## SQL Test

**Question 1.**

*Uber has several different products (varying by city); for instance, in New York City, it offers uberX, uberXL, UberBLACK, and UberSUV.*

*In some cases Uber practices “cross-dispatch”; e.g., generally a driver/vehicle for uberXL can also accept uberX trips. In specific cities, UberBLACK (or equivalent) drivers will be sent uberX requests.*

Think about ways in which cross dispatch might make querying Uber’s data complicated. Write a paragraph or two about common types of analyses that might fail if the researcher didn’t think carefully enough about cross dispatch.

[If it matters, assume Vertica 9.0; feel free to look up the documentation. Don’t worry about syntax errors in the queries below, as these queries are known to run]

---------------------------------------------------------------------------

**Question 2.**

*The Uber data warehouse stores records on cities and trips in tables named cities and trips, respectively. Uber offers different products in different cities; each city-product, e.g., UberBLACK in New York City, is a distinct “vehicle view”. When a user requests a trip, the product requested for that trip is recorded in column request\_vehicle\_view\_id (which joins to a table, vehicle\_views, on the vehicle\_views.id column).*

**cities**

|  |  |  |
| --- | --- | --- |
| id | name | timezone |
| 1 | san\_francisco | America/Los\_Angeles |
| ... | ... | ... |
| 5 | new\_york | America/New\_York |

**trips**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| request\_at | driver\_id | city\_id | fare | status | request\_vehicle\_view\_id |
| 2011-04-05 18:04:36+00 | 8134971 | 5 | 10.31 | completed | 8 |
| ... | ... | ... | ... | ... | ... |
| 2015-01-13 10:45:06+00 | 3425215 | 1 | 13.37 | canceled | 2 |

**vehicle\_views**

|  |  |  |
| --- | --- | --- |
| id | city\_id | name |
| 2 | 1 | uberX |
| 3 | 1 | uberBLACK |
| ... | ... | ... |
| 8 | 5 | uberX |

Please write a query that answers the following: “For each driver in New York City, report the number of trips completed by week for the weeks of March, 2014.”

---------------------------------------------------------------------------

**Question 3.**

*The following query is intended to compute the total number of driver-partners, total hours supplied by driver-partners (using a table, driver\_shifts, that gives the time in seconds that drivers were on-app), total gross fares, and average fare per hour, for drivers in New York City in January, 2015:*

WITH  
driver\_fares AS (  
 SELECT  
 tt.driver\_id,  
 vv.name AS vehicle\_view\_name,  
 SUM(tt.fare) AS total\_fares  
 FROM trips tt  
 INNER JOIN cities cc ON cc.id = tt.city\_id  
 INNER JOIN vehicle\_views vv ON cc.id = vv.city\_id AND vv.id = tt.request\_vehicle\_view\_id  
 WHERE 1=1  
 AND cc.name = 'new\_york'  
 AND request\_at >= '2015-01-01'  
 AND request\_at < '2015-02-01'  
 AND status = 'completed'  
 GROUP BY 1,2  
),  
driver\_times AS (  
 SELECT  
 driver\_id,  
 SUM(seconds\_on\_shift)/3600 AS hours\_on\_shift  
 FROM driver\_shifts ds  
 INNER JOIN cities cc ON cc.id = ds.city\_id  
 WHERE 1=1  
 AND cc.name = 'new\_york'  
 AND occurred\_at >= '2015-01-01'  
 AND occurred\_at < '2015-02-01'  
 GROUP BY driver\_id  
)  
SELECT  
 COUNT(driver\_id) AS n\_drivers,  
 SUM(hours\_on\_shift) AS aggregate\_hours\_supplied,  
 SUM(total\_fares) AS aggregate\_fares,  
 AVG(total\_fares / hours\_on\_shift) AS avg\_fares\_per\_hour  
FROM driver\_times dt  
INNER JOIN driver\_fares df ON dt.driver\_id = df.driver\_id

Comparing the results of this query to counts of drivers and supply hours in another tool generally considered reliable, we find that this query’s results for these quantities are inflated by some factor. Why? Are aggregate\_fares and avg\_fares\_per\_hour also wrong? In what directions?

**Bonus:** What else is wrong with this query?

---------------------------------------------------------------------------

## Analysis Test

We’ve attached a JSON dataset of client logins from an Uber city on the eastern seaboard of the United States. Using this, please do the following:

1. Using your analysis tool of choice (e.g., Python or R), generate a graph showing the long-term trend of logins for this city.

2. Add a best fit line or curve to this graph, and include any relevant metrics/statistics to quantify the quality of fit.

3. Discuss any significant trends or deviations you observe in the dataset.

4. Repeat this analysis by graphing logins by day of week and by hour of day, noting any interesting findings. Based on what you find, why do you think this is?