sql典型例题

select新建字段

select新建字段

1. 聚合函数，avg函数求比率, 所需的类别为1，其余为0即可。结合case when then else end语句：

avg函数求比率 SQL

```sql
select seller_id, avg(case when feedback_score = 1 then 1 else 0 end) rate
from trans_fdback
group by seller_id
having rate >= 0.5;
```

2. 新建分组，if语句新建分组，作为后续group by依据

if语句新建分组 SQL

```sql
-- 计算25岁以上和以下的用户数量
select
if(age>=25,'25岁及以上','25岁以下') age_cut,
count(distinct device_id) number
from user_profile
group by age_cut
```

age_cut	number
25岁以下	4
25岁及以上	3

3. 计算用户的平均次日留存率, lead窗口函数，利用日期位移。别忘了with a as(xx)跟的括号

计算用户的平均次日留存率 SQL

```sql
with a as
(select distinct device_id, date from question_practice_detail),
b as
(select
a.device_id as id,
a.date as date1,
lead(a.date,1) over(partition by device_id order by a.date) as date2
from a)

select
count(id) / (select count(distinct device_id,date) from  question_practice_detail) as avg_ret
from b
where DATEDIFF(date2,date1) =1
```

4. 找出每个学校GPA最低的同学

way1:关联子查询 SQL

```sql
SELECT device_id, university, gpa
FROM user_profile u
WHERE gpa = (SELECT min(gpa)
             FROM user_profile
             WHERE university = u.university)
```

way2:窗口函数 SQL

```sql
-- 注意：rank()使用rank别名会报错，我也不知道为啥
with a as
(Select device_id,university,gpa,
row_number() over(partition by university order by gpa) as rk
From user_profile)

Select device_id,university,gpa
From a
where rk = 1
```

5. 删除exam_record表中作答时间小于5分钟整的记录中，开始作答时间最早的3条记录。

delete from也可用limit子句 SQL

```sql
DELETE FROM exam_record
WHERE TIMESTAMPDIFF(MINUTE,start_time,submit_time) < 5
ORDER BY start_time
LIMIT 3
```

1.truncate使用语法

truncate的作用是清空表或者说是截断表，只能作用于表。truncate的语法很简单，后面直接跟表名即可，例如：`truncate table tbl_name` 或者 `truncate tbl_name` 。

执行truncate语句需要拥有表的drop权限，从逻辑上讲，truncate table类似于delete删除所有行的语句或drop table然后再create table语句的组合。为了实现高性能，它绕过了删除数据的DML方法，因此，它不能回滚。尽管truncate table与delete相似，但它被分类为DDL语句而不是DML语句。

2.truncate与drop,delete的对比

上面说过truncate与delete，drop很相似，其实这三者还是与很大的不同的，下面简单对比下三者的异同。

- truncate与drop是DDL语句，执行后无法回滚；delete是DML语句，可回滚。
- truncate只能作用于表；delete，drop可作用于表、视图等。
- truncate会清空表中的所有行，但表结构及其约束、索引等保持不变；drop会删除表的结构及其所依赖的约束、索引等。
- truncate会重置表的自增值；delete不会。
- truncate不会激活与表有关的删除触发器；delete可以。
- truncate后会使表和索引所占用的空间会恢复到初始大小；delete操作不会减少表或索引所占用的空间，drop语句将表所占用的空间全释放掉。

6.

7. 求distinct a,b后的数据，然后求汇总数据条数，可以group by rollup (a), 分类汇总。不可以count (distinct a,b)

group by rollup()分类汇总 SQL

```sql
select count(distinct grid_guid)
from ev_schedule.ev_schedule_oho_task_liugang_dispatch_48h_di
where pt between '${dt-7}' and '${dt-1}'
group by rollup (dispatch_date)
```

8. 某一列就算都是一个值，也不可以当作常数用！该写进group by还是要写的。要不然就新建一个常数项。也就是说，select里面只有常数项、聚合键和聚合函数，如果没有group by，那么也就没有聚合键，就算某列全部都是同一个数，既不是聚合键，也不可以当作常数项用，就不能写进select里面。

9. sql执行顺序：
    a. from
    b. where 初步筛选
    c. group by 分组
    d. 计算聚合函数，配合with roll up 或者 cube
    e. having对分组结果筛选
    f. **select 选出指定列**
    g. distinct 去重
    h. order by排列
    i. limit/offset指定返回行
10. 请找到2021年10月有过取消订单记录的司机，计算他们每人全部已完成的有评分订单的平均评分及总体平均评分，保留1位小数。先按driver_id升序输出，再输出总体情况。注意：select最后运行，所以ifnull是对group by rollup之后的数据进行操作，所以‘总体’指的是总汇总行。不过这个前提是drive_id是not null的，所以不会出现null值对应的分组，从而不会造成混淆。

group by xx with rollup,mysql　　SQL

```sql
select   IFNULL(driver_id,'总体') driver_id
        ,round(avg(grade),1) from tb_get_car_order
where driver_id in (select distinct driver_id from tb_get_car_order
                    where start_time is NULL and date_format(order_time,'%Y-%m')='2021-10')
group by driver_id with rollup
```

11. 在sql处理中，窗口函数都是最后一步执行，而且仅位于order by子句之前。因此，group by后的聚合函数可以和窗口函数连用。
12. 求日均，千万不要忘记count(distinct date)！！！

注意：count(distinct date(order_time))

SQL

```sql
#每个城市中评分最高的司机平均评分、日均接单量和日均行驶里程数。
with t1 as
(
select  b.city
        ,a.driver_id
        ,avg(grade) as avg_grade
        ,count(*)/count(distinct date(order_time)) as avg_order_num
        ,(sum(mileage)/count(distinct date(order_time)) as avg_mileage
        ,rank() over(partition by b.city order by avg(grade) desc) as rk
        --窗口函数，结合group by的聚合函数使用
from tb_get_car_order a join tb_get_car_record b
on a.order_id = b.order_id
group by b.city,a.driver_id
)

select  city
        ,driver_id
        ,avg_grade
        ,avg_order_num
        ,avg_mileage
from    t1
where rk = 1
```