SQL

（黄色高亮是我的答案，写法不唯一，各位酌情参考）

1.

PRIME TABLE ("PRIME")
Customer ID
Start Date
End Date

ORDERS TABLE ("ORDERS")
Customer ID
Order ID
Order Day
Product ID
Quantity Sold

Tell me the customer who bought the most units on each day

```
Select Order Day, Customer ID
From (
select Order Day, Customer ID, dense_rank() over (partition by  Orderday order by sum(Quantity Sold) desc)
From Orders
Group by Order Day, Customer ID ) x
Where r = 1;
```

2.

1. 两个table：Employee，Department
Employee schema: eid, did, ename,...（还有些column不记得了，不过不重要。。）
Department schema: did, dname
问题是找出在invalid department的employee，其实就是看employee table里谁的did 不在 Department table 里。

```
select employee
from Employee as e left join department as d
on e.did = d.did
where dname is NULL;

Select employee
```

From employee e
Where not exists(
Select * from department d where e.did = d.did);

Select employee
From Employee
Where did not in (select did from department);

3.

Table 1 长这样（3个1）
1
1
1
Table 2 长这样（6个1）
1
1
1
1
1
1
问题是left/right/inner join 的result分别都什么样。

Left/right/inner 18

4.

Department table 大概长这样

Department	Revenue	Month
A	8000	Jan
B	9000	Jan
C	10000	Feb
A	7000	Feb
A	6000	Mar

...
问题是写个query变成这样的result format

Department	Jan_Revenue	Feb_Revenue	Mar_Revenue...	Dec_Revenue
A	8000	7000	6000	...
B...				

Select Department,
Case when Month = Jan then sum(Revenue) as Jan Revenue,

```
Case when Month = Feb then sum(Revenue) as Feb Revenue,
…
From Department
Group by Department;

Select Department,
[Jan] as Jan_revenue,
[Feb] as Feb_revenue,
…
From (select * from department) sourcetable

Pivot (sum(revenue) for Month in ([Jan],[Feb],…) as t

Declare
@query Nvarchar(Max) = '',
@cols nvarchar(Max) = '';

Select @cols = quotename(month) + ','
From (select distinct Month from department) x
Order by month          (field(month, 'Jan','Feb'…)

Set @cols = left(@cols, len(@cols) -1)

Set @query = '
Select * from (
Select * from department) x
Pivot (sum() for month in ('+ @cols +')) as t;' ;
Exec(@query)
```

5.

date, device, answers
2019-01-01 echo 10.
2019-01-02 echo 20
2019-01-03 echo 30
.
2019-01-01 dot 10
2019-01-02 dot 10
问by decive， trailing 7 days的 answer总和是什么, 其实就是想得到一个 date, device, trailing 7 days'answers 这样一个目标表

```
select a.date, a.device, sum(b.answer)
```

```
from table a join table b
on a.device = b.device and a.date>=b.date and a.date-6<=b.date  (a.date-b.date between 0 and 6)
group by 1, 2
```

可以用sum preceding。。。。。具体可以看这个链接
https://docs.aws.Amazon.com/redshift/latest/dg/r_Examples_of_sum_WF.html

```
Select date, device, sum(answers) over (partition by device  order by date rows between 6 preceding and current row) from table;
```

6.

Top Three books sold in each city, during the last 3 month

```
Select city, books, r
From (select city, books, dense_rank() over ( partition by city order by sum(quantity) desc) as r
From t where  order_date >= dateadd(month,-3,getdate()
Group by city, books) x
Where r <= 3;
```

```
With sales as (select city, book, sum(q) as q from table group by city,book where order_date >= dateadd(month, -3, getdate()) )
Select * from sales s1 where 3 > (select count(distincts2.q) from sales s2 where s2.q>s1.q and s1.city = s2.city);
```

7.

给了date，然后要count total for past month by week

```
select year(date) as year, month(date) as month, ceiling(right(date,2)/7) as week_number, count(OrderItemsID)
from ORDER
where year(date)=2019
and month(date)=10
group by year(date), month(date), ceiling(right(date,2)/7)
```

8.

ORDERS: ORDER_ID, CUSTOMER_ID
ORDER_ITEM: ORDER_ID, ITEM_ID
ITEMS: ITEM_ID, ITEM_NAME

CUSTOMERS: CUSTOMER_ID

Q: Write a sql query to find the # of customers who purchased both kindle and Alexa.

```
select count(distinct customer_id)
From (select customer_id
from orders as t1 left join order_item as t2 on t1.order_id = t2.order_id
join item as t3 on t2.item_id = t3.item_id
where t3.item_name in ('Alexa','Kindle')
group by t1.customer_id
having count(distinct item_name) = 2) x;

with alexa
as
(
select
c.customer_id
from orders o
inner join customers c on c.customer_id = o.customer_id
inner join orderitem oi on oi.order_id = o.order_id
inner join items i on i.item_id = oi.item_id

where oi.item_name ='Alexa'
),kindle
as
(
select
c.customer_id
from orders o
inner join customers c on c.customer_id = o.customer_id
inner join orderitem oi on oi.order_id = o.order_id
inner join items i on i.item_id = oi.item_id

where oi.item_name ='kindle'
)
select
count(1)
from alexa a
inner join kindle b on a.customer_id = b.customer_id
```

9.

You have an orders table and a book catalog table below:

TABLE 1 – Book Transaction
MARKETPLACE_ID NUMERIC(38)
TXN_DAY DATE
CUSTOMER_ID VARCHAR(50)
ASIN VARCHAR(10)
QUANTITY NUMERIC(38)

TABLE 2 - Catalog
MARKETPLACE_ID NUMERIC(38)
ASIN VARCHAR(10)
TITLE_NAME VARCHAR(100)

Q1: How do you find the top 100 books sold for the current month?

You now have a third table:
TABLE 3 – Magazine Transaction
MARKETPLACE_ID NUMERIC(38)
TXN_DAY DATE
CUSTOMER_ID VARCHAR(50)
ASIN VARCHAR(10)
QUANTITY NUMERIC(38)

Q2: How do you find only the customers that purchased book and not magazines?

```
Select top 100 with ties title_name, sum(q)
From t1 Inner Join t2
Where datepart(month,txn_day) = datepart(month, getdate())
Group by asin
Order by sum(q) desc;
```

```
Select distinct customer_id
From t1
Where not exists (select * from t3 where t1.customer_id = t3.customer_id)
```

10.

Table X: customer_id, pro_key
Table Y: pro_key

Q: 找出买过Y中所有产品的customer_id

```
Select customer_id
From x inner join y on x.pro_key = y.pro_key
Group by x.customer_id
Having count(distinct x.pro_key) = (select count(distinct y.pro_key ) from y);
```

11.

Table A:
Columns: actor, dir, date
Q: 找出跟同一个导演至少合作过3次的演员

```
Select actor
From t
Group by actor, dir
Having count(date) >= 3;
```

12.

Q1. Table: seller_id, date, status (block, suspend, reinstate). 求问在reinstate状态后（假设只有一次reinstate) 有过block or suspend状态的seller_id

Q2. Same table as 1. 求问1st status after the latest reinstate.

Q3. Table: id, order_date, product. 求问在2010买过A且在2018买过B的id

```
Q1.Select seller_id
From t
Group by seller_id
Having Max(case when status != reinstate then date end) > Max(case when status = reinstate then date end) ;

With reinstate as (
Select seller_id, max(date) as date
From t
Group by seller_id
Where status = reinstate)

Select seller_id
From t inner join reinstate
On t.seller_id = reinstate.seller_id and t.date > reinstate.date
```

```
Where status in (block, suspend);

Q2.With reinstate as
(select seller_id, max(date) as reinstate_date
from Table
where status = reinstate
group by 1)

select distinct seller_id, first_value(status) over (partition by seller_id order by date) as
first_status
from Table t
left join reinstate a
on t.seller_id=a.seller_id and t.date>a.reinstate_date;

with next_t as(
select *, LEAD(status,1) over (partition by id order by date) as next_status
from table)
Select next_status
from next_t
where status=reinstate;

Q3.
Select id
From t t1 , t t2
Where t1.id = t2.id and year(t1.order_date) = 2010 and t1. Product = A and year(t2.order_date)
= 2018 and t2.product = B;

select id
from Table
where (year(order_date)=2010 and product=A)
or (year(order_date)=2018 and product=B)
group by id
having count (distinct year(order_date))=2

Select distinct id from table t1 where year(order_date) = 2010 and product = a and exists (select
* from table t2 where year(order_date) = 2018 and product = b and t1.id = t2.id)
```

13.

给一个flight的table，有depature city和 arrival city，求unique的不论顺序的 组合

```
SELECT DISTINCT
```

```
CASE WHEN departure > arrival THEN arrival ELSE departure AS departure,
CASE WHEN departure > arrival THEN departure ELSE arrival AS arrival
FROM flight
```

14.

SQL就是所有order history+ 每个customer在各个产品品类下面place过的首个 和最后一个order的记录，求1).每天各产品品类下的order中，是某顾客在该品类首个order的比例；2).每天所有order中，是某顾客首个order的比例

Order_id, date, category

customer_id, category, type, order_id

```
Select date, category, count(c.order_id)/count(o.order_id) as pctg
From o left join (select * from c
Where c.type = 'first') x on o.order_id = x.order_id
Group by date, category;
```

15.

CostumerID TITLE DATE
找出每个用户第一次看的电影中最受欢迎的那个

```
With first_movie as(select *, rank() over (partition by customerid order by date) as r from t where r = 1)
Select top 1 title
From first_movie
Group by title
Order by count(customerid) desc;

Select top 1 *
From t inner join(
Select id, min(date) from t group by id) x
On t.id = x.id and t.date = x.date
Group by t.title
Order by count(t.id);
```

Tech Questions

1. how you import data; database 的一些概念; view的pros and cons

2. view和table哪个更快 为什么？那table update了 view会跟着update 吗

3. 考了个Data Modelling的题，用star schema设计一个Amazon Book的Data Model，然后给我出了道题：

我的dimension table有customer(id, name, address)，store(id, city, address)，date(id, year, month, day)，book(id, name, author);factor table是除了dimension table的id还有order的quantity和price

既然是amazon book感觉store在这里不是很合理，但除了store也没啥大毛病
个人感觉用这4个dimension会更好，如果他没有别的req，只是amazon book然后freestyle的话

product_dimension,
customer_dimension,
time_dimension,
promotion_dimension,

Fact table: Key + Unit_sale + sale + cost

4. 先设计几个表用于记录购物页面的(以下是我的思路，童鞋们自由发挥)
— 露珠后来反应过来应该先问是设计OLTP还是OLAP的，这样设计思路很不一样，但鉴于这个只是为了之后写code而用的structure，个人认为偏向于Analysis purpose 所以应该是OLAP，在这里面试官没有过分为难我， 毕竟面的不是architect XD；但以下俩种思路给各位借鉴希望大家答题的时候可以加分。

—a. OLTP（尽量做到normalized，答题思路不局限,合理即可）
ordertable
orderid pk int,
customerid fk int,
orderdate datetime,
orderamount numeric

customertable
customerid pk int,
name varchar,
address(country, state,zip)

producttable
productid pk int,-baidu 1point3acres
price

order-producttable(conjuction table)
orderid fk int,
productid fk int-baidu

—b. OLAP

FACT_Order
OrderID int,
ProductID int,
ProductQuantity int,
CustomerID int,
Datekey int,
OrderAmount numeric

Dim_Customer
Customerid pk int,
Name varchar,
country varchar,
state varchar,
zipcode int

Dim_Product
略
Dim_Date
略

5. 假设有一些csv文件，关于用户order的信息，文件名如下格式：
USA_order.csv
UK_order.csv. From 1point 3acres bbs

文件里面的内容如下格式:
order_id, user_id

问题：要把以上内容放到database warehouse里，问怎样设计schema
follow up question: if use country_id, 用什么方式加country_id到table 中

6. how to optimize SQL query？

7. 了解清楚query plan（execution plan）的应用

8. sql performance optimization 相关的，问的是用group by还是 rank 的sql run的时间最短， total 1B row数据，两种情况：1. 3 个类型产品，每个类型333333333 rows 2. 1000个类型产品，每个类型1000000 row

9. stats 什么是null hypothesis variance 和 std的关系

10. 简单解释什么是P value