Analyst Technical Assessment

Please use the csv data set along with the below data table dictionaries to answer the below questions.

Table Descriptions

User Table

product_account_id	user identifier specific and unique to each user. Primary and unique key in user table
registration_date	date that the user signed up for FanDuel
first_deposit_date	date the user made their first deposit on the FanDuel site/app
first_paid_entry_date	date the user made their first paid entry on the FanDuel site/app
first_deposit_amount	dollar amount of the users first deposit. Relates to first_deposit_date
registration_paid_media_partner_name	media partner that is associated with driving the user's registration. All partners are paid media partners aside from "ORGANIC"
first_deposit_paid_media_partner_name	media partner that is associated with driving the user's first deposit. All partners are paid media partners aside from "ORGANIC"
first_paid_entry_paid_media_partner_name	media partner that is associated with driving the user's first paid entry. All partners are paid media partners aside from "ORGANIC"
registration_state	the state the user was in when they signed up for FanDuel. Relates to registration_date

product_account_id	registration_date	first_deposit_date	first_paid_entry_date	first_deposit_amount
74299	2/4/2018			
74994	2/4/2018			
75110	4/2/2018	4/2/2018	4/2/2018	11.43
75557	4/3/2018			
75589	4/3/2018	4/3/2018	4/3/2018	5.72
75922	4/3/2018			
76000	6/14/2018			
76120	6/14/2018			
76236	6/14/2018	6/14/2018	6/16/2018	14.29
76740	6/14/2018			
76793	6/14/2018			
77271	8/18/2018			
77395	8/18/2018	8/18/2018	8/18/2018	14.29
77794	8/18/2018	8/18/2018	8/18/2018	5.72
77807	8/18/2018			
77907	8/18/2018			
78278	9/4/2018			
78465	9/4/2018			
78642	9/4/2018			
78965	9/4/2018	9/6/2018	9/6/2018	2.86
79172	9/10/2018	9/10/2018	9/13/2018	5.72

registration_paid_media_partner_name	first_deposit_paid_media_partner_name	first_paid_entry_paid_media_partner_name	registration_state
NUMBERFIRE	INFERRED ORGANIC		
TWITTER	USER REFERRAL	USER REFERRAL	
TWITTER	USER REFERRAL	USER REFERRAL	
SEARCH			GEORGIA
GOOGLE	TAPJOY	UNTRACKED	GEORGIA
SNAPCHAT			WISCONSIN
ROTOGRINDERS			
NUMBERFIRE			ca
SNAPCHAT	ADPERIO	UNTRACKED	MICHIGAN
TWITTER			ARIZONA
INSTAGRAM			FLORIDA
SNAPCHAT			KENTUCKY
SNAPCHAT	Topgolf	Topgolf	
TWITTER	SNAPCHAT	SNAPCHAT	NORTH CAROLINA
INSTAGRAM	USER REFERRAL	USER REFERRAL	
SNAPCHAT			nc
FACEBOOK			TENNESSEE
GOOGLE			nv
FACEBOOK			NEBRASKA

Entry Table

product_account_id	user identifier specific and unique to each user. Is not unique on the entry table
entry_week	the week a user entered a contest. Week is noted as the first day of the week

entry_sport	the sport associated with the user's entry
contest_type	the type of contest associated with the user's entry
entry_source	the platform associated with the user's entry
weekly_entry_fee	the summed total of an users weekly entries
weely_total_entry_winnings	the summed total of an user's weekly winnings represented as a gross total. To find net winnings you would do "weekly_entry_fee" - "weekly_total_entry_winnings"
total_entries	the total count of entries a user made in a week
avg_contest_size	the average size of the contest the users entered into based on the combination of "entry_sport", "contest_type", and "entry_source" in a week

product_account_id	entry_week	entry_sport	contest_type	entry_source
74299				
74994	2/5/2018	NBA	freeroll	MOBILE ENTRY
74994	2/12/2018	NBA	freeroll	MOBILE ENTRY
75110	4/2/2018	MLB	auto league	MOBILE ENTRY
75110	4/2/2018	MLB	freeroll	DESKTOP ENTRY
75110	4/2/2018	MLB	tournament	DESKTOP ENTRY
75110	4/2/2018	MLB	tournament	MOBILE ENTRY
75110	4/2/2018	NBA	auto league	DESKTOP ENTRY
75110	4/2/2018	NBA	auto league	MOBILE ENTRY
75110	4/2/2018	NBA	freeroll	MOBILE ENTRY
75110	4/2/2018	NBA	tournament	DESKTOP ENTRY
75110	4/2/2018	NBA	tournament	MOBILE ENTRY
75110	4/2/2018	PGA	tournament	MOBILE ENTRY
75110	4/9/2018	NBA	auto league	DESKTOP ENTRY
75110	4/9/2018	NBA	auto league	MOBILE ENTRY
75110	4/9/2018	NBA	freeroll	DESKTOP ENTRY
75110	4/9/2018	NBA	freeroll	MOBILE ENTRY
75110	4/9/2018	NBA	tournament	DESKTOP ENTRY
75110	4/9/2018	NBA	tournament	MOBILE ENTRY

weekly_entry_fee	weely_total_entry_winnings	total_entries	avg_contest_size
		1	1427
		1	82567
1.14	1.03	2	55
		1	348
1.71		1	110
1.71		2	1052
2.29		2	100
5.72		4	100
	0.39	1	654
9.14	2.82	7	846
22	26.57	15	6186
1.14		1	595
4.57		3	84
5.72		5	83
		1	10
		1	32286
5.14	2.29	3	1921
28.15	18.25	13	3164
6.06	3.43	2	13164
13.03	15.72	6	6378

Data Querying

1. Given the above data sets show what month and year (eg. March 2019, April 2018, etc.) has the largest number of registrations?

```
SELECT Year, Month
FROM (SELECT EXTRACT(YEAR FROM registration_date) AS Year,
            EXTRACT(MONTH FROM registration_date) AS Month,
            COUNT(*) AS Registrations
     FROM user_data
     GROUP BY Year, Month
     ORDER BY Registrations DESC) AS monthly_registrations
LIMIT 1;
```

YEAR	MONTH
2020	9

2. Given the above data sets show the top 10 ad partners by total users per First Paid Entry Credit (column "*first_paid_entry_paid_media_partner_name*").
   a. Be sure to not include users who have not made a first paid entry.

```
SELECT AdPartner
FROM (SELECT first_paid_entry_paid_media_partner_name AS AdPartner,
             COUNT(*) AS UserCount
      FROM user_data
      WHERE first_paid_entry_date IS NOT NULL
      GROUP BY AdPartner
      ORDER BY UserCount DESC) AS top_ad_partners
LIMIT 10;
```

```
ADPARTNER

ORGANIC

USER REFERRAL

UNTRACKED

SNAPCHAT INSTALLS

ADPERIO

FEEDMOB

APPLE SEARCH ADS

FPBOUNTY

FACEBOOK INSTALLS

SNAPCHAT
```

3. Like the above show the top 10 Ad Partners, grouped by First Paid Entry Credit (column "*first_paid_entry_paid_media_partner_name*") and ranked by total entry fees for users whose first paid entry data was in September.
   a. In addition to total entry fees, include the total count of entries.

```
SELECT u.first_paid_entry_paid_media_partner_name AS AdPartner,
       SUM(e.weekly_entry_fee) AS TotalEntryFees,
       SUM(e.total_entries) AS TotalEntries
FROM user_data AS u
RIGHT JOIN entry_data AS e
ON u.product_account_id = e.product_account_id
WHERE EXTRACT(MONTH FROM u.first_paid_entry_date) = 9
GROUP BY AdPartner
ORDER BY TotalEntryFees DESC
LIMIT 10;
```

ADPARTNER	TOTALENTRYFEES	TOTALENTRIES
ORGANIC	56846.56	26632
USER REFERRAL	55871.93	42160
ADPERIO	28833.86	18225
UNTRACKED	23211.43	14652
FACEBOOK INSTALLS	5438.22	2784
APPLE SEARCH ADS	4676.78	2167
MEDIAMATH	3275.12	1295
YEP ADS	2573.32	811
MUNDOMEDIA	1936.93	882
FEEDMOB	1610.23	654

4. For users who registered in 2019, which month, based on a user's first_paid_entry_date, saw the largest number of unique users make an NBA entry at any point in time?

```
SELECT FirstPaidEntryDateMonth
FROM (SELECT EXTRACT(MONTH FROM u.first_paid_entry_date) AS FirstPaidEntryDateMonth,
             COUNT(DISTINCT u.product_account_id) AS UniqueUsers
      FROM user_data AS u
      RIGHT JOIN entry_data AS e
      ON u.product_account_id = e.product_account_id
      WHERE EXTRACT(YEAR FROM u.registration_date) = 2019 AND
            e.ENTRY_SPORT = 'NBA' AND
            u.first_paid_entry_date IS NOT NULL
      GROUP BY FirstPaidEntryDateMonth
      ORDER BY UniqueUsers DESC) AS monthly_uniques
LIMIT 1;
```

FIRSTPAIDENTRYDATEMONTH
9

5. What were the top 5 sports in both 2018 and 2019 based on total entries for users that made their first paid entry within one week of registering? Include the average first_deposits_amount, and average time between registration and first paid entry per sport and year as well.
   a. Please write this as a single query using the "rank()" function.
   b. Note, we want to know the top 5 sports for each year not overall in 2018 and 2019 combined.

```
SELECT YearlyRanking,
       Year,
       EntrySport,
       FirstDeposit,
       TimeFirstEntry
FROM (SELECT RANK() OVER (PARTITION BY Year ORDER BY Entries DESC) AS YearlyRanking
      Year,
      EntrySport,
      Entries,
      FirstDeposit,
      TimeFirstEntry
      FROM (SELECT EXTRACT (YEAR FROM e.entry_week) AS Year,
                   e.entry_sport AS EntrySport,
                   SUM(e.total_entries) AS Entries,
                   AVG(u.first_deposit_amount) AS FirstDeposit,
                   AVG(u.first_paid_entry_date - u.registration_date) AS TimeFirstEntry
            FROM user_data AS u
            RIGHT JOIN entry_DATA AS e
            ON u.product_account_id = e.product_account_id
            WHERE EXTRACT(YEAR FROM e.entry_week) < 2020 AND
                  (u.first_paid_entry_date - u.registration_date) < 7
            GROUP BY Year, EntrySport) AS year_sport_data) AS ranked_data
WHERE YearlyRanking <= 5
ORDER BY Year ASC, YearlyRanking ASC;
```

YEARLYRANKING	YEAR	ENTRYSPORT	FIRSTDEPOSIT	TIMEFIRSTENTRY
1	2018	NFL	16.30	0.81
2	2018	NBA	12.70	0.77
3	2018	MLB	15.69	0.56
4	2018	NHL	16.48	0.66
5	2018	CFB	11.41	0.89
1	2019	NFL	22.07	0.80
2	2019	NBA	20.61	0.67
3	2019	MLB	20.49	0.64
4	2019	NHL	21.33	0.61
5	2019	PGA	23.14	0.86