 **🧪 Lab** | GameJet Transactions

**INTRODUCTION:** In order to make the most of a given dataset, you may need to perform engineering of additional calculated fields. Performing calculations to transform the data lets an analyst extract new information that could not be obtained from the data as originally stored.

**HOW IT WORKS:** Follow the prompts in the questions below to investigate your data. Post your answers in the provided boxes: the **yellow boxes** for the queries you write, and **blue boxes** for text-based answers.

PROMPT: In this lab, you’ll help a mobile game company summarize and describe some general patterns in the app usage and spending for one of their games. Revenue from this game is supported by a traditional microtransaction model, where users can spend real money on in-app purchases for items that allow players to access more game features.

Monetization is a difficult problem for companies in the mobile games space, and they need to balance converting as many users from free users to paid users, without being so aggressive that they actually end up driving away more users and actually make less revenue. But before they can start developing strategies for improving the game from a business perspective, the team needs your help to summarize what has been done so far!

**—** Data Set **Description**

The data for this lab (**game\_jet**) describes activity for users of a mobile-based game with microtransactions. In the game, two currencies called ‘passes’ and ‘gems’ are used to unlock more game content and to purchase cosmetic items. These currencies can be earned slowly over regular play, but gems can also be purchased as in-app purchases. You will be working with three tables in this lab: gamejet\_users, gamejet\_sessions, and gamejet\_iaps.

The gamejet\_users table lists users who have downloaded the game app, one per row. The table contains six columns, of which include:

* **udid** - unique id for the user / device
* **install\_date** - date of user installing the app
* **lang** - two-letter code for native language for device
* **country** - two-letter code for user’s country

The gamejet\_sessions table records every session taken by a user where they opened the app. There are four columns in this table:

* **udid** - user / device id
* **ts** - timestamp for start of session
* **date** - pre-truncated date for session
* **session\_num** - cumulative session number for the user

The gamejet\_iaps (In-app purchases) table records all purchases made by users, exchanging real money for in-game currency. There are six columns in this table:

* **udid** - user / device id
* **ts** - timestamp for purchase
* **date** - pre-truncated date for session
* **prod\_type** - product type, can be gems, passes, or a value pack.
* **prod\_qty** - quantity code that is proportional to each product (e.g. a quantity of 1 for gems might actually represent 10 gems)
* **rev** - revenue / price of the purchase, in cents

**— Task 1:** App usage behaviors

How long does a user spend with the app in terms of the number of sessions or the days that they stick with the app? This can have implications for how much time we have to ‘hook’ a player and give them a reason to spend money on the app.

1. Query the gamejet\_users data table to discover the total number of users represented in the data.

(paste your query below 👇)

|  |
| --- |
| select count(distinct udid)  from gamejet\_users |

(write your **answer** below 👇)

|  |
| --- |
| 22576 |

1. Next, write a query to count the number of distinct users in the sessions data table. Recalling your answer from Task 1A, how many users downloaded the app but didn’t actually open it? HINT: A user who downloaded the app will always add a tally in the users table, but won’t show up in sessions if they never use the app.

(paste your query below 👇)

|  |
| --- |
| select count(distinct udid)  from gamejet\_sessions |

(write your **answer** below 👇)

|  |
| --- |
| 22576-22544 = 32 |

1. Write a query that returns the number of sessions made by each user. Sort the output by the number of sessions made, from largest to smallest. How many sessions did the user with the most sessions have with the app?

(paste your query below 👇)

|  |
| --- |
| select  udid,  count(udid) as num\_sessions  from gamejet\_sessions  group by 1  order by 2 desc |

(write your **answer** below 👇)

|  |
| --- |
| 1939 |

1. We should also be interested in how many sessions were taken by a ‘typical’ app user. The average number of sessions with the app is 32.1. Modify your query from part C to only return users who have made more than that many sessions. How many users are there? (Read this from the SQL app interface.) Based on that number, how well does the average represent a ‘typical’ player? HINT: Your query should include the HAVING keyword here!

(paste your query below 👇)

|  |
| --- |
| SELECT  u.udid,  COUNT (session\_num)  FROM gamejet\_users u  JOIN gamejet\_sessions s  ON u.udid = s.udid  GROUP BY u.udid  HAVING COUNT (session\_num) > 32.1  ORDER BY COUNT (session\_num) DESC; |

(write your **answer** below 👇)

|  |
| --- |
| 5262 |

**— Task 2:** In-App Purchases

How large is our user base in terms of who is spending money? Is it a lot of users spending a little, or is it a few users who are spending a lot? Different types of users will likely be responsive to different types of tactics to get them to spend more money.

1. Using the gamejet\_iaps table, write a query to determine the number of users who have made at least one purchase. Remember that each time a user makes a purchase, a new row is added to the gamejet\_iaps table.

(paste your query below 👇)

|  |
| --- |
| select count(distinct udid)  from gamejet\_iaps |

1. Both tables share values in the **udid** column. Write a query that left joins the gamejet\_iaps table to the gamejet\_users table and returns the total (sum) amount of money spent on in-app purchases, grouped by user id. Your query should return every user in the users table, regardless of whether the user made a purchase; the total spent for users with no record in the gamejet\_iaps table should be a *null* value.

(paste your query below 👇)

|  |
| --- |
| select  gu.udid,  sum(rev) as total\_money\_spent  from gamejet\_users gu  left join gamejet\_iaps gi  on gu.udid = gi.udid  group by 1 |

1. Modify your query in part B to create a new feature called persona that segments users into four purchasing personas:

* a “free player” who does not spend any money on in-app purchases,
* a “minnow” who spends less than $20 on in-app purchases,
* a “dolphin” who spends between $20 and $100 on in-app purchases,
* and a “whale” who spends at least $100 on in-app purchases.

Remember: revenue is recorded in terms of cents, so $20 is equal to 2000 cents and $100 is equal to 10 000 cents.

(paste your query below 👇)

|  |
| --- |
| with purchases as (  select  gu.udid,  sum(rev) as total\_money\_spent  from gamejet\_users gu  left join gamejet\_iaps gi  on gu.udid = gi.udid  group by 1  )  select  udid,  total\_money\_spent,  case when total\_money\_spent is NULL then 'free\_player'  when cast(total\_money\_spent as float)/100 < 20.0 then 'minnow'  when cast(total\_money\_spent as float)/100 between 20.0 and 100.0 then 'dolphin'  else 'whale' end as persona  from purchases  order by total\_money\_spent desc |

**— Task 3:** Analyzing Player Engagement Patterns

The answers to the questions in the lab tasks above should paint a picture of the game app being mostly supported off of a relatively small, dedicated group of players. Most users only engage with the app for a short amount of time and end up not buying anything. Even a fair amount of users who do buy something don’t make too many purchases.

1. Another angle of attack you might be interested in looking at is how long it takes for a user to make their first purchase, if they make any. You can do this in two steps. First, take the difference between the user’s install date and the date of each purchase they made to get the amount of days between when they started playing, and when they made a purchase. Their first purchase will then be the purchase with the smallest difference in dates. How many users make a purchase on the first day they have the app? Are there any other periods where we could possibly entice game players to make purchases?

(paste your query below 👇)

|  |
| --- |
| with first\_day\_purchasers as (  select  gu.udid,  min(julianday(date) - julianday(install\_date))  from gamejet\_users gu  join gamejet\_iaps gi  ON gu.udid = gi.udid  group by gu.udid  having min(julianday(date) - julianday(install\_date)) < 1  )  select count(\*)  from first\_day\_purchasers |

(write your **answer** below 👇)

|  |
| --- |
| 724 purchaser on the first day. |