**Richard Cheung**

**SQL Test**

For an upcoming football campaign FanDuel are going to send an email to existing players letting them know headline tournaments and new site features for the season. In addition there will be targeted offers to certain players based on their transactional history.

As an analyst your job is to help define the segmentation and then use the SQL database to generate lists of users. The data tables you have at your disposal are below:

**USER TABLE** – This is where we store information about the user’s account

|  |  |
| --- | --- |
| FIELD | DESCRIPTION |
| user\_id | Unique identifier for users |
| username | Username of user on site |
| email | Email address of user |
| deposit\_count | The total number of deposits the player has made |
| account\_suspended | A binary variable to show if the account has been suspended |
| registration\_date | Shows the date on which the user initially registered on the site |

**ENTRY TABLE** – This is where all the information about every entry into games is stored

|  |  |
| --- | --- |
| FIELD | DESCRIPTION |
| entry\_id | Unique identifier for each entry |
| game\_id | Identifier for the game the entry is valid for |
| user\_id | Identifier for user who made the entry |
| entry\_date | The datetime value of when the entry was made |
| entry\_fee | The value of the entry fee into the game (can be 0, 5, 10, 25, 50 or 100) |
| winnings | The value of the winnings from the entry by finishing in a paying final position |
| mobile\_entry | A binary variable to show whether the entry was made on a mobile device |

**GAME TABLE** – This is where all the specifics about games are stored

|  |  |
| --- | --- |
| FIELD | DESCRIPTION |
| game\_id | Unique identifier for each game |
| sport | Indicator of which sport the game is for (MLB, NBA, NFL, NHL) |
| size | The total size of the game from 2 players up to 1,000 players |

**PAYMENTS TABLE** – This is where we store information about and deposits and withdrawals

|  |  |
| --- | --- |
| FIELD | DESCRIPTION |
| payment\_id | Unique identifier for each payment |
| user\_id | Unique identifier for users |
| payment\_type | Identifier for deposits and withdrawals (values either ‘D’ or ‘W’) |
| payment\_date | Shows the date when the payment was processed |
| amount | The value of the transaction |

The CRM Manager has an idea for a test, but wants to know if there are enough users available to make the results significant. There are a couple of filters to apply before he gets in to further detail so asks you to work out the count of users who have made at least one deposit and registered in the previous year (2013). How would you do this with SQL code? Are there any other constraints you might consider adding to give a more accurate estimate?

**SELECT** username

**FROM** user\_table

**WHERE** deposit\_count >= 1

**AND** to\_char(registration\_date, 'YYYY') = '2013';

* I would add account\_suspended as an additional constraint because if a player’s account is suspended, it wouldn’t be useful in the test that the CRM Manager wants to perform.

**SELECT** username

**FROM** user\_table

**WHERE** deposit\_count >= 1

**AND** to\_char(registration\_date, 'YYYY') = '2013’

**AND** account\_suspended = 0;

1. The CRM Manager is happy with the base user size so fills you in on the additional restrictions. He now wants to see how many users have made at least one deposit, registered in 2013 and played NFL in 2013. What would be the SQL code needed to answer this?

**SELECT** distinct username

**FROM** user\_table u, entry\_table e, game\_table g

**WHERE** u.user\_id = e.user\_id

**AND** e.game\_id = g.game\_id

**AND** g.sport = 'NFL'

**AND** u.deposit\_count >= 1

**AND** to\_char(u.registration\_date, 'YYYY') = '2013'

**AND** u.account\_suspended = 0;

1. Now we have the list size finalized it’s time to generate the list of users to be sent the email. With the same constraints as above you need to generate a list with the following fields, so that the CRM Manager can start to build his custom segments:
   1. user\_id
   2. email
   3. total entry fees for NFL in 2013

**SELECT** username, email, sum(entry\_fee) NFL\_entry\_fee

**FROM** user\_table u, entry\_table e, game\_table g

**WHERE** u.user\_id = e.user\_id

**AND** e.game\_id = g.game\_id

**AND** g.sport = 'NFL'

**AND** u.deposit\_count >= 1

**AND** to\_char(u.registration\_date, 'YYYY') = '2013'

**AND** u.account\_suspended = 0

**GROUP BY** username, email;

* 1. total entry fees for the other sports combined in 2013

**SELECT** username, email, sum(entry\_fee) other\_entry\_fee

**FROM** user\_table u, entry\_table e, game\_table g

**WHERE** u.user\_id = e.user\_id

**AND** e.game\_id = g.game\_id

**AND** g.sport <> 'NFL'

**AND** u.deposit\_count >= 1

**AND** to\_char(u.registration\_date, 'YYYY') = '2013'

**AND** u.account\_suspended = 0

**GROUP BY** username, email;

* 1. the number of winning entries the user has ever had

**SELECT** username, email, count(winnings) winnings\_entries

**FROM** user\_table u, entry\_table e, game\_table g

**WHERE** u.user\_id = e.user\_id

**AND** e.game\_id = g.game\_id

**AND** u.deposit\_count >= 1

**AND** to\_char(u.registration\_date, 'YYYY') = '2013'

**AND** u.account\_suspended = 0

**AND** e.winnings > 0

**GROUP BY** username, email;

* 1. the percentage of entries on mobile across ALL sports in 2013

**SELECT** d.username, concat(round(b.mobile\_entry/x.user\_id\_cnt\*100), '%') mobile\_entry\_percentage

**FROM** user\_table d, entry\_table b,

(**SELECT** u.username username, count(e.user\_id) user\_id\_cnt

**FROM** user\_table u, entry\_table e

**WHERE** u.user\_id = e.user\_id

**GROUP BY** u.username) x

**WHERE** d.user\_id = b.user\_id

**AND** d.username = x.username

**AND** b.mobile\_entry = 1

**AND** d.deposit\_count >= 1

**AND** to\_char(d.registration\_date, 'YYYY') = '2013'

**AND** d.account\_suspended = 0;

* 1. the last **paid** entry date of the user

**SELECT** username, max(payment\_date)

**FROM** user\_table u, payment\_table p

**WHERE** u.user\_id = p.user\_id

**AND** u.deposit\_count >= 1

**AND** to\_char(u.registration\_date, 'YYYY') = '2013'

**AND** u.account\_suspended = 0

**AND** p.payment\_type = 'p'

**GROUP BY** username;

* 1. the total net deposit value, i.e. deposits – withdrawals

**SELECT** u.username, y.deposit - x.payment net\_deposit\_value

**FROM** user\_table u,

(**SELECT** user\_id, sum(amount) payment

**FROM** payment\_table

**WHERE** payment\_type = 'p'

**GROUP BY** user\_id) x,

(**SELECT** user\_id, sum(amount) deposit

**FROM** payment\_table

**WHERE** payment\_type = 'D'

**GROUP BY** user\_id) y

**WHERE** u.user\_id = x.user\_id

**AND** u.user\_id = y.user\_id

**AND** u.deposit\_count >= 1

**AND** to\_char(u.registration\_date, 'YYYY') = '2013'

**AND** u.account\_suspended = 0;

How would you create this list in SQL?

**CREATE TABLE** crm\_manager

**AS SELECT** x.username, x.email, NFL\_entry\_fee, other\_entry\_fee, winnings\_entries, mobile\_entry\_percentage, latest\_paid\_entry\_date, net\_deposit\_value

**FROM**

(**SELECT** username, email, sum(entry\_fee) NFL\_entry\_fee

**FROM** user\_table u, entry\_table e, game\_table g

**WHERE** u.user\_id = e.user\_id

**AND** e.game\_id = g.game\_id

**AND** g.sport = 'NFL'

**AND** u.deposit\_count >= 1

**AND** to\_char(u.registration\_date, 'YYYY') = '2013'

**AND** u.account\_suspended = 0

**GROUP BY** username, email) x,

(**SELECT** username, email, sum(entry\_fee) other\_entry\_fee

**FROM** user\_table u, entry\_table e, game\_table g

**WHERE** u.user\_id = e.user\_id

**AND** e.game\_id = g.game\_id

**AND** g.sport <> 'NFL'

**AND** u.deposit\_count >= 1

**AND** to\_char(u.registration\_date, 'YYYY') = '2013'

**AND** u.account\_suspended = 0

**GROUP BY** username, email) y,

(**SELECT** username, email, count(winnings) winnings\_entries

**FROM** user\_table u, entry\_table e, game\_table g

**WHERE** u.user\_id = e.user\_id

**AND** e.game\_id = g.game\_id

**AND** u.deposit\_count >= 1

**AND** to\_char(u.registration\_date, 'YYYY') = '2013'

**AND** u.account\_suspended = 0

**AND** e.winnings > 0

**GROUP BY** username, email) z,

(**SELECT** d.username, concat(round(b.mobile\_entry/x.user\_id\_cnt\*100), '%') mobile\_entry\_percentage

**FROM** user\_table d, entry\_table b,

(**SELECT** u.username username, count(e.user\_id) user\_id\_cnt

**FROM** user\_table u, entry\_table e

**WHERE** u.user\_id = e.user\_id

**GROUP BY** u.username) x

**WHERE** d.user\_id = b.user\_id

**AND** d.username = x.username

**AND** b.mobile\_entry = 1

**AND** d.deposit\_count >= 1

**AND** to\_char(d.registration\_date, 'YYYY') = '2013'

**AND** d.account\_suspended = 0) xx,

(**SELECT** username, max(payment\_date) latest\_paid\_entry\_date

**FROM** user\_table u, payment\_table p

**WHERE** u.user\_id = p.user\_id

**AND** u.deposit\_count >= 1

**AND** to\_char(u.registration\_date, 'YYYY') = '2013'

**AND** u.account\_suspended = 0

**AND** p.payment\_type = 'p'

**GROUP BY** username) xy,

(**SELECT** u.username, y.deposit - x.payment net\_deposit\_value

**FROM** user\_table u,

(**SELECT** user\_id, sum(amount) payment

**FROM** payment\_table

**WHERE** payment\_type = 'p'

**GROUP BY** user\_id) x,

(**SELECT** user\_id, sum(amount) deposit

**FROM** payment\_table

**WHERE** payment\_type = 'D'

**GROUP BY** user\_id) y

**WHERE** u.user\_id = x.user\_id **AND**

u.user\_id = y.user\_id **AND**

u.deposit\_count >= 1 **AND**

to\_char(u.registration\_date, 'YYYY') = '2013'**AND**

u.account\_suspended = 0) xz

**WHERE** x.username = y.username

**AND** x.username = z.username

**AND** x.username = xx.username

**AND** x.username = xy.username

**AND** x.username = xz.username;

1. The Marketing VP is concerned that there has been a movement away from head-to-head (2 player) games over the last 3 years for football. He believes it may be a reason for declining tenure so has requested that you investigate. Specifically he wants to know in each year between 2011 and 2013 what percentage of entries and entry fees were head-to-head games in each user’s first 30 days after registration. Provide the SQL code required to answer his questions.

**SELECT** u.username, to\_char(u.registration\_date, 'YYYY') year, concat(round(x.entry\_fee\_cnt/y.entry\_fee\_cnt\*100),'%') h2h\_entries\_percentage, concat(round(x.h2h\_entry\_fee/y.total\_entry\_fee\*100),'%') h2h\_entries\_fee\_percentage

**FROM** user\_table u,

(**SELECT** u.username, count(e.entry\_fee) entry\_fee\_cnt, sum(e.entry\_fee) h2h\_entry\_fee

**FROM** user\_table u, entry\_table e, game\_table g

**WHERE** u.user\_id = e.user\_id

**AND** e.game\_id = g.game\_id

**AND** to\_char(u.registration\_date, 'YYYY') >= '2011'

**AND** to\_char(u.registration\_date, 'YYYY') <= '2013'

**AND** g.size = 2

**AND** e.entry\_date <= u.registration\_date + 30

**AND** e.entry\_fee > 0

**GROUP BY** u.username) x,

(**SELECT** u.username, count(e.entry\_fee) entry\_fee\_cnt, sum(e.entry\_fee) total\_entry\_fee

**FROM** user\_table u, entry\_table e

**WHERE** u.user\_id = e.user\_id

**AND** to\_char(u.registration\_date, 'YYYY') >= '2011'

**AND** to\_char(u.registration\_date, 'YYYY') <= '2013'

**AND** e.entry\_date <= u.registration\_date + 30

**AND** e.entry\_fee > 0

**GROUP BY** u.username) y

**WHERE** u.username = x.username

**AND** u.username = y.username;