FREENOW CASE STUDY
For : Product analyst -B2B

Given Data:

Freenow_Business_Customer.csv
Freenow_Business_Customer_Activity.csv
Freenow_Business_Customer_employees.csv

Tools Required for analysis:

Online SQLite editor
Google Spreadsheets

Project Repository:

Temporary Github repository location: https://github.com/Akash-Apturkar/Freenow-Task-temp
Note: since the repository is public, only the queries are added and no data is published

Task 1: How do you define the business customer' lifecycle based on the provided data? Clearly

define the stages and relevant KPIs to measure the customer lifecycle

Based on the given data, the three tables can be joined using the *id_business* field and the basic customer life cycle can be mapped with the key KPIs as follows:

Step 1: *signup_date* is the first step in the life cycle of the customer and is obtained from the Freenow_Business_Customer table.

Step 2: first_employee_invite_date , the second step in the customer's lifecycle is the date when the first employee was invited.
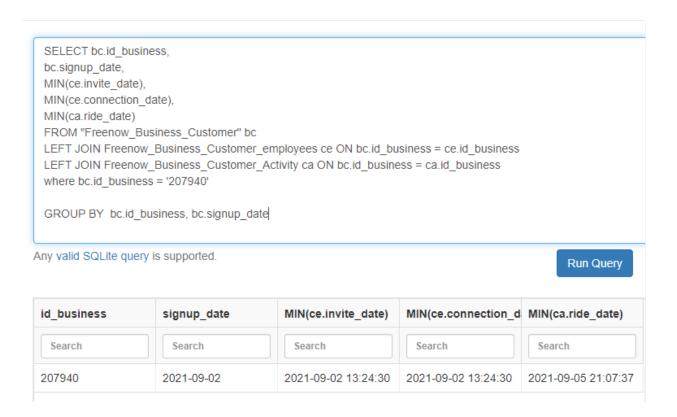
Step 3: first_employee_connect_date , the third step in the customer's lifecycle is the date when the first employee was connected.

Step 4: Date of first ride, gives the date of the first ride by the given business

Step 5: Date Milestone achieved, For the fifth step, we can define a business relevant custom milestone to measure. This can be a certain number of employees connected or rides achieved or a certain amount of revenue reached. For our example, I will assume the ride_value is in Euros and I will set a milestone to the date when a cumulative ride_value of 100 EUR was achieved by the customer.

Step 5: The final KPI that can be measured in a customer's life cycle can be the offboarding date, however the given data does not contain this field.

For Example, I selected a random business id () and used the following query to get the first 4 KPIs

```
SELECT bc.id_business,
bc.signup_date,
MIN(ce.invite_date),
MIN(ce.connection_date),
MIN(ca.ride_date)
FROM "Freenow_Business_Customer" bc
LEFT JOIN Freenow_Business_Customer_employees ce ON bc.id_business = ce.id_business
LEFT JOIN Freenow_Business_Customer_Activity ca ON bc.id_business = ca.id_business
where bc.id_business = '207940'

GROUP BY  bc.id_business, bc.signup_date
```

Any valid SQLite query is supported.   Run Query

id_business	signup_date	MIN(ce.invite_date)	MIN(ce.connection_d	MIN(ca.ride_date)
Search	Search	Search	Search	Search
207940	2021-09-02	2021-09-02 13:24:30	2021-09-02 13:24:30	2021-09-05 21:07:37

For the 4th KPI in the customer lifecycle, I have written a common table expression to find the cumulative sum of ride values to see at which date it crossed 100. (note: I have assumed the

id_ride is incremental):

```
WITH cte_1 AS (
SELECT  a1.id_ride, a1.ride_value, SUM(a2.ride_value) as cum_sum

FROM (SELECT * FROM Freenow_Business_Customer_Activity where id_business = '207940')  a1
inner join  (SELECT * FROM Freenow_Business_Customer_Activity where id_business = '207940') a2
ON a1.id_ride >= a2.id_ride

GROUP BY a1.id_ride,a1.ride_value
order by a1.id_ride
)
SELECT id_business,id_ride, ride_date AS date_100EUR_crossed
FROM Freenow_Business_Customer_Activity
where id_ride = (SELECT min(id_ride) FROM cte_1 WHERE cum_sum > 100 )
```

Any valid SQLite query is supported.

[Run Query]

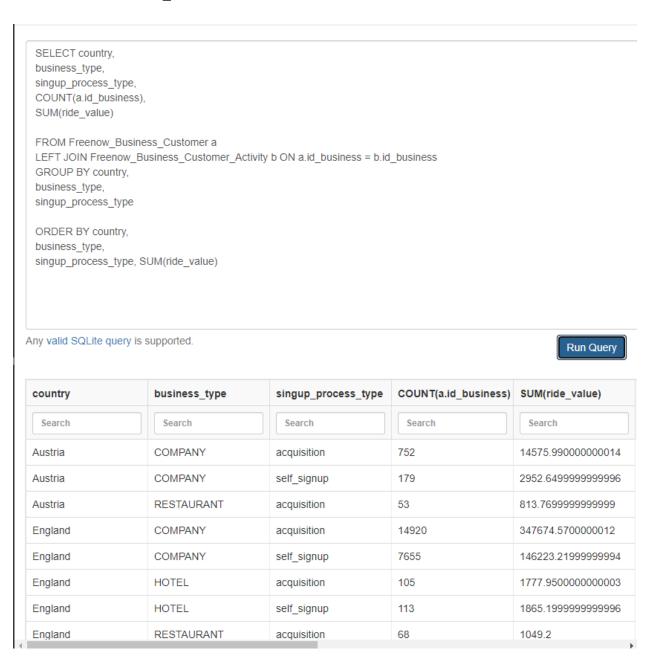| id_business	id_ride	date_100EUR_crossed
Search	Search	Search
207940	326601258	2021-09-13 12:11:33

Task 2: How would you segment our business customers and what segments are doing well?

Looking at the given data, the customers can be segmented in the following ways:

1. Based on location
2. Based on number of employees connected
3. Based on business type
4. Based on signup process
5. Days since first signup

Using the matplotlib library in Python, we can plot the graph of one of these methods against the number of customers. If we see clear clusters forming in such a graph that is the best metric for segmentation.

To find which segments are doing well, we will group the customers in a combination of clusters and use the total of ride_value as a metric to determine success.

```
SELECT country,
business_type,
singup_process_type,
COUNT(a.id_business),
SUM(ride_value)

FROM Freenow_Business_Customer a
LEFT JOIN Freenow_Business_Customer_Activity b ON a.id_business = b.id_business
GROUP BY country,
business_type,
singup_process_type

ORDER BY country,
business_type,
singup_process_type, SUM(ride_value)
```

Any valid SQLite query is supported.   [Run Query]

country	business_type	singup_process_type	COUNT(a.id_business)	SUM(ride_value)
Search	Search	Search	Search	Search
Austria	COMPANY	acquisition	752	14575.990000000014
Austria	COMPANY	self_signup	179	2952.6499999999996
Austria	RESTAURANT	acquisition	53	813.7699999999999
England	COMPANY	acquisition	14920	347674.5700000012
England	COMPANY	self_signup	7655	146223.21999999994
England	HOTEL	acquisition	105	1777.9500000000003
England	HOTEL	self_signup	113	1865.1999999999996
England	RESTAURANT	acquisition	68	1049.2

Using the above query, we can analyze the best performing segments for each country separately. Looking at the results it is abandonedly clear that the customers having business types as 'COMPANY' and which are acquisitions are the highest grocessing in each country.

A visualization tool, such as Tableau, can be used to portray this more clearly.

Task 3: Can you help us to identify the stage in which we have to fight for customer' attention? Can

you recommend any business initiatives or product changes that could help us to have

customer life cycle progression?

For this analysis, I plan to use the customer life cycle as mapped in task 1. During this life cycle, I will calculate the average days required for the business to first invite and then first connect the customer, over different types of employee onboarding methods for different types of customers. This will be calculated from the sign up date. Then we will check the difference between the average days to connect and average days to invite.

```
WITH cte_1 AS (
SELECT bc.id_business,
bc.signup_date,
business_type,
method,
MIN(date(ce.invite_date)) AS first_invite,
MIN(date(ce.connection_date)) AS first_connect
FROM "Freenow_Business_Customer" bc
LEFT JOIN Freenow_Business_Customer_employees ce ON bc.id_business = ce.id_business


GROUP BY bc.id_business, bc.signup_date,business_type,method
)
SELECT business_type, method,
avg(julianday(first_invite) - julianday(signup_date)) AS avg_days_to_invite,
avg( julianday(first_connect) - julianday(signup_date)) AS avg_days_to_connect ,
avg( julianday(first_connect) - julianday(signup_date)) - avg(julianday(first_invite) - julianday(signup_date))  as difference
FROM cte_1 GROUP BY business_type, method
```

Any valid SQLite query is supported.   [Run Query]

business_type	method	avg_days_to_invite	avg_days_to_connect	difference
Search	Search	Search	Search	Search
COMPANY	invite	17.85323383084577	21.049751243781095	3.196517412935325
COMPANY	manual	2.018158047646717	2.018158047646717	0
HOTEL	invite	52.357142857142854	53.642857142857146	1.2857142857142918
HOTEL	manual	28.46875	28.46875	0
RESTAURANT	invite	4	12.166666666666666	8.166666666666666
RESTAURANT	manual	45.5	45.5	0

The above analysis gives us an orientation of where the delays are taking place in the customer timeline .

From the analysis, we can see that:
- If we look at the difference field, where business type is a restaurant, they are taking more days to connect their employees after inviting them. Here, one thing that can be done is the company can deploy a survey to the invited employees of the business to find out the reason for the delay to connect.
- If we look at the average days to invite for Hotels and restaurants is very high. Here, the company is clearly struggling to get the customer's attention.
  One thing that can be improved with the product, is to set up automated reminders for invites.
  The other thing that can be improved is the product can recommend employees in the same location already with Freenow to the customer to invite.