1) 查询所有客户的名称、手机号和邮箱信息。查询结果按照客户编号排序。
2  -- 请用一条SQL语句实现该查询:
3  SELECT c_name,c_phone,c_mail
4  FROM client
5  ORDER BY c_id;
6
7
8
9  /* end of your code */
10
11
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.37	51
2	共有2组测试集, 其中有2组测试结果不匹配	0.4	51
3	评测通过	0.73	52.56



第2关 邮箱为null的客户

最后通关代码 通过了本题, 在作业时间内最后一次提交的通关代码

代码修改行数 3

src/test2/test2.sql

```
1  -- 2) 查询客户表(client)中邮箱信息为null的客户的编号、名称、身份证号、手机号。
2  -- 请用一条SQL语句实现该查询:
3
4  select c_id,c_name,c_id_card,c_phone
5  from client
```

```
6 where c_mail is null;  
7  
8 /* end of your code */
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.75	52.56
2	共有2组测试集，其中有2组测试结果不匹配	0.85	52.56



第3关 既买了保险又买了基金的客户

最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 10

src/test3/test3.sql

```
1  -- 3) 查询既买了保险又买了基金的客户的名称、邮箱和电话。结果依c_id排序  
2  -- 请用一条SQL语句实现该查询：  
3  SELECT c_name,c_mail,c_phone  
4  FROM client  
5  WHERE c_id IN  
6  (SELECT pro_c_id  
7   FROM property  
8   WHERE pro_type = 2 AND pro_c_id IN  
9   (SELECT pro_c_id  
10  FROM property  
11  WHERE pro_type = 3))  
12 ORDER BY c_id;  
13  
14  
15  
16
```

1	评测通过	0.74	52.58
2	共有2组测试集，其中有2组测试结果不匹配	0.73	52.58
3	共有2组测试集，其中有2组测试结果不匹配	0.75	52.58
4	评测通过	0.74	52.58



第4关 办理了储蓄卡的客户信息

最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 4

src/test4/test4.sql

```
1  -- 4) 查询办理了储蓄卡的客户名称、手机号、银行卡号。 查询结果结果依客户编号排序。
2  -- 请用一条SQL语句实现该查询：
3  SELECT c_name,c_phone,b_number
4  FROM client,bank_card
5  WHERE c_id = b_c_id AND b_type = '储蓄卡'
6  ORDER BY c_id;
7
8
9
10
11 /* end of your code */
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.73	52.58
2	共有2组测试集，其中有2组测试结果不匹配	0.74	52.58



第5关 每份金额在30000 ~ 50000之间的理财产品

最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 4

src/test5/test5.sql

```
1  -- 5) 查询理财产品中每份金额在30000~50000之间的理财产品的编号,每份金额,理财年限,并按照金额升序排序,金额相同的按照理财年限降序排序。
2  -- 请用一条SQL语句实现该查询:
3  SELECT p_id,p_amount,p_year
4  FROM finances_product
5  WHERE p_amount BETWEEN 30000 AND 50000
6  ORDER BY p_amount, p_year DESC;
7
8  /* end of your code */
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.75	52.58



最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 6

src/test6/test6.sql

```
1  -- 6) 查询资产表中所有资产记录里商品收益的众数和它出现的次数。
2  -- 请用一条SQL语句实现该查询：
3
4  select pro_income,count(pro_income) as presence
5  from property
6  group by pro_income
7  having count(pro_income) >= all(select count(pro_income)
8  |   |   |   |   |   |   |   |   from property
9  |   |   |   |   |   |   |   |   group by pro_income);
10
11
12
13  /* end of your code */
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.56	51.72
2	共有2组测试集，其中有2组测试结果不匹配	0.56	51.72
3	共有2组测试集，其中有2组测试结果不匹配	0.57	51.72
4	评测通过	0.57	51.72
5	共有2组测试集，其中有2组测试结果不匹配	0.63	51.48
6	共有2组测试集，其中有2组测试结果不匹配	0.61	51.48
7	共有2组测试集，其中有2组测试结果不匹配	0.63	51.48



第7关 未购买任何理财产品的武汉居民

最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 7

src/test7/test7.sql

```
1  -- 7) 查询身份证隶属武汉市没有买过任何理财产品的客户的名称、电话号、邮箱。
2  --  请用一条SQL语句实现该查询：
3
4  select c_name, c_phone, c_mail
5  from client
6  where c_id_card like '4201%' and not exists
7      (SELECT *
8       FROM property
9       WHERE pro_type = 1 and c_id = pro_c_id)
10 ORDER BY c_id;
11
12
13
14
15  /* end of your code */
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.5	51.56
2	共有2组测试集，其中有2组测试结果不匹配	0.5	51.56



第8关 持有两张信用卡的用户

最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 11

src/test8/test8.sql

```
1  -- 8) 查询持有两张(含)以上信用卡的用户的名称、身份证号、手机号。
2  --  请用一条SQL语句实现该查询：
3
4  select c_name, c_id_card, c_phone
5  from client
6  where c_id_in
```

```

6   where c_id in (
7       select b_c_id
8       from bank_card
9       -- where b_c_id = c_id
10      where b_type = '信用卡'
11      group by b_c_id
12      having count(*) >= 2
13  )
14  order by c_id;
15
16

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.48	56.44
2	共有2组测试集，其中有2组测试结果不匹配	0.48	55.82
3	共有2组测试集，其中有2组测试结果不匹配	0.48	55.28
4	共有2组测试集，其中有2组测试结果不匹配	0.48	51.56
5	共有2组测试集，其中有2组测试结果不匹配	0.49	51.56



第9关 购买了货币型基金的客户信息

最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 7

src/test9/test9.sql

```

1  -- 9) 查询购买了货币型(f_type='货币型')基金的用户的名称、电话号、邮箱。
2  -- 请用一条SQL语句实现该查询:
3  SELECT c_name,c_phone,c_mail
4  FROM client
5  WHERE c_id IN
6      (SELECT pro_c_id
7       FROM fund,property
8       WHERE pro_pif_id = f_id AND f_type = '货币型' and pro_type = 3)
9  ORDER BY c_id;
10
11

```

```
12
13
14  /* end of your code */
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.39	51.36
2	共有2组测试集，其中有2组测试结果不匹配	0.38	51



第10关 投资总收益前三名的客户

最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 10

src/test10/test10.sql

```
1  -- 10) 查询当前总的可用资产收益(被冻结的资产除外)前三名的客户的名称、身份证号及其总收益，按收益降序输出，总收益命名为total_income。不考虑并列排名形。
2  -- 请用一条SQL语句实现该查询：
3
4  SELECT c_name, c_id_card, Sum_pro.sum_income AS total_income
5  FROM (SELECT pro_c_id, SUM(pro_income)
6        FROM
7        (SELECT *
8         FROM property
9         where pro_status = '可用') as temp1
10       GROUP BY pro_c_id
11       ) AS Sum_pro(id, sum_income), client
12 WHERE c_id = id
13 ORDER BY Sum_pro.sum_income DESC limit 3;
14
15  /* end of your code */
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.23	48.74
2	共有2组测试集，其中有2组测试结果不匹配	0.24	48.74

3	共有2组测试集，其中有2组测试结果不匹配	0.22	48.74
4	共有2组测试集，其中有2组测试结果不匹配	0.24	48.74
5	共有2组测试集，其中有2组测试结果不匹配	0.24	48.74
6	共有2组测试集，其中有2组测试结果不匹配	0.24	48.74
7	共有2组测试集，其中有2组测试结果不匹配	0.23	48.74
8	共有2组测试集，其中有2组测试结果不匹配	0.23	48.74
9	共有2组测试集，其中有2组测试结果不匹配	0.22	48.74



最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 5

src/test11/test11.sql

```
1  -- 11) 给出黄姓用户的编号、名称、办理的银行卡的数量(没有办卡的卡数量计为0),持卡数量命名为number_of_cards,
2  --      按办理银行卡数量降序输出,持卡数量相同的,依客户编号排序。
3  -- 请用一条SQL语句实现该查询:
4  select c_id, c_name, count(b_number) as number_of_cards
5  from client left outer join bank_card on (client.c_id = bank_card.b_c_id)
6  where c_name like '黄%'
7  group by c_id
8  order by number_of_cards desc, c_id
9
10
11
12
13
14
15
16
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.23	48.74
2	共有2组测试集，其中有2组测试结果不匹配	0.22	48.74
3	共有2组测试集，其中有2组测试结果不匹配	0.23	48.74
4	共有2组测试集，其中有2组测试结果不匹配	0.22	48.74
5	共有2组测试集，其中有2组测试结果不匹配	0.22	48.74
6	共有2组测试集，其中有2组测试结果不匹配	0.23	48.74



```
1  -- 12) 综合客户表(client)、资产表(property)、理财产品表(finances_product)、保险表(insurance)和
2  -- 基金表(fund)，列出客户的名称、身份证号以及投资总金额（即投资本金，
3  -- 每笔投资金额=商品数量*该产品每份金额），注意投资金额按类型需要查询不同的表，
4  -- 投资总金额是客户购买的各类资产(理财,保险,基金)投资金额的总和，总金额命名为total_amount。
5  -- 查询结果按总金额降序排序。
6  -- 请用一条SQL语句实现该查询：
7  select c_name, c_id_card, ifnull(ii_n, 0) + ifnull(ff_n, 0) + ifnull(pp_n, 0) as total_amount
8  from (((client left outer join
9  (
10     select pro_c_id, sum(pro_quantity * i_amount)
11     from property join insurance
12     on(property.pro_pif_id = insurance.i_id)
13     where pro_type = 2
14     group by pro_c_id
15 ) as i_n(ii_id, ii_n) on (c_id = i_n.ii_id))
16 left outer join
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.32	50.42
2	共有2组测试集，其中有2组测试结果不匹配	0.31	50.42
3	共有2组测试集，其中有2组测试结果不匹配	0.32	50.42
4	共有2组测试集，其中有2组测试结果不匹配	0.32	50.08
5	共有2组测试集，其中有2组测试结果不匹配	0.38	60.92
6	共有2组测试集，其中有2组测试结果不匹配	0.36	57.18
7	共有2组测试集，其中有2组测试结果不匹配	0.37	56.5
8	共有2组测试集，其中有2组测试结果不匹配	0.36	55.65

9	共有2组测试集，其中有2组测试结果不匹配	0.34	54.03
10	共有2组测试集，其中有2组测试结果不匹配	0.35	50.92

<>

第13关 客户总资产

最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 61

src/test13/test13.sql

```
1  -- 13) 综合客户表(client)、资产表(property)、理财产品表(finances_product)、
2  --  保险表(insurance)、基金表(fund)和投资资产表(property)，
3  --  列出所有客户的编号、名称和总资产，总资产命名为total_property。
4  --  总资产为储蓄卡余额，投资总额，投资总收益的和，再扣除信用卡透支的金额
5  --  (信用卡余额即为透支金额)。客户总资产包括被冻结的资产。
6  --  请用一条SQL语句实现该查询：
7
8  -- 13) 综合客户表(client)、资产表(property)、理财产品表(finances_product)、
9  --  保险表(insurance)、基金表(fund)和投资资产表(property)，
10 --  列出所有客户的编号、名称和总资产，总资产命名为total_property。
11 --  总资产为储蓄卡余额，投资总额，投资总收益的和，再扣除信用卡透支的金额
12 --  (信用卡余额即为透支金额)。客户总资产包括被冻结的资产。
13 --  请用一条SQL语句实现该查询：
14
15 select c_id, c_name, ifnull(ii_n, 0) + ifnull(ff_n, 0) + ifnull(pp_n, 0) + ifnull(in_n, 0) + ifnull(b_n, 0) - ifnull(o_n, 0) as
total property
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.26	49.7
2	共有2组测试集，其中有2组测试结果不匹配	0.25	49.7
3	共有2组测试集，其中有2组测试结果不匹配	0.27	49.7
4	共有2组测试集，其中有2组测试结果不匹配	0.26	49.7

5	共有2组测试集，其中有2组测试结果不匹配	0.26	49.7
6	共有2组测试集，其中有2组测试结果不匹配	0.26	49.7
7	共有2组测试集，其中有2组测试结果不匹配	0.26	49.7
8	共有2组测试集，其中有2组测试结果不匹配	0.26	49.7
9	共有2组测试集，其中有2组测试结果不匹配	0.26	49.7
10	共有2组测试集，其中有2组测试结果不匹配	0.26	49.7



第14关 第N高问题

最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 8

src/test14/test14.sql

```
1  -- 14) 查询每份保险金额第4高保险产品的编号和保险金额。
2  --      在数字序列8000,8000,7000,7000,6000中，
3  --      两个8000均为第1高，两个7000均为第2高，6000为第3高。
4  -- 请用一条SQL语句实现该查询：
5  SELECT i_id, i_amount
6  FROM insurance
7  WHERE i_amount = (
8      SELECT i_amount
9      FROM insurance
10     GROUP BY i_amount
11     ORDER BY i_amount DESC
12     LIMIT 3, 1
13 );
14
15
16
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.26	49.7



第15关 基金收益两种方式排名

最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 16

src/test15/test15.sql

```
1  -- 15) 查询资产表中客户编号，客户基金投资总收益，基金投资总收益的排名(从高到低排名)。  
2  -- 总收益相同时名次亦相同(即并列名次)。总收益命名为total_revenue，名次命名为rank。  
3  -- 第一条SQL语句实现全局名次不连续的排名，  
4  -- 第二条SQL语句实现全局名次连续的排名。  
5  
6  -- (1) 基金总收益排名(名次不连续)  
7  
8  select  
9      pro_c_id,  
10     sum(pro_income) as total_revenue,  
11     rank() over(order by sum(pro_income) desc) as "rank"  
12 from property  
13 where pro_type = 3  
14 group by pro_c_id  
15 order by total_revenue desc, pro_c_id;  
16
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)
1	评测通过	0.56	51.66



第16关 持有完全相同基金组合的客户

最新代码 未通过本题, 展示最后提交评测的代码详情

src/test16/test16.sql

```
1  -- 16) 查询持有相同基金组合的客户对, 如编号为A的客户持有的基金, 编号为B的客户也持有, 反过来, 编号为B的客户持有的基金, 编号为A的客户也持有, 则(A,B)
   为持有相同基金组合的二元组, 请列出这样的客户对。为避免过多的重复, 如果(1,2)为满足条件的元组, 则不必显示(2,1), 即只显示编号小者在前的那一对, 这一组
   户编号分别命名为c_id1,c_id2。
2
3  -- 请用一条SQL语句实现该查询:
4
5
6
7
8
9
10
11 /* end of your code */
```

本关无评测记录



第17关 购买基金的高峰期

最新代码 未通过本题, 展示最后提交评测的代码详情

src/test17/test17.sql

```
1  -- 17 查询2022年2月购买基金的高峰期。至少连续三个交易日, 所有投资者购买基金的总金额超过100万(含), 则称这段连续交易日为投资者购买基金的高峰期。只有
   交易日才能购买基金, 但不能保证每个交易日都有投资者购买基金。2022年春节假期之后的第1个交易日为2月7日, 周六和周日是非交易日, 其余均为交易日。请列出高峰时
   的日期和当日基金的总购买金额, 按日期顺序排序。总购买金额命名为total_amount。
2  -- 请用一条SQL语句实现该查询:
3
4
5
6
7
8
9  /* end of your code */
```

本关无评测记录



第18关 至少有一张信用卡余额超过5000元的客户信用卡总余额

最新代码 未通过本题，展示最后提交评测的代码详情

src/test18/test18.sql

```
1  -- 18) 查询至少有一张信用卡余额超过5000元的客户编号，以及该客户持有的信用卡总余额，总余额命名为credit_card_amount。
2  -- 请用一条SQL语句实现该查询：
3
4
5
6
7
8
9  /* end of your code */
10
11
12
```

本关无评测记录



第19关 以日历表格式显示每日基金购买总金额

最新代码 未通过本题，展示最后提交评测的代码详情

src/test19/test19.sql

```
1  -- 19) 以日历表格式列出2022年2月每周每日基金购买总金额，输出格式如下：
2  -- week_of_trading Monday Tuesday Wednesday Thursday Friday
3  --           1
4  --           2
5  --           3
6  --           4
7  -- 请用一条SQL语句实现该查询：
8
9
10
11
```

```
12  
13  
14  
15  /* end of your code */
```

本关无评测记录