-- question 1

SELECT dayofweek(sub.dates) as "day of week", avg(sub.total/sub.difference) as "avg\_hourly\_rate"

FROM (

SELECT dash\_end\_time as "dates",

sum(total\_pay) as "total",

sum(datediff(second, dash\_end\_time,dash\_start\_time)/3600.0)) as "hours\_worked"

FROM Dash

GROUP BY dash\_end\_time

)

sub

GROUP BY dayofweek(sub.dates);

-- In the first query, we are taking the sum of the total pay for each entry and taking the difference in hours for when the shift

-- was started and ended to calculate the total hours worked.

-- The sums are grouped by the dash\_end\_time to get the totals for each date.

-- The outer query selects the average of the hourly rates for each day of the week and groups them by the number of day that corresponds to the day of the week.

-- question 2

SELECT sum(d.total\_pay)/3 as "total pay" ,

da.submarket\_id as "submarket\_id"

FROM Dash d

JOIN Dasher da

ON (d.dasher\_id=da.id)

WHERE hour(dash\_start\_time) >= 11 and hour(dash\_end\_time) <= 14

GROUP BY d.dasher\_id

HAVING submarket\_id = 3;

-- Here we are joining both the Dash and Dasher tables to be able to extract information on the submarket\_id and the earnings per hours for lunch.

-- We divide the sum of the total pay by 3 since we are calculating from 11am to 2pm (3 hours) to obtain our hourly rate.

-- We specify criteria for the times we would like to look at (11am and 2pm), group everything by the dasher\_id and then we make sure we filter for the submarket\_id at the end.

-- question 3

SELECT sub2.effective\_hourly\_rate, sub2.dasher\_email

FROM (

SELECT sub1.total/sub1.hours\_worked as "effective\_hourly\_rate",

sub1.email as "dasher\_email",

sub1.dasher\_id as "dasher\_id"

FROM (

SELECT sum(d.total\_pay) as "total",

sum(datediff(second,d.dash\_start\_time,d.dash\_end\_time)/3600.0)) as "hours\_worked",

da.email\_address as "email",

d.dasher\_id

FROM Dash d

JOIN Dasher da

ON (d.dasher\_id=da.id)

WHERE dash\_end\_time BETWEEN NOW() - INTERVAL 30 DAY AND NOW()

GROUP BY dasher\_id

)

sub1

order by sub1.total/sub1.hours\_worked desc

) sub2

WHERE median\_val IN (

SELECT AVG(dd.sub2.effective\_hourly\_rate) as median\_val

FROM (

SELECT d.val, @rownum:=@rownum+1 as `row\_number`, @total\_rows:=@rownum

FROM data d, (SELECT @rownum:=0) r

WHERE d.val is NOT NULL

ORDER BY d.val

) as dd

WHERE dd.row\_number IN ( FLOOR((@total\_rows+1)/2), FLOOR((@total\_rows+2)/2) ));

--This one was quite challenging... My logic goes as follows:

--Obtain the total pay for each dasher for the last 30 days

--Obtain the total hours worked for each dasher for the last 30 days

--Divide the total pay by the total hours worked to obtain the effective hourly rate for each dasher and group by each dasher\_id for the last 30 days

--Here it gets a bit tricky as mySQL does not have a built in function to calculate the median (50th percentile), but my logic was to write

--a function that stored the median\_val and then write a query that goes as such:

-- SELECT effective\_hourly\_rate, dasher\_email

-- FROM table

-- WHERE effective\_hourly\_rate < median

--There are other ways to programmatically obtain the same result if one was to use different tools such as Python/Pandas as well with ease.

--My approach would be to perform the same calculations and take advantage of Pandas's built-in median function to create a conditional that will

--return all of the hourly rates that are lower than the median.