大数据技术之企业SQL面试题

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 |
| 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/detail 57.3.2.16

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

