尚硅谷大数据技术之企业 SQL 面试题

(作者：尚硅谷大数据研发部)

版本：V1.0

第 1 题

我们有如下的用户访问数据

userId	visitDate	visitCount
u01	2017/1/21	5
u02	2017/1/23	6
u03	2017/1/22	8
u04	2017/1/20	3
u01	2017/1/23	6
u01	2017/2/21	8
u02	2017/1/23	6
u01	2017/2/22	4

要求使用 SQL 统计出每个用户的累积访问次数，如下表所示：

用户 id	月份	小计	累积
u01	2017-01	11	11
u01	2017-02	12	23
u02	2017-01	12	12
u03	2017-01	8	8
u04	2017-01	3	3

数据：

```
u01     2017/1/21       5
u02     2017/1/23       6
u03     2017/1/22       8
u04     2017/1/20       3
u01     2017/1/23       6
u01     2017/2/21       8
u02     2017/1/23       6
u01     2017/2/22       4
```

1）创建表

```
create table action
     (userId string,
     visitDate string,
     visitCount int)
row format delimited fields terminated by "\t";
```

1）修改数据格式

```
select
    userId,
    date_format(regexp_replace(visitDate,'/','-'),'yyyy-
MM') mn,
    visitCount
from
    action;t1
```

2）计算每人单月访问量

```
select
   userId,
   mn,
   sum(visitCount) mn_count
from
   t1
group by userId,mn;t2
```

3）按月累计访问量

```
select
   userId,
   mn,
   mn_count,
   sum(mn_count) over(partition by userId order by mn)
from t2;
```

最终 SQL

```
select
   userId,
   mn,
   mn_count,
   sum(mn_count) over(partition by userId order by mn)
from
(   select
       userId,
       mn,
       sum(visitCount) mn_count
   from
        (select
           userId,
           date_format(regexp_replace(visitDate,'/','-
'),'yyyy-MM') mn,
           visitCount
        from
           action)t1
group by userId,mn)t2;
```

第 2 题 京东

有 50W 个京东店铺，每个顾客访客访问任何一个店铺的任何一个商品时都会产生一条

访问日志，访问日志存储的表名为 Visit，访客的用户 id 为 user_id，被访问的店铺名称为 shop，请统计：

```
u1 a
u2 b
u1 b
u1 a
u3 c
u4 b
u1 a
u2 c
u5 b
u4 b
u6 c
u2 c
u1 b
u2 a
u2 a
u3 a
u5 a
u5 a
u5 a
```

建表：

```
create table visit(user_id string,shop string) row format delimited fields terminated by '\t';
```

1）每个店铺的 UV（访客数）

```
select shop,count(distinct user_id) from visit group by shop;
```

2）每个店铺访问次数 top3 的访客信息。输出店铺名称、访客 id、访问次数

（1）查询每个店铺被每个用户访问次数

```
select shop,user_id,count(*) ct
from visit
group by shop,user_id;t1
```

（2）计算每个店铺被用户访问次数排名

```
select shop,user_id,ct,rank() over(partition by shop order by ct) rk
from t1;t2
```

（3）取每个店铺排名前 3 的

```
select shop,user_id,ct
from t2
where rk<=3;
```

（4）最终 SQL

```
select
```

```
    shop,
    user_id,
    ct
from
    (select
        shop,
        user_id,
        ct,
        rank() over(partition by shop order by ct) rk
    from
        (select
            shop,
            user_id,
            count(*) ct
        from visit
        group by
            shop,
            user_id)t1
    )t2
where rk<=3;
```

第 3 题

已知一个表 STG.ORDER，有如下字段:Date，Order_id，User_id，amount。请给出 sql 进行统计:数据样例:2017-01-01,10029028,1000003251,33.57。

建表：

```
create table order_tab(dt string,order_id string,user_id string,amount decimal(10,2)) row format delimited fields terminated by '\t';
```

1）给出 2017 年每个月的订单数、用户数、总成交金额。

```
select
   date_format(dt,'yyyy-MM'),
   count(order_id),
   count(distinct user_id),
   sum(amount)
from
   order_tab
group by date_format(dt,'yyyy-MM');
```

2）给出 2017 年 11 月的新客数(指在 11 月才有第一笔订单)

```
select
   count(user_id)
from
   order_tab
   group by user_id
having date_format(min(dt),'yyyy-MM')='2017-11';
```

第 4 题

有一个 5000 万的用户文件(user_id，name，age)，一个 2 亿记录的用户看电影的记录文件(user_id，url)，根据年龄段观看电影的次数进行排序？

第 5 题

有日志如下，请写出代码求得所有用户和活跃用户的总数及平均年龄。（活跃用户指连续两天都有访问记录的用户）

日期 用户 年龄
2019-02-11,test_1,23
2019-02-11,test_2,19
2019-02-11,test_3,39
2019-02-11,test_1,23
2019-02-11,test_3,39
2019-02-11,test_1,23
2019-02-12,test_2,19
2019-02-13,test_1,23
2019-02-15,test_2,19
2019-02-16,test_2,19

```
create table user_age(dt string,user_id string,age int)row
format delimited fields terminated by ',';
```

1）按照日期以及用户分组，按照日期排序并给出排名

```
select
   dt,
   user_id,
   min(age) age,
   rank() over(partition by user_id order by dt) rk
from
   user_age
group by
   dt,user_id;t1
```

2）计算日期及排名的差值

```
select
   user_id,
   age,
   date_sub(dt,rk) flag
from
   t1;t2
```

3）过滤出差值大于等于 2 的，即为连续两天活跃的用户

```
select
   user_id,
   min(age) age
from
   t2
group by
   user_id,flag
having
   count(*)>=2;t3
```

4）对数据进行去重处理（一个用户可以在两个不同的时间点连续登录），例如：a 用户在 1 月 10 号 1 月 11 号以及 1 月 20 号和 1 月 21 号 4 天登录。

```
select
   user_id,
   min(age) age
from
   t3
group by
   user_id;t4
```

5）计算活跃用户（两天连续有访问）的人数以及平均年龄

```
select
   count(*) ct,
   cast(sum(age)/count(*) as decimal(10,2))
from t4;
```

6）对全量数据集进行按照用户去重

```
select
   user_id,
   min(age) age
from
   user_age
group by
   user_id;t5
```

7）计算所有用户的数量以及平均年龄

```
select
   count(*) user_count,
   cast((sum(age)/count(*)) as decimal(10,1))
from
   t5;
```

8）将第 5 步以及第 7 步两个数据集进行 union all 操作

```
select
   0 user_total_count,
   0 user_total_avg_age,
   count(*) twice_count,
   cast(sum(age)/count(*)          as          decimal(10,2))
twice_count_avg_age
```

```
from
(
   select
   user_id,
   min(age) age
from
   (select
   user_id,
   min(age) age
from
   (
    select
   user_id,
   age,
   date_sub(dt,rk) flag
from
   (
    select
      dt,
      user_id,
      min(age) age,
      rank() over(partition by user_id order by dt) rk
    from
      user_age
    group by
      dt,user_id
    )t1
    )t2
group by
   user_id,flag
having
   count(*)>=2)t3
group by
   user_id
)t4

union all

select
   count(*) user_total_count,
   cast((sum(age)/count(*)) as decimal(10,1)),
   0 twice_count,
   0 twice_count_avg_age
from
   (
     select
       user_id,
       min(age) age
     from
       user_age
```

```
    group by
      user_id
  )t5;t6
```

9）计算最终结果

```
select
   sum(user_total_count),
   sum(user_total_avg_age),
   sum(twice_count),
   sum(twice_count_avg_age)
from
(select
  0 user_total_count,
  0 user_total_avg_age,
  count(*) twice_count,
  cast(sum(age)/count(*)          as          decimal(10,2))
twice_count_avg_age
from
(
  select
  user_id,
  min(age) age
from
  (select
  user_id,
  min(age) age
from
  (
   select
  user_id,
  age,
  date_sub(dt,rk) flag
from
  (
   select
     dt,
     user_id,
     min(age) age,
     rank() over(partition by user_id order by dt) rk
   from
     user_age
   group by
     dt,user_id
   )t1
   )t2
group by
  user_id,flag
having
  count(*)>=2)t3
group by
```

```
   user_id
)t4

union all

select
   count(*) user_total_count,
   cast((sum(age)/count(*)) as decimal(10,1)),
   0 twice_count,
   0 twice_count_avg_age
from
   (
      select
         user_id,
         min(age) age
      from
         user_age
      group by
         user_id
   )t5)t6;
```

第 6 题

请用 sql 写出所有用户中在今年 10 月份第一次购买商品的金额，表 ordertable 字段（购买用户：userid，金额：money，购买时间：paymenttime(格式：2017-10-01)，订单 id：orderid）

第 7 题

现有图书管理数据库的三个数据模型如下：

图书（数据表名：BOOK）

序号	字段名称	字段描述	字段类型
1	BOOK_ID	总编号	文本
2	SORT	分类号	文本
3	BOOK_NAME	书名	文本
4	WRITER	作者	文本
5	OUTPUT	出版单位	文本
6	PRICE	单价	数值（保留小数点后2位）

读者（数据表名：READER）

序号	字段名称	字段描述	字段类型
1	READER_ID	借书证号	文本
2	COMPANY	单位	文本

3	NAME	姓名	文本
4	SEX	性别	文本
5	GRADE	职称	文本
6	ADDR	地址	文本

借阅记录（数据表名：BORROW LOG）

序号	字段名称	字段描述	字段类型
1	READER_ID	借书证号	文本
2	BOOK_D	总编号	文本
3	BORROW_ATE	借书日期	日期

（1）创建图书管理库的图书、读者和借阅三个基本表的表结构。请写出建表语句。

（2）找出姓李的读者姓名（NAME）和所在单位（COMPANY）。

（3）查找“高等教育出版社”的所有图书名称（BOOK_NAME）及单价（PRICE），结果按单价降序排序。

（4）查找价格介于 10 元和 20 元之间的图书种类(SORT）出版单位（OUTPUT）和单价（PRICE），结果按出版单位（OUTPUT）和单价（PRICE）升序排序。

（5）查找所有借了书的读者的姓名（NAME）及所在单位（COMPANY）。

（6）求”科学出版社”图书的最高单价、最低单价、平均单价。

（7）找出当前至少借阅了 2 本图书（大于等于 2 本）的读者姓名及其所在单位。

（8）考虑到数据安全的需要，需定时将“借阅记录”中数据进行备份，请使用一条 SQL 语句，在备份用户 bak 下创建与“借阅记录”表结构完全一致的数据表 BORROW_LOG_BAK. 井且将“借阅记录”中现有数据全部复制到 BORROW_1.0G_ BAK 中。

（9）现在需要将原 Oracle 数据库中数据迁移至 Hive 仓库，请写出“图书”在 Hive 中的建表语句（Hive 实现，提示：列分隔符|；数据表数据需要外部导入：分区分别以 month__part、day__part 命名）

（10）Hive 中有表 A，现在需要将表 A 的月分区 201505 中 user__id 为 20000 的 user__dinner 字段更新为 bonc8920，其他用户 user__dinner 字段数据不变，请列出更新的方法步骤。（Hive 实现，提示：Hlive 中无 update 语法，请通过其他办法进行数据更新）

第 8 题

有一个线上服务器访问日志格式如下（用 sql 答题）

时间 接口 ip 地址

2016-11-09 14:22:05 /api/user/login 110.23.5.33

2016-11-09 14:23:10 /api/user/detail57.3.2.16

2016-11-09 15:59:40 /api/user/login 200.6.5.166

……

求 11 月 9 号下午 14 点（14-15 点），访问/api/user/login 接口的 top10 的 ip 地址

```
select
   ip,
   count(*) ct
from
   web
where
   date_format(dt,'yyyy-MM-dd HH')>='2016-11-09 14'
   and
   date_format(dt,'yyyy-MM-dd HH')<'2016-11-09 15'
   and
   interface='/api/user/login'
group by
   ip
order by
   ct desc
limit 10;
```

第 9 题

有一个充值日志表如下：

CREATE TABLE `credit_log`

(

`dist_id` int（11）DEFAULT NULL COMMENT '区组 id',

`account` varchar（100）DEFAULT NULL COMMENT '账号',

`money` int(11) DEFAULT NULL COMMENT '充值金额',

`create_time` datetime DEFAULT NULL COMMENT '订单时间'

)ENGINE=InnoDB DEFAUILT CHARSET-utf8

请写出 SQL 语句，查询充值日志表 2015 年 7 月 9 号每个区组下充值额最大的账号，要求结果：

区组 id，账号，金额，充值时间

```
select
   *
from
   credit_log t1
where
(
    select
       count(*)
```

```
    from
       credit_log t2
    where
       t1.dist_id=t2.dist_id
       and
       t1.money>t2.money
)>2;
```

第 10 题

有一个账号表如下，请写出 SQL 语句，查询各自区组的 money 排名前十的账号（分组取前 10）

CREATE TABIE `account`

(

`dist_id` int（11）

DEFAULT NULL COMMENT '区组 id',

`account` varchar（100）DEFAULT NULL COMMENT '账号',

`gold` int（11）DEFAULT NULL COMMENT '金币'

PRIMARY KEY （`dist_id`，`account_id`），

）ENGINE=InnoDB DEFAULT CHARSET-utf8

第 11 题

1）有三张表分别为会员表（member）销售表（sale）退货表（regoods）

（1）会员表有字段 memberid（会员 id，主键）credits（积分）；

（2）销售表有字段 memberid（会员 id，外键）购买金额（MNAccount）；

（3）退货表中有字段 memberid（会员 id，外键）退货金额（RMNAccount）；

2）业务说明：

（1）销售表中的销售记录可以是会员购买，也可是非会员购买。（即销售表中的 memberid 可以为空）

（2）销售表中的一个会员可以有多条购买记录

（3）退货表中的退货记录可以是会员，也可是非会员

（4）一个会员可以有一条或多条退货记录

查询需求：分组查出销售表中所有会员购买金额，同时分组查出退货表中所有会员的退货金额，把会员 id 相同的购买金额-退款金额得到的结果更新到会员表中对应会员的积分字

段（credits）

第 12 题 百度

现在有三个表 student（学生表）、course(课程表)、score（成绩单），结构如下：

```
create table student
(
    id bigint comment ‘学号’,
    name string comment ‘姓名’,
    age bigint comment ‘年龄’
);
create table course
(
    cid string comment ‘课程号，001/002 格式’,
    cname string comment ‘课程名’
);
Create table score
(
    Id bigint comment ‘学号’,
    cid string comment ‘课程号’,
    score bigint comment ‘成绩’
) partitioned by(event_day string)
```

其中 score 中的 id、cid，分别是 student、course 中对应的列请根据上面的表结构，回答下面的问题

1）请将本地文件（/home/users/test/20190301.csv）文件，加载到分区表 score 的 20190301 分区中，并覆盖之前的数据

2）查出平均成绩大于 60 分的学生的姓名、年龄、平均成绩

3）查出没有‘001’课程成绩的学生的姓名、年龄

4）查出有‘001’\’002’这两门课程下，成绩排名前 3 的学生的姓名、年龄

5）创建新的表 score_20190317，并存入 score 表中 20190317 分区的数据

6）如果上面的 score 表中，uid 存在数据倾斜，请进行优化，查出在 20190101-20190317 中，学生的姓名、年龄、课程、课程的平均成绩

7）描述一下 union 和 union all 的区别，以及在 mysql 和 HQL 中用法的不同之处？

8）简单描述一下 lateral view 语法在 HQL 中的应用场景，并写一个 HQL 实例

第 13 题

订单表 order

id	user_id	city	order_time
1	A	深圳	2018-01-01 10:10:30
2	B	上海	2018-01-10 10:10:30
3	C	北京	2018-02-01 12:10:30
4	A	深圳	2018-01-14 21:10:30
5	C	成都	2018-01-18 10:10:30
6	D	广州	2018-01-22 10:10:30
7	Y	南宁	2018-03-16 04:10:30
8	F	天津	2018-03-29 09:10:30
9	T	北京	2018-01-09 10:10:30
10	F	天津	2018-01-01 09:10:30
...	...	...	...

用户表 user

user_id	user_name
A	张三
B	李四
C	王五
D	刘六
Y	赵雪
F	陈俊
T	韩梅
...	...

题目 1：假设订单表中只有 2018 年 1 月到 2018 年 3 月，共三个月的数据，运营同学需要以下表格，请写出 SQL：

city	2018 年 1 月订单数	2018 年 2 月订单数	2018 年 3 月订单数
A	2	0	0
...	...	...	...

题目 2：查找用户的最后一次购买时间及其城市，表格如下：

user_name	city	最后购买时间
张三	深圳	2018-01-14 10:10:30
王五	北京	2018-02-01 10:10:30
...	...	...

题目 3（10 分）：计算用户画像，规则是用户在哪个城市下单次数最多就属于哪个城市，但如果下单最多的城市有多个，则取最多的城市中下单时间最大的那个作为用户所在城市，输出表格如下：

user_id	city
A	深圳
C	北京
...	...

第 14 题

原始数据

班级	日期	数学	语文	英语
01	0705	90	80	85
01	0705	88	79	66
01	0705	50	80	96
01	0705	60	90	30
01	0705	40	70	59

要变成

班级	日期	总人数	科目	及格占比	不及格占比	是否是重要科目
01	0705	5	数学	60	40	是
01	0705	5	语文	100	0	是
01	0705	5	英语	60	40	否