**Warner Bros Analytics (Technical Test)**

| LoginDate | UserName | Platform | GameEdition |
|---|---|---|---|
| 2020-03-20 | PaulM | Ps4 | Standard |
| 2020-03-10 | JohnL | XboxOne | Premium |
| 2020-03-09 | RingoS | PC | Deluxe |
| ... | ... | ... | ... |

1) In SQL, what statement would count unique users who logged in after Mar 5, 2020 and before Mar 19, 2020?

**Answer:**

```
Select Distinct UserName
from GameSessionStart
where LoginDate Between '2020-03-05' and '2020-03-19';
```

2) In SQL, what statement would yield the number of logins on XboxOne in March 2019?

**Answer:**

```
select count(LoginDate)
from GameSessionStart
where DATEPART(YEAR, LoginDate) = '2019' AND DATEPART(MONTH, LoginDate) = '03'
and Platform = 'XboxOne';
```

3) In SQL, what statement would give us the breakdown of game edition ownership numbers by platform?

**Answer:**

```
Select Platform, GameEdition, count(*)
From GameSessionStart
Group by Platform, GameEdition
```

4) In SQL, how do we get the number of users whose user name contains ‘John’?

**Answer:**

```
Select count(UserName)
From GameSessionStart
Where UserName like ‘%John%’
```

5) In SQL, how do we get WAU for the first 12 weeks of 2020?

**Answer:**

*WAU = Weekly Active Users* ***(assumption)***

```
Select datepart(week, ‘2020-01-01’), count(distinct UserName)
From GameSessionStart
Where datepart(week, ‘2020-01-01’) <= 12
And GameSessionStart.LoginDate between '2020-01-01' and '2020-03-31'
Group by datepart(week, ‘2020-01-01’)
```

| FirstLoginDate | UserName | Country | BetaParticipant |
|---|---|---|---|
| 2019-02-20 | JackMa | China | 1 |
| 2020-01-17 | ElonM | US | 0 |
| 2019-01-19 | RichardB | UK | 1 |
| ... | ... | ... | ... |

6) In SQL, how would you count the total number of days a user has been active in the game?

**Answer:**

**My thoughts (Scenario 1):** I am assuming we want to total number of days a user has been active. Every time a user logged in, GameSessionStart table had a entry with date, username etc. We would group by user and then count number of distinct days (in case user logged in more than once a day).

**Scenario 2:** I am not sure if you want to get number of total days since FirstLoginDate. For that calculation we would get Latest or Max date by user and then do LatestDate minus FirstLoginDate to get total number of days user has been on platform. We would join GameSessionStart table with UserDetail table using GameSessionStart.username = UserDetail.username (join to get all records from GameSessionStart with maximum date)

**Scenario 1:**

```
Select UserName, count(Distinct LoginDate)
From GameSessionStart
Group by UserName, LoginDate
```

**Scenario 2:**

```
SELECT UD.UserName, DATEDIFF(day, UD.FirstLoginDate, groupedtt.MaxDateTime) AS
NumberofDays
FROM UserDetail UD
INNER JOIN
   (SELECT Username, MAX(LoginDate) AS MaxDateTime
   FROM GameSessionStart
   GROUP BY UserName) groupedtt
ON UD.UserName = groupedtt.UserName
```

7) Someone from the Publishing team wants to know the country composition of the player base in the first week of March 2020. Write a SQL query to get them the result!

**Answer:**

```
Select datepart(week, '2020-03-01'), distinct UD.country
From GameSessionStart GS
JOIN UserDetail UD
ON GS.UserName = UD.UserName
Where datepart(week, '2020-03-01') <= 1
And GS.LoginDate between '2020-03-01' and '2020-03-07'
Group by datepart(week, '2020-03-01'), UD.country
```

8) A marketing manager claims that beta participants tend to be more active after release than users who did not play during the beta phase. How would you check that?

**Answer:**

**Assumption:** I am assuming we want to see number of logins by Beta(1) vs nonBeta (0)Participants. We group by each group and see count of LoginDate to see who is higher (being more active). The counts may prove or disapprove the marketing manager's claim.

```
Select UD.BetaParticipant, count(GS.LoginDate)
From GameSessionStart GS
JOIN UserDetail UD
ON GS.UserName = UD.UserName
Group by UD.BetaParticipant
```

# Telemetry Design: WheelsCO

**Data Points needed:**

1) Vehicle Type
2) # of Passengers
3) Rental Date
4) Return Date
5) # of Days rented (#4 – #3)
6) Rental Rate Charged Per Day
7) Total Rate Charged (#5 * #6)
8) Maintenance Cost when refurbished
9) Distance Traveled by renter
10) Location Type (Airport, City, Country etc)

**KPI's:**

**Lifetime Value of customer by Vehicle Type:** We want to see how much we are getting over the lifetime of the customer by vehicle type.

**Income to Maintenance Ratio:** By vehicle type, we see how much each type cost us.

**WashOut Percentage:** Total rental income for the asset type and subtract the original purchase price, any carrying cost and maintenance expense over its lifetime.

**Financial Utilization:** Total Revenue generated by each asset type / Cost of Acquiring these assets

**Time Available and Time Utilized: How** many days the vehicle in our fleet was available for rent and how many days they were actually under contract by customers.

**Revenue Growth Over time by Vehicle Type:** We look at the growth rate by looking at previous time period to see the trajectory of Supply/Demand

**Relationship between RPD & Utilization:** Pricing and utilization KPI's

**Residual Value :** How much are we getting when we sell our vehicles on used market

**Average Fleet by Vehicle Type vs. Rental Days by Vehicle Type:** Calculate these KPI's

**Cost of Debt:** How much is costing to borrow funds

**Whom would I talk to:** I would talk to finance & operations folks to understand cost & utilization of our assets.

**Output needed for reasonable analysis:**

1) Revenue by vehicle type
2) Cost by vehicle type
3) Growth Rate by vehicle type
4) Profitability by Vehicle Type

Use above data/KPI's to answer more questions

**Tracking Events:** Finance Data Points (purchase vehicles etc.), Operations data points

## SQL statements for KPI

**Lifetime Value by Vehicle Type:**

```
Select Vehicle_type, sum(total_customer_cost) / count(total_customer_cost) as
Avg_lifetime_value
From Transaction
Group by Vehicle_type
```

**Return on Original Cost**

```
Select Vehicle_Type, sum(Total_customer_cost) / V.Original_cost_vehicle as Return_On_Cost
From Transaction T
JOIN Vehicle V
ON T.vehicleID = V.vehicleID
Group by Vehicle_Type
```