

1. Variables like stake, GGR, bet type and sport name can help identify users' betting preferences and history. With those variables in mind, distinct groups can be created, and incentives can be personalized for each of them.

For the upcoming football season, I would focus on users whose first or second favorite sport is football based how much of their net stake was bet on football. From that, a "football stake" segment can be created, based on users' total stake in football during the last football season. Then I would determine whether a user's profit was positive or negative overall (positive if their GGR is negative, negative if their GGR is positive) to create a "user profit" segment, and lastly, a segment related to the type of bet that the user prefers. The breakdown below refers to the type of incentives I would recommend for each group based on their betting experiences.

| | | | |
|--|-----------------------|--------------------|-----------------|
| Users that have not been profitable for the company and are likely to return without much incentive. | Football Stake | User Profit | Bet Type |
| | High | Positive | Straight |
| | High | Positive | Parlay |
| | High | Negative | Straight |
| | High | Negative | Parlay |
| | Medium | Positive | Straight |
| | Medium | Positive | Parlay |
| | Medium | Negative | Straight |
| | Medium | Negative | Parlay |
| | Low | Positive | Straight |
| | Low | Positive | Parlay |
| | Low | Negative | Straight |
| | Low | Negative | Parlay |

2. Code:

```
SELECT COUNT(DISTINCT(playerid))  
FROM user  
WHERE DATEPART('year', activation_date) = 2019;
```

This is assuming that "first activity" in the description of the activation_date field refers to a bet. For a more accurate number of new users, we could include a filter using registered_date to show only users who also registered in 2019.

3. Code:

```
SELECT u.playerid,
       alias,
       email,
       SUM(CASE WHEN DATEPART('year', betplaceddate) = 2019 THEN stake END) AS total_
stake_2019,
       ROUND(SUM(CASE WHEN DATEPART('year', betplaceddate) = 2019 THEN winning END)::
decimal / total_stake_2019, 2) AS margin_2019,
       MAX(betplaceddate) AS last_bet_placed_date,
       SUM(payment_amount) AS total_approved_deposits
FROM user u
JOIN activity a ON u.playerid = a.playerid
JOIN deposit d ON u.playerid = d.playerid AND d.payment_stauts = 'a'
GROUP BY u.playerid, alias, email;
```

4. Code:

```
SELECT ROUND(COUNT(DISTINCT CASE WHEN bettype = 'parlay' then betid END)::DECIMA
L / COUNT(DISTINCT(betid)) * 100, 2) AS pct_parlays,
       DATEPART('month', betpaceddate) AS month
FROM activity
WHERE betplaceddate BETWEEN '2019-01-01' and '2019-04-30'
GROUP BY month;
```

5. Code:

```
SELECT COUNT(registration_date)/COUNT(activation_date) AS reg_to_act,
       DATEPART('month', betpaceddate) AS month
FROM user
WHERE activation_date BETWEEN '2019-01-01' and '2019-04-30'
GROUP BY month;
```