Flyber Data Strategy MVP

Introduction

Flyber has been massively successful. Results have beaten expectations and projections! This is good news for Flyber, but now it’s time to plan for what's next. With success came some challenges. While we were able to grow, the original data pipelines to receive and process data are unable to keep up with the current and future growth.

As a Data Product Manager, working with multiple teams and stakeholders is imperative to success. To understand what our needs are, what scale we are growing at, and how we can build for the future, we need to consider all relevant stakeholders. In this proposal, present your findings along with the analysis and reasoning behind the choices made in order to help Flyber continue its success.

Section 1: Data Customers & Needs

Flyber is a two-sided platform. You have customers who are riders, and you have partners who are drivers/pilots (think Uber: riders and drivers). For the Minimum Viable Product, you will be focusing on the Riders side of the business. To build an end to end data pipeline the very first step is to understand who needs data and why they need that data. Within Flyber, identify who your primary data customers/stakeholders are, why they are your primary data stakeholders and how they want to use the data (primary use-cases).

Identify your primary internal stakeholders and their use-cases:
(You may add more rows if necessary.)

Stakeholder	**Why are they primary stakeholders?**	**Use-Case**
Engineering	having a working product	monitor flyber service and the app performance
Safety department	having a safe and secure product	monitor flying logs, engine logs
Marketing	getting new customers	targeted advertising
Customer Service department	retaining current users	provide tailored solutions to users

Section 2: Data Collection and Data Modelling

To support our primary stakeholders's use-cases we need following data:
(You may add more rows if necessary.)

Stakeholder	**Use-Case**	**Data**	**Why is this the primary use-case?**
Engineering	monitor flyber service	type: Event Data element: Event ID, timestamp, Event type	Develop and monitor functional hardware and software solution
Safety department	monitor flying logs, engine logs	type: Event Data element: Event ID, timestamp, Event type	Comply safety and security regulation, ensure safe travel
Marketing	targeted advertising	type: Entity Data element: user name, email, phone, address, user order history	Generate new users
Customer service	provide tailored solutions to users	type: Entity Data element: Time stamp when ticket was placed, user id, email, ticket history	keep user, minimize churn rate

The tables we need are:

Note: As a best practice, we should establish these relationships between tables from the very beginning. To complete this exercise we will focus on fundamental concepts of relational databases - tables, normalization and unique keys. Please provide the table header row for each table, tables might be different lengths. Make sure you include the following for each table. You can create as many tables as you feel are necessary (copy and paste from one of the table sections):

Table 1:

User

(You may add more columns if necessary.)

User ID (PK)	*Flyber_cab ID (FK)*	*First Name*	*Last Name*

Rationale for Choosing Primary and Foreign Keys for the Table 1:

Everything starts with user and flyber cab: we need users in our flying service and we need flyber flying cab..

Table 2:

Flyber_cab

(You may add more columns if necessary.)

Flyber_cab ID (PK)	*Ride ID (FK)*	*license plate*	*active*

Rationale for Choosing Primary and Foreign Keys for the Table 2:

Flying service another key element is the flying cab, the flying cab ID is the PK and to link another table is essential to add the rides that is flying service transaction.

Table 3:

Ride

Ride ID (PK)	*User ID (FK)*	*Flyber_cab (FK)*	*Ride Date*	*Pick-up Date*	*Drop-off Date*

Rationale for Choosing Primary and Foreign Keys for the Table 3:

The Ride is the event with user and cab which is the main event for the service.

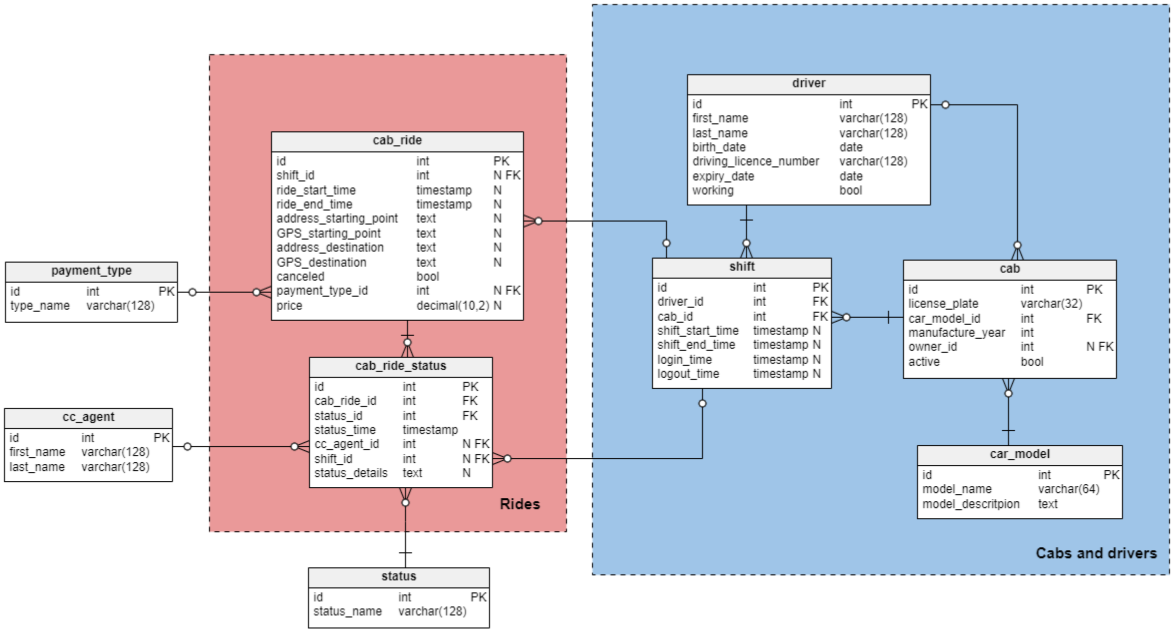
Another important table is the Ride table as need to analyze the duration, distance

Supplement materials:

https://www.vertabelo.com/blog/ [Creating Data models]

https://www.vertabelo.com/blog/a-saas-subscription-data-model/

Cab company data model:



source: https://my.vertabelo.com/model/QeenxMOaGHarJa5feJMcS4NANytJ033l

Section 3: Extraction and Transformation

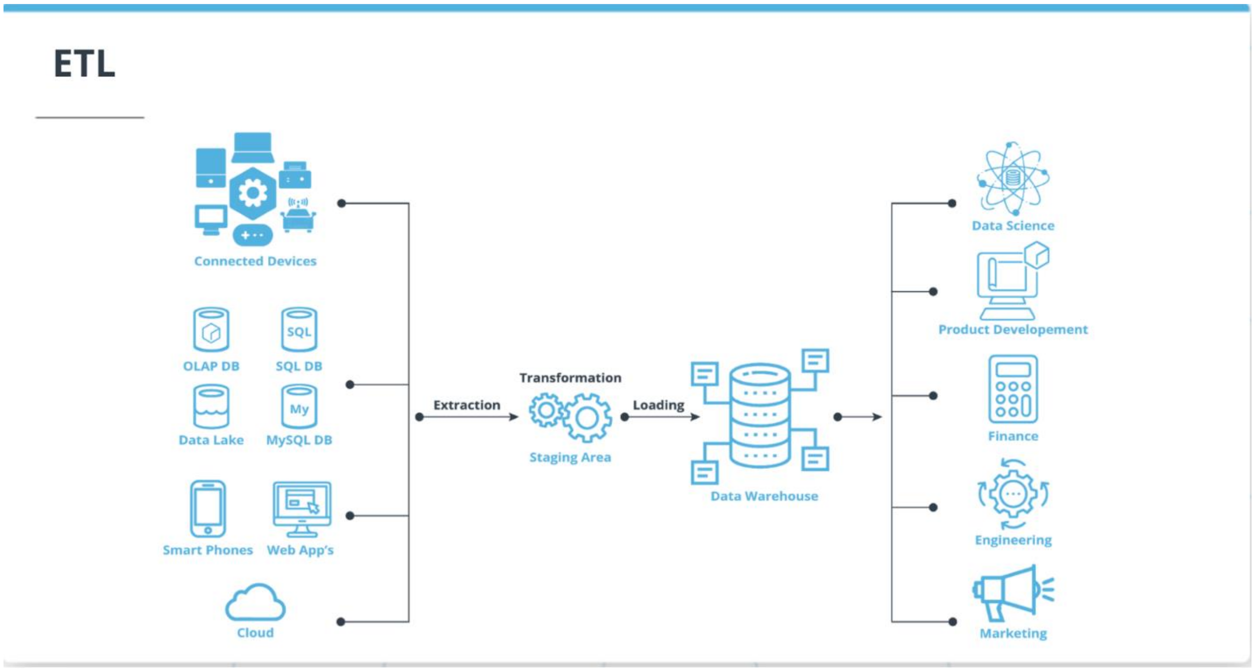
Now that you have the requirements from your stakeholders, you want to understand the current state of what data is collected. That is how you recognize **which additional data you need to achieve the future state**. You ask the engineering team what data they are currently collecting in the pipelines and they provide you with section_3_event_logs template (which you can download from the classroom) generated by rider's activities on the Flyber App. Also provided in the Project Resources.

Extraction and Transformation-1

ETL is performed on the provided Event Logs Template and results will be transferred to the proposal template. The project's ETL should be created inside of your copy of the Event Logs template in the tab titled, ETL. Clicking on the link above will create a copy of the Event Logs for you

After being provided with a CSV log file, use extraction techniques to be able to get the data into a usable form. Because this needs to be a repeatable process we need to document it in order to assess its feasibility. Below,

1. Write the steps you took to extract the data and provide reasoning for why you used this method *Note: Don't forget to include any file type changes*:
2. Perform cleaning and transformation of the data in the ETL tab and document.
3. Document and provide rationale for all of your steps below as well.



ETL

1. ***Extract | Data Collection***

This step comprises data extraction from the source system into the staging area. Any transformations can be done in the staging area without degrading the performance of the source system.

Tasks: Verifying records for spam or unwanted data, checking data type, Removal of fragmented/duplicate data, Checking the placement of keys

2. ***Transform***

The data that is extracted from the source server is incomplete and not usable in its original form. Because of this, you need to cleanse, map, and transform it.

Tasks: Filtering – To select only specific columns for loading, Data standardization using lookup tables and rules, encoding handling and character set conversion, Conversion of measuring units like currency, numerical, and date/time conversions, Checking data threshold validation

3. ***Load***

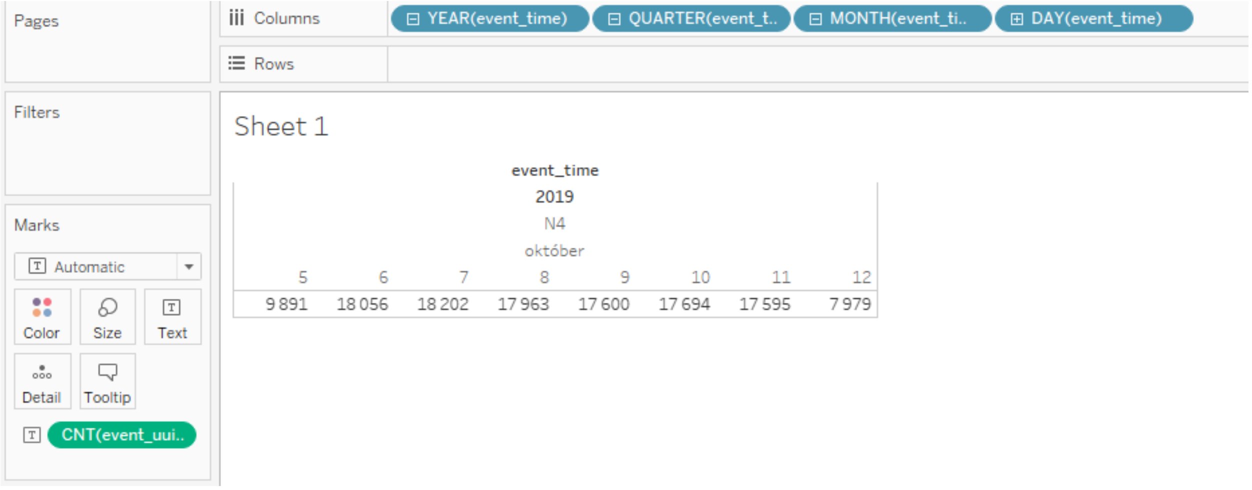
The last step of the ETL process includes loading data into the target database of the data warehouse. In a standard data warehouse, large volumes of data have to be loaded in a comparatively short period. As a result, the loading process needs to be streamlined for performance.

Transformation-2

Analyze the data from part 1 to answer the following questions:

1. How many events are being recorded per day?

Date	10/5/2019	10/6/2019	10/7/2019	10/8/2019	10/9/2019	10/10/2019	10/11/2019
Event Count	9891	18056	18202	17963	17600	17694	17595



Columns: YEAR(event_time) QUARTER(event_t.. MONTH(event_ti.. DAY(event_time)

Rows

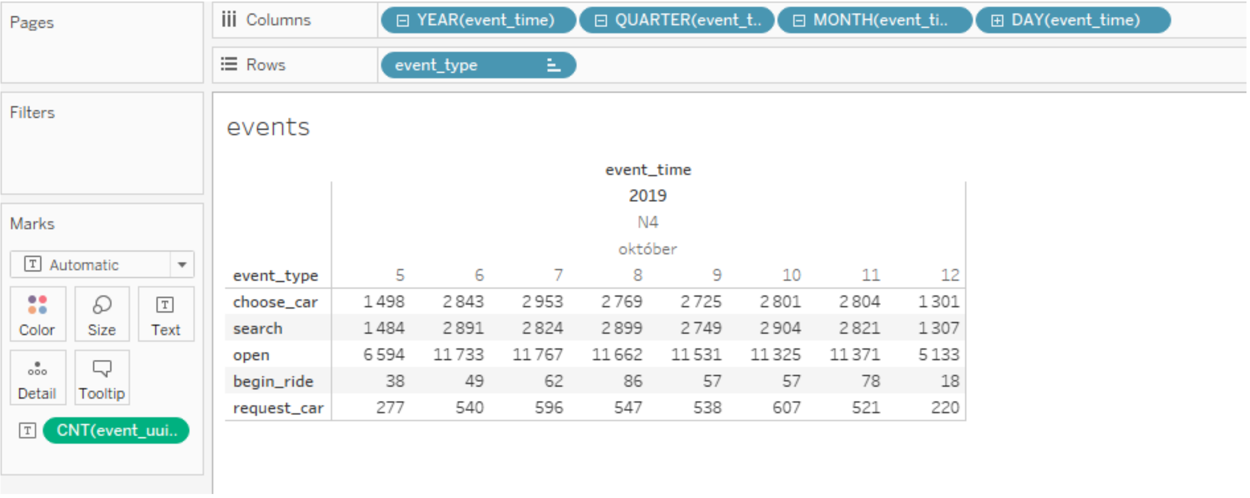
Marks: CNT(event_uui..

Sheet 1

event_time							
2019							
N4							
október							
5	6	7	8	9	10	11	12
9 891	18 056	18 202	17 963	17 600	17 694	17 595	7 979

2. How many events of each event type per day?

Date	10/5/2019	10/6/2019	10/7/2019	10/8/2019	10/9/2019	10/10/2019	10/11/2019
Choose Car	1498	2843	2953	2769	2725	2801	2804
Search	1484	2891	2824	2899	2749	2904	2821
Open	6594	11733	11767	11662	11531	11325	11371
Begin Ride	38	49	62	86	57	57	78
Request Car	277	540	596	547	538	607	521



Columns: YEAR(event_time) | QUARTER(event_t.. | MONTH(event_ti.. | DAY(event_time)

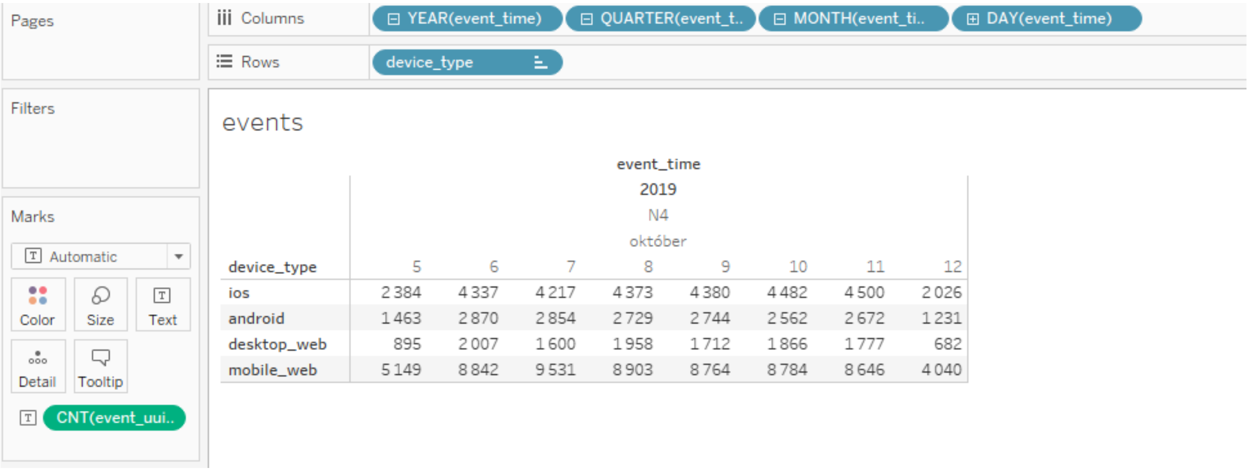
Rows: event_type

events

	event_time							
	2019							
	N4							
	október							
event_type	5	6	7	8	9	10	11	12
choose_car	1 498	2 843	2 953	2 769	2 725	2 801	2 804	1 301
search	1 484	2 891	2 824	2 899	2 749	2 904	2 821	1 307
open	6 594	11 733	11 767	11 662	11 531	11 325	11 371	5 133
begin_ride	38	49	62	86	57	57	78	18
request_car	277	540	596	547	538	607	521	220

3. How many events per device type per day?

Date	10/5/2019	10/6/2019	10/7/2019	10/8/2019	10/9/2019	10/10/2019	10/11/2019
ios	2384	4337	4217	4373	4380	4482	4500
android	1463	2870	2854	2729	2744	2562	2672
Desktop Web	895	2007	1600	1958	1712	1866	1777
Mobile Web	5149	8842	9531	8903	8764	8784	8646



Columns: YEAR(event_time) | QUARTER(event_t.. | MONTH(event_ti.. | DAY(event_time)

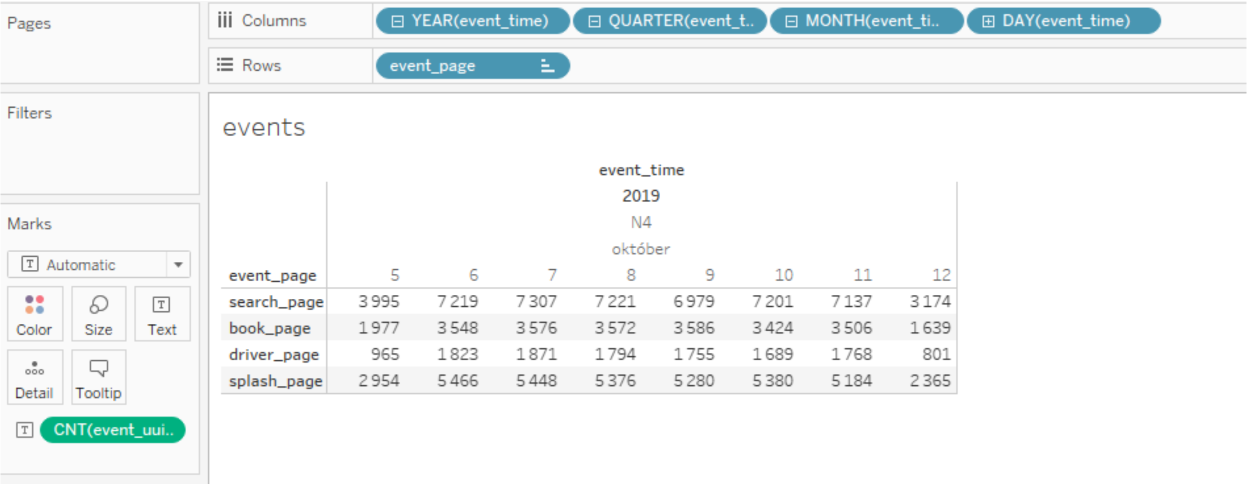
Rows: device_type

events

	event_time							
	2019							
	N4							
	október							
device_type	5	6	7	8	9	10	11	12
ios	2 384	4 337	4 217	4 373	4 380	4 482	4 500	2 026
android	1 463	2 870	2 854	2 729	2 744	2 562	2 672	1 231
desktop_web	895	2 007	1 600	1 958	1 712	1 866	1 777	682
mobile_web	5 149	8 842	9 531	8 903	8 764	8 784	8 646	4 040

4. How many events per page type per day?

Date	10/5/2019	10/6/2019	10/7/2019	10/8/2019	10/9/2019	10/10/2019	10/11/2019
Search Page	3995	7219	7307	7221	6979	7201	7137
Book Page	1977	3548	3576	3572	3586	3424	3506
Driver Page	965	1823	1871	1794	1755	1689	1768
Splash Page	2954	5466	5448	5376	5280	5380	5184



Columns: ⊟ YEAR(event_time) ⊟ QUARTER(event_t.. ⊟ MONTH(event_ti.. ⊞ DAY(event_time)

Rows: event_page

Marks: Automatic

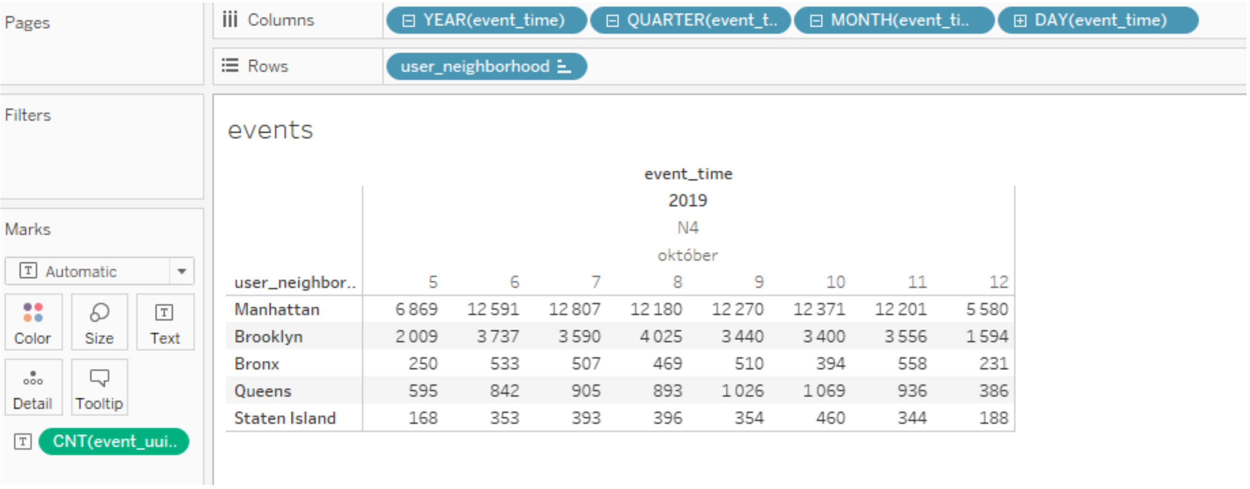
CNT(event_uui..

events

	event_time							
	2019							
	N4							
	október							
event_page	5	6	7	8	9	10	11	12
search_page	3 995	7 219	7 307	7 221	6 979	7 201	7 137	3 174
book_page	1 977	3 548	3 576	3 572	3 586	3 424	3 506	1 639
driver_page	965	1 823	1 871	1 794	1 755	1 689	1 768	801
splash_page	2 954	5 466	5 448	5 376	5 280	5 380	5 184	2 365

5. How many events for each location per day?

Date	10/5/2019	10/6/2019	10/7/2019	10/8/2019	10/9/2019	10/10/2019	10/11/2019
Manhattan	6869	12591	12807	12180	12270	12371	12201
Brooklyn	2009	3737	3590	4025	3440	3400	3556
Bronx	250	533	507	469	510	394	558
Queens	595	842	905	893	1026	1069	936
Staten Island	168	353	393	396	354	460	344



events

	event_time 2019 N4 október							
user_neighbor..	5	6	7	8	9	10	11	12
Manhattan	6 869	12 591	12 807	12 180	12 270	12 371	12 201	5 580
Brooklyn	2 009	3 737	3 590	4 025	3 440	3 400	3 556	1 594
Bronx	250	533	507	469	510	394	558	231
Queens	595	842	905	893	1 026	1 069	936	386
Staten Island	168	353	393	396	354	460	344	188

ETL Automation and Scalability:

Provide an analysis about this ETL process. Address and provide rationale for manually extracting, loading and transforming the data from the raw logs. Also address potential preliminary recommendations on improving this process.

- *The advantage of manual processing is that the product manager knows the quality of the data sources and can develop a process for processing and loading that is suitable for achieving product KPIs and at the same time can improve automation.*
- *Automated ETL process helps in saving time and engineering resources, we can easily handle big data. Provides blueprints anyone in the organization can use.*
- *Creating automated ETL process is data engineering tasks (great examples: Data Engineering Principles - Build frameworks not pipelines - Gatis Seja, Uber's Big Data Platform: 100+ Petabytes with Minute Latency*

Section 4: Choosing Relevant Dataset

The previous exercise gave you a sneak peek into the Extraction and Loading aspects of ETLs in data pipelines. For making business decisions, a data consumer would like to have all the data they want. However, for any ecosystem, it is impossible to collect or provide everything that the customers need. In this exercise, you will get a taste of real-world scenarios wherein:

- All the resources are not always available to get what you need.
- You have to get creative and get the most insights with a minimal data set.

Oftentimes your stakeholders/customers will "ask for the moon", but you'll have to push them to work with the small amount of information you have and get creative.

Note: As you learned in the course, being a Data Product Manager involves an extraordinary amount of collaboration. Complete the next sections based on the following scenario.

After the analysis in section 3, we made sense of the numbers, and realized the total number of events seems to be too small (this was a week's worth of data, but you need at least a month). Further investigation reveals that this was a subset of logs, but the actual data that is being collected is much bigger. Working through this small data set was tedious, and repeating this exercise on a much bigger data set manually won't be feasible. Considering the time constraints of this project, engineering is willing to help with some automation. They also have limited bandwidth and are busy scaling systems up.

Engineering is willing to provide some data, but they have asked for the criterion that is most important. To First provide your business question and provide a rationale for why this is the most important.

Choose one of the following prompts that you think can get you the most relevant information to proceed further.

1. How many events are being recorded per day?
2. How many events of each event type per day?
3. How many events per device type per day?
4. How many events per page type per day?
5. How many events for each location per day?

For your chosen question also answer the following using the data from section 3 to support your answer:

1. How much is the customer data increasing?
2. How much is the transactional data increasing?
3. How much is the event log data increasing?

Which of the following data is ***most*** important to answer this question? Why?

- **Event Log Data**
- Transactional Data
- Customer Data

Event Log Data

This data contains the different event types that corresponds to the activity of the different users.

It contains data for the different months which is specifically what we need to compare the different activities between months.

Use of Automated ETL which can be done more easily on the Event Log Data. Also, with an event we can have transactions and the customes assist in this event. The event can give more data for the analysis.

Section 5: [Optional] Loading and Visualization On Your Own

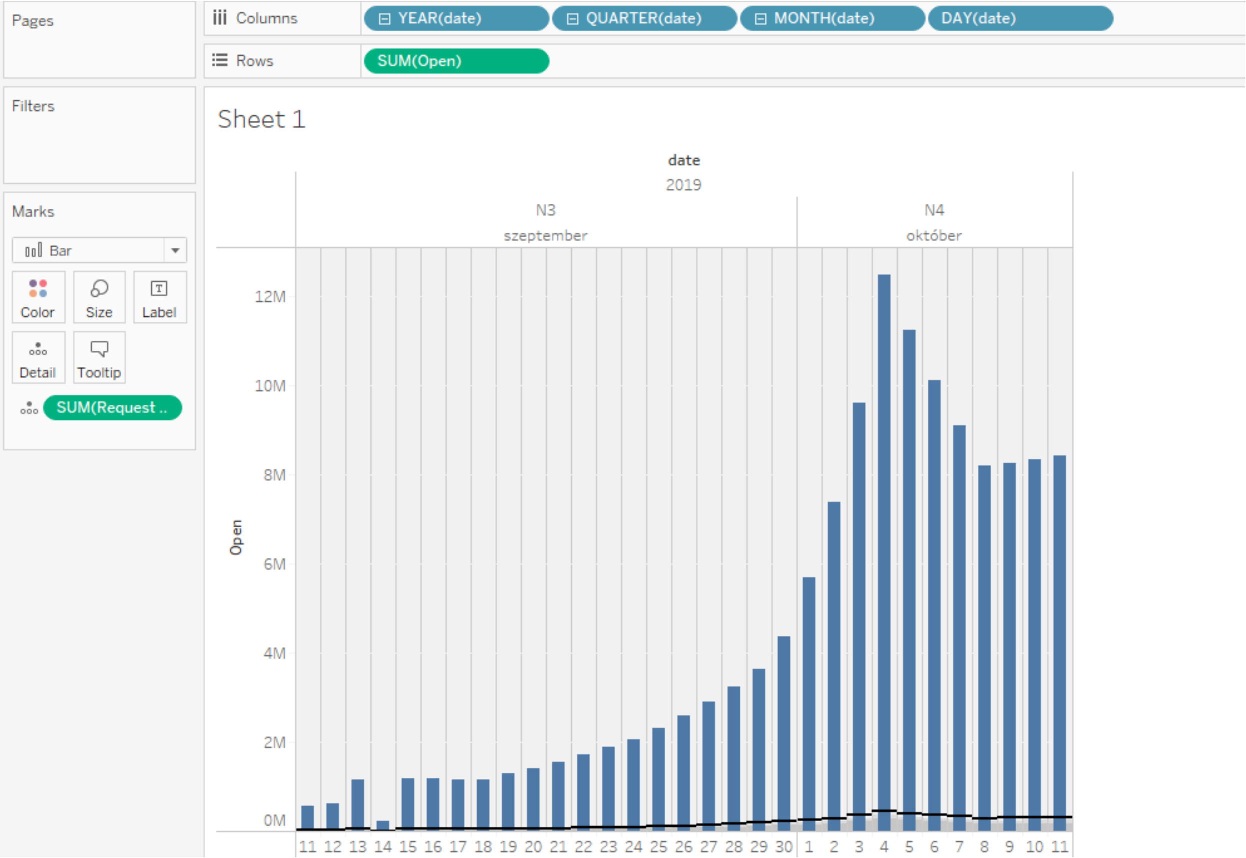
This sectional is an optional part of the project that you can do to make it standout. We have provided visualizations in the appendix if you decide not to do this section. You can also use our visualizations to compare what you created

After sharing your criterion with engineering, they give you a new set of data: Section 5 Event Type Log also available in the classroom resources. Also provided in the project resources section.

Engineering provided you with the data you want, but you still have yet to achieve your ultimate goal as a Data Product Manager. Now, utilize the data to make business decisions. Your executives do not want you to give them a bunch of data tables; instead, they prefer visualizations to help convey the key insights succinctly. Visualizing this data will help you understand the underlying trends and help you determine the story that needs to be told in your proposal to executives.

In this section, you can load and visualize the data into whatever platform you would like. A Python Notebook, Tableau or any other visualization tool you are familiar with. Create two visualizations that might help you to better understand your data trends and place either a screenshot or exported image of your visualizations and the details of each below. Please provide the steps you took to visualize your data and what the visualization tells you about your data.

Visualization 1:



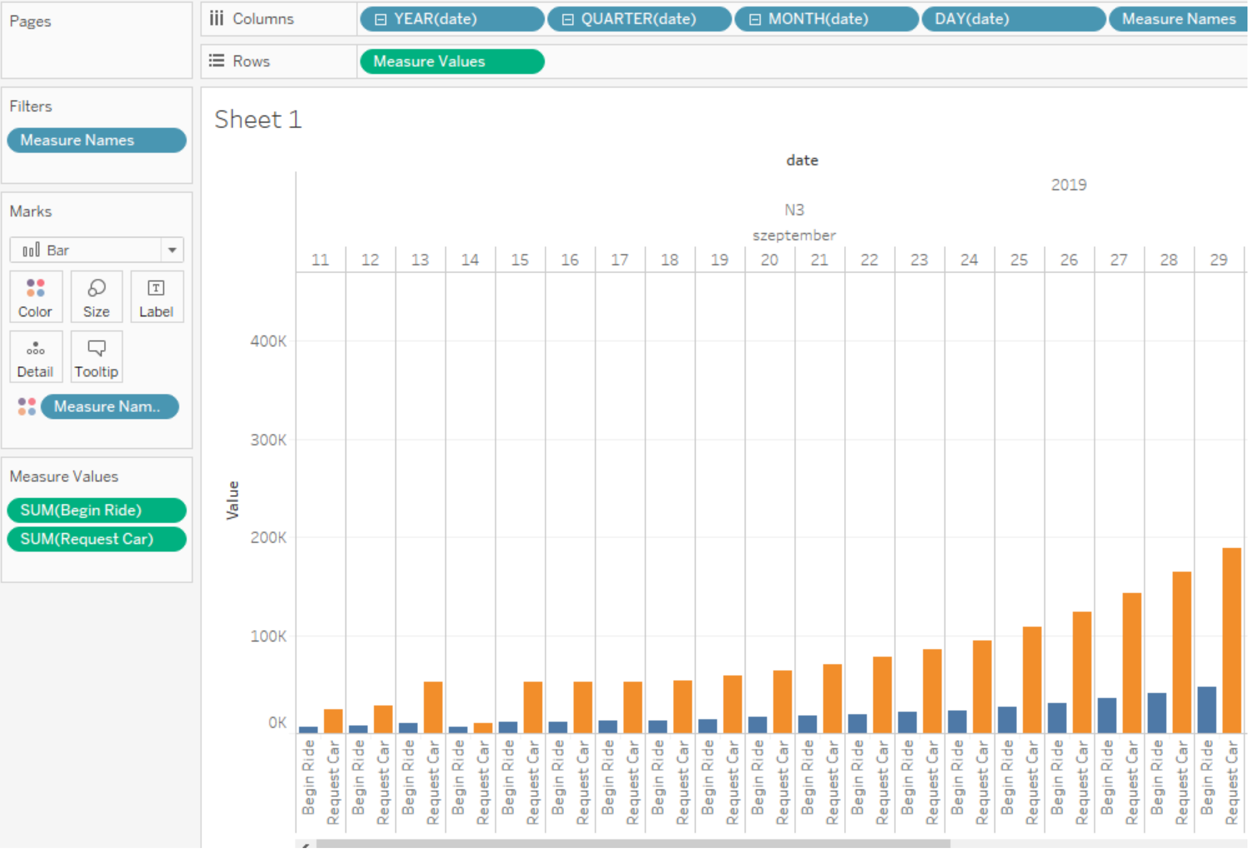
Data Story: This graph tells us:

What is the user behavior between using the app and ordering a car.
Business question: Why is so low the requested car vs user open the app rate? What is the reason?

This graph was created using the following steps:

1. *Load the excel table into Tableau*
2. *Date put into columns*
3. *Open put in row as sum*
4. *Request put into marks*

Visualization 2:



Columns: YEAR(date) QUARTER(date) MONTH(date) DAY(date) Measure Names

Rows: Measure Values

Filters: Measure Names

Marks: Bar

Measure Values: SUM(Begin Ride) SUM(Request Car)

Data Story: This graph tells us:

Ratio between Request a car vs begin drive. The gap is huge between the two events.

This graph was created using the following steps:

1. *Load the excel table into Tableau*
2. *Date put into columns (Day)*
3. *Measure two values SUM (Request Car and Begin Ride)*

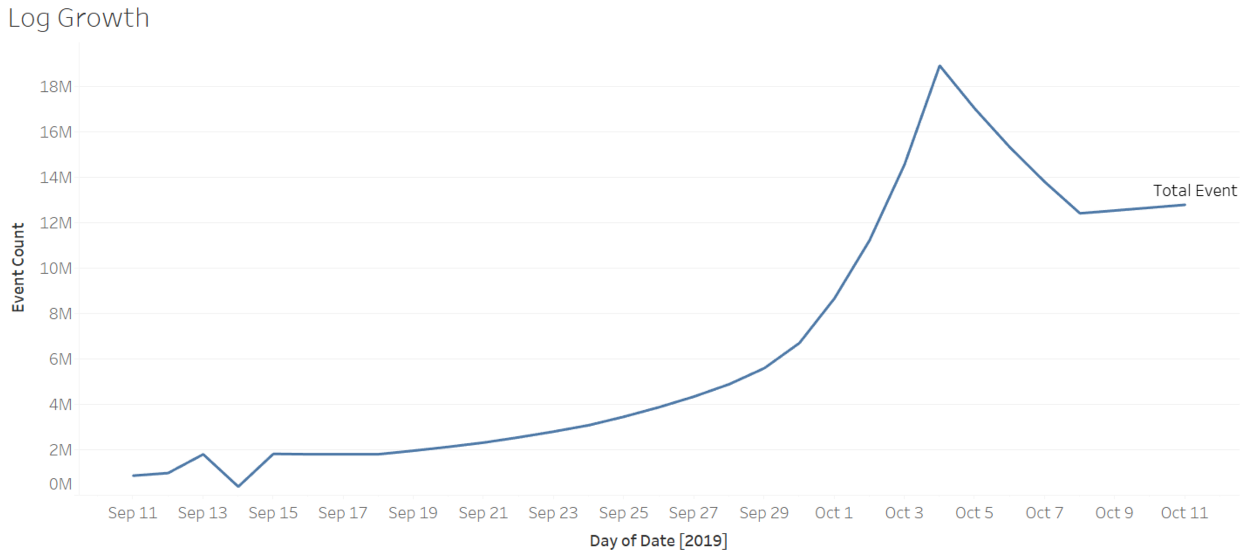
Section 6: Business Insights

The Data is loaded and ready for analysis. We want to use this data as evidence to support our recommendations. It is important that we understand this data and the underlying trends and nuances that these visualizations show us. As you already know, any proposal backed up by data is always better received and considered more robust.

What is the story the data is telling you about Flyber's data growth? If you created Visualizations, you can use them as well, but they are not required). Include any data and calculations that were made to help tell that story and quantify the data growth.

Data Growth for Last Month

Visualization:



Data and calculations used for quantifying of Flyber's Data Growth:

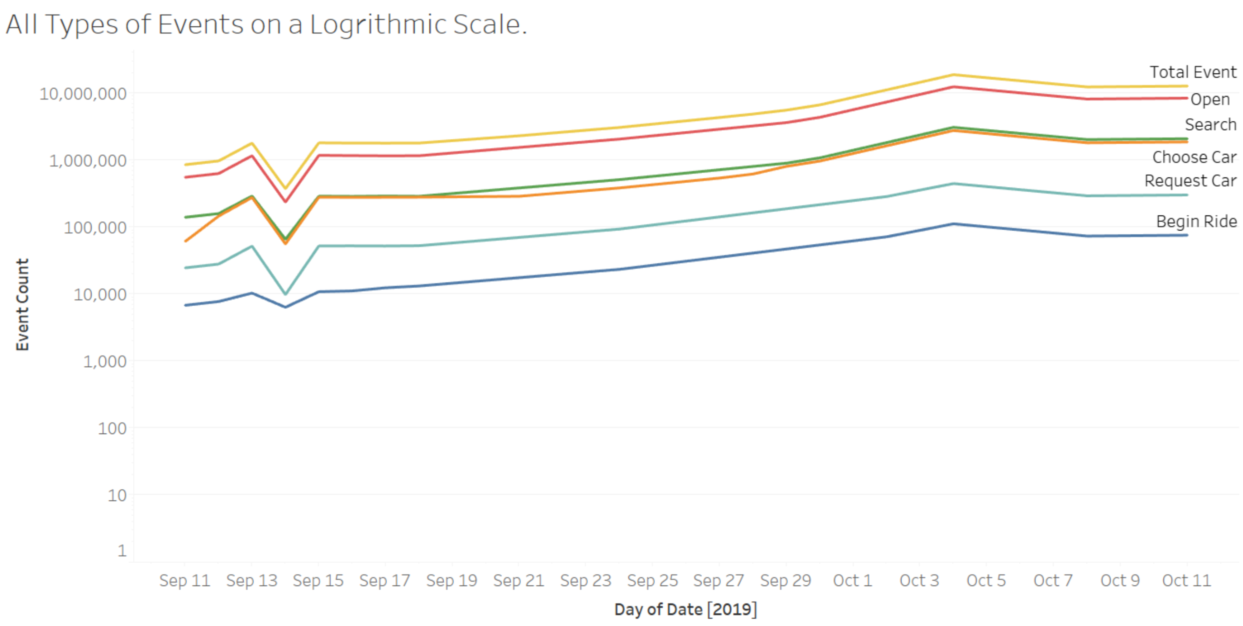
- *Used Log growth (Total Event)*

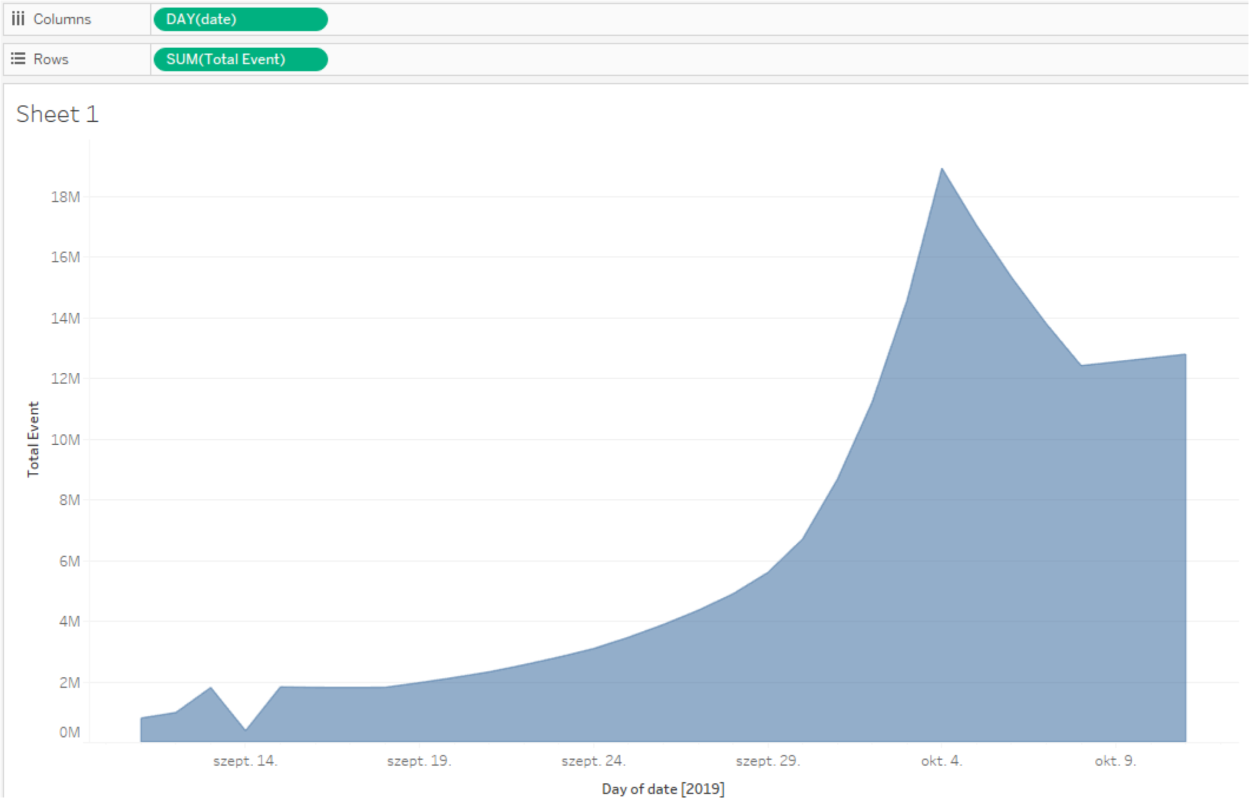
What is the fastest growing data and why?

- Total Events summarize all events

All Event Type Data

Visualization:





Columns DAY(date)

Rows SUM(Total Event)

What is the Data Story our data tells for each of the following:

- Graph Pattern
- Good or Bad
- October Marketing Campaign
- Marketing Campaign Impact
- Importance of Relationship Between Marketing Campaigns and Data Generation

Growth Pattern: *Beg of October there is a peak in Total Events after a plateau. Growth rate is. If we look on the curve, we can see a 2M events in September vs 12M events in October. Because the data is given for a short period of time, we cannot see trends.*

Good or Bad: *The Marketing activities has a huge impact to drive traffic to the application, but one of the main KPI is begin drive (monetization) event has a low ratio vs total events shows some UX or Value Proposition issues.*

October Marketing Campaign *focused on to drive new users to the application was successful. If this was the KPI it is a good sign.*

Marketing Campaign Impact: *Peak user activity in beg of October.*

Importance of Relationship Between Marketing Campaigns and Data Generation: *The marketing activities drive more traffic to the service so more user data is generated which can generates more data insights.*

Section 7: Data Infrastructure Strategy

Thus far we have:

- identified data stakeholders and their data needs.
- Identified what data is currently being collected and what data needs to be collected.
- Identified data insights and growth trends.

Now, it's time to tie all the loose threads together and bring this process to its logical conclusion by suggesting which Data Warehouse (DWH) Flyber should invest in and why. Using data warehouse options below, suggest whether Flyber should choose an on-premise or Cloud data warehouse system and which specific data warehouse would best serve Flyber's data needs.

Data Warehouse Options:

Cloud:

- Amazon Redshift
- Google BigQuery
- Snowflake
- Microsoft Azure

On-Premise:

- Oracle Exadata
- Teradata, Vertica
- Apache
- Hadoop

You will address the following factors with a rationale as to why the DWH chosen is the best for Flyber:

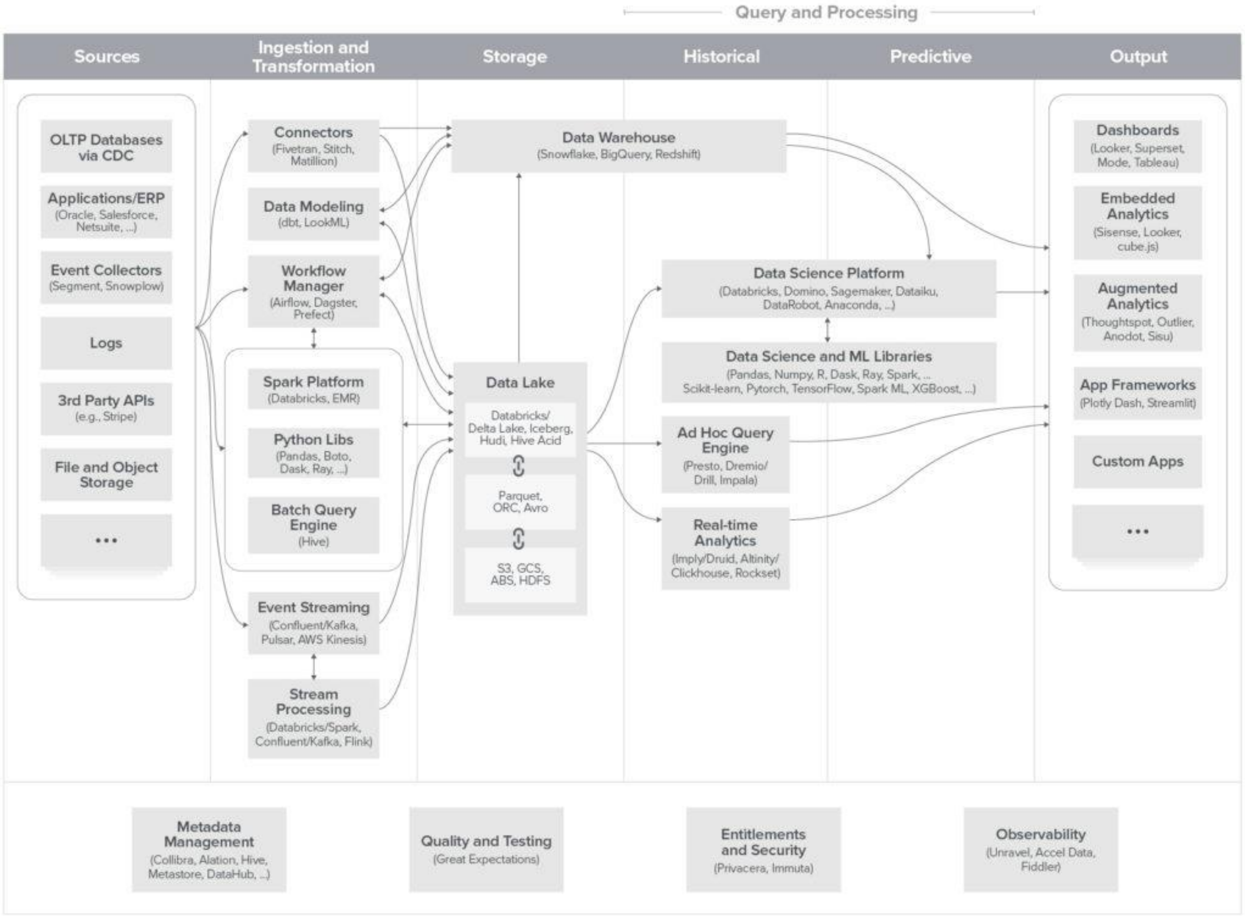
- Cost
- Scalability
- In-house Expertise
- Latency/Connectivity
- Reliability

Cloud vs On-Premise

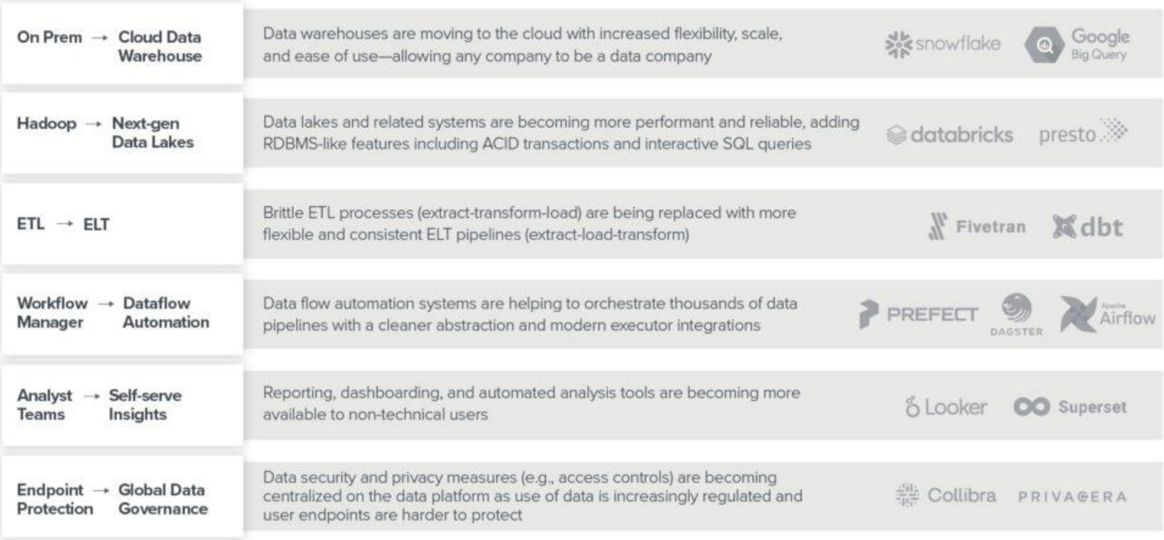
Provide an evidence-based solution as to why Flyber would be best served by a Cloud or on-premise DWH. In this response, you don't need to specify *which* specific Cloud or on-premise DWH product you will choose, just if it will be Cloud or on-premise. Remember to address the factors above.

Analysis: Cloud vs On-Premise

modern Data Architecture (source: [Emerging Architectures for Modern Data Infrastructure](#))

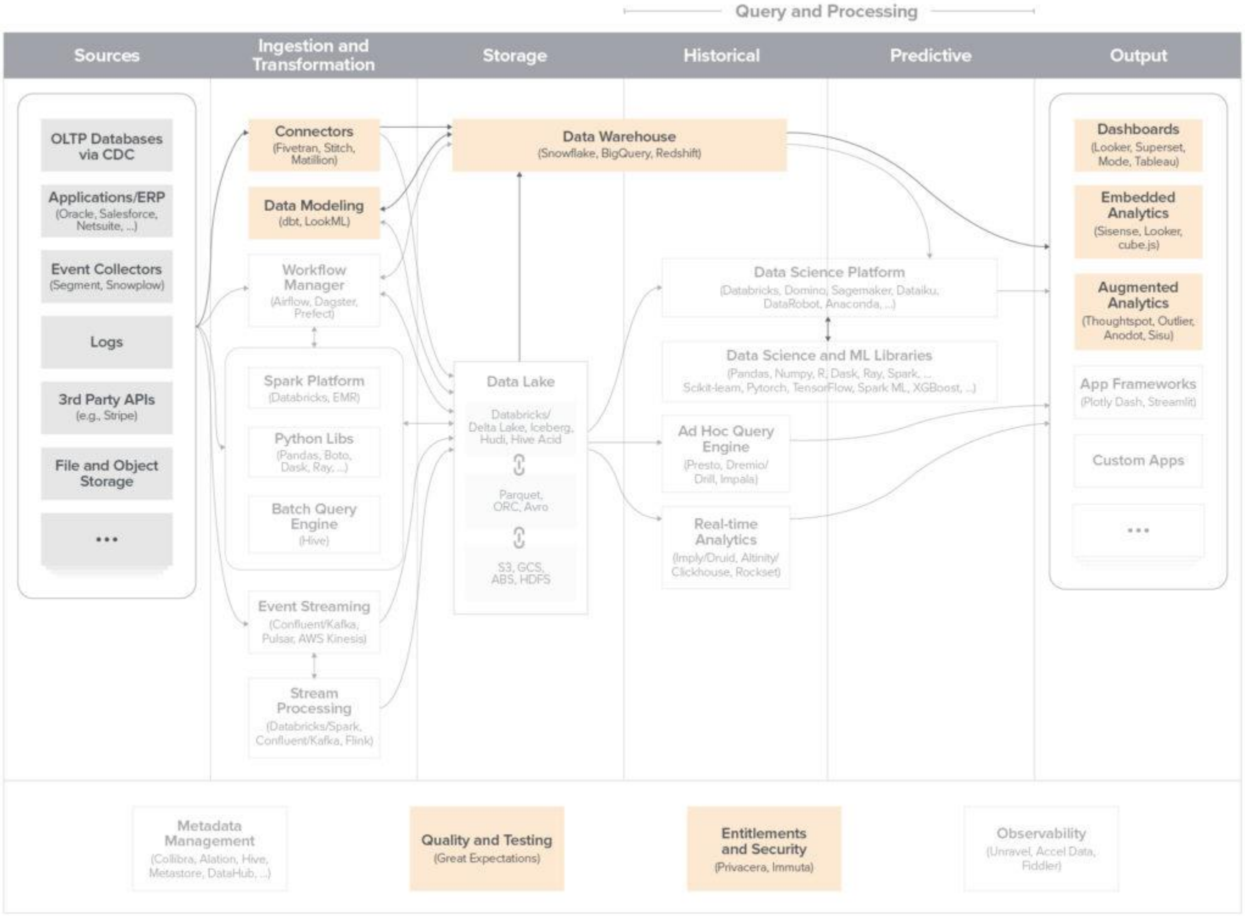

Query and Processing
Sources
Ingestion and Transformation
Storage
Historical
Predictive
Output
OLTP Databases via CDC
Applications/ERP (Oracle, Salesforce, Netsuite, ...)
Event Collectors (Segment, Snowplow)
Logs
3rd Party APIs (e.g., Stripe)
File and Object Storage
• • •
Connectors (Fivetran, Stitch, Matillion)
Data Modeling (dbt, LookML)
Workflow Manager (Airflow, Dagster, Prefect)
Spark Platform (Databricks, EMR)
Python Libs (Pandas, Boto, Dask, Ray, ...)
Batch Query Engine (Hive)
Event Streaming (Confluent/Kafka, Pulsar, AWS Kinesis)
Stream Processing (Databricks/Spark, Confluent/Kafka, Flink)
Data Warehouse (Snowflake, BigQuery, Redshift)
Data Lake
Databricks/ Delta Lake, Iceberg, Hudi, Hive Acid
Parquet, ORC, Avro
S3, GCS, ABS, HDFS
Data Science Platform (Databricks, Domino, Sagemaker, Dataiku, DataRobot, Anaconda, ...)
Data Science and ML Libraries (Pandas, Numpy, R, Dask, Ray, Spark, ... Scikit-learn, Pytorch, TensorFlow, Spark ML, XGBoost, ...)
Ad Hoc Query Engine (Presto, Dremio/ Drill, Impala)
Real-time Analytics (Imply/Druid, Altinity/ Clickhouse, Rockset)
Dashboards (Looker, Superset, Mode, Tableau)
Embedded Analytics (Sisense, Looker, cube.js)
Augmented Analytics (Thoughtspot, Outlier, Anodot, Sisu)
App Frameworks (Plotly Dash, Streamlit)
Custom Apps
• • •
Metadata Management (Collibra, Alation, Hive, Metastore, DataHub, ...)
Quality and Testing (Great Expectations)
Entitlements and Security (Privacera, Immuta)
Observability (Unravel, Accel Data, Fiddler)

Architectural Shifts



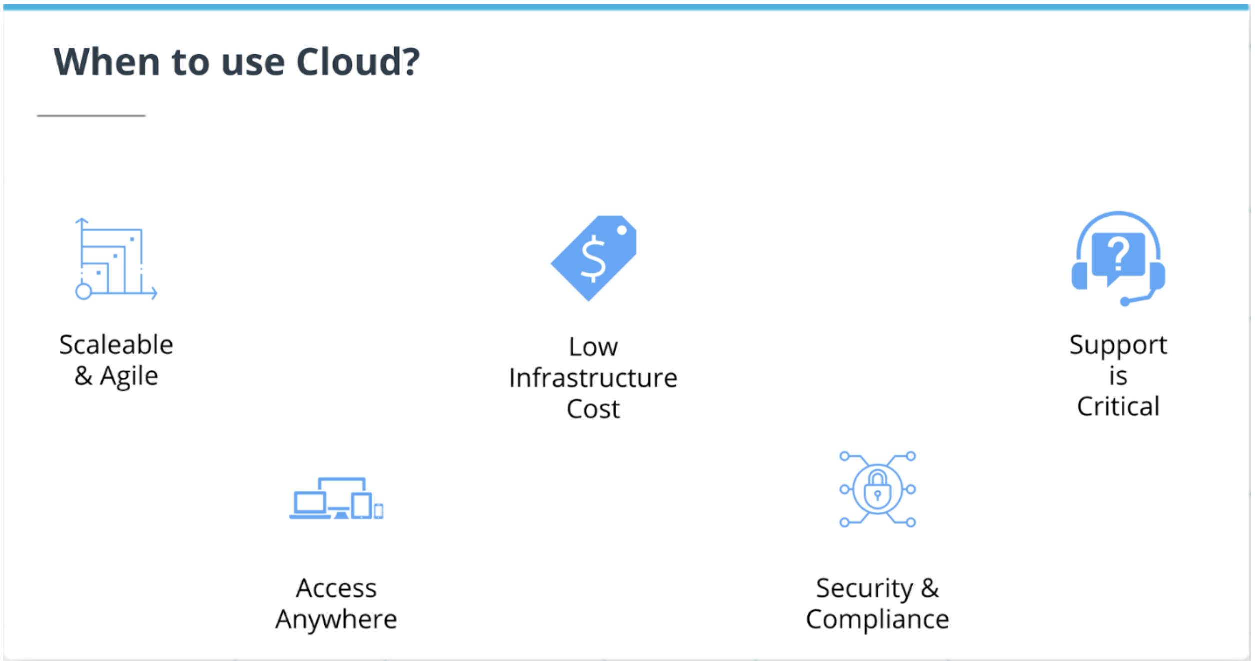
On Prem → Cloud Data Warehouse	Data warehouses are moving to the cloud with increased flexibility, scale, and ease of use—allowing any company to be a data company	snowflake Google Big Query
Hadoop → Next-gen Data Lakes	Data lakes and related systems are becoming more performant and reliable, adding RDBMS-like features including ACID transactions and interactive SQL queries	databricks presto
ETL → ELT	Brittle ETL processes (extract-transform-load) are being replaced with more flexible and consistent ELT pipelines (extract-load-transform)	Fivetran dbt
Workflow Manager → Dataflow Automation	Data flow automation systems are helping to orchestrate thousands of data pipelines with a cleaner abstraction and modern executor integrations	PREFECT DAGSTER Apache Airflow
Analyst Teams → Self-serve Insights	Reporting, dashboarding, and automated analysis tools are becoming more available to non-technical users	Looker Superset
Endpoint Protection → Global Data Governance	Data security and privacy measures (e.g., access controls) are becoming centralized on the data platform as use of data is increasingly regulated and user endpoints are harder to protect	Collibra PRIVACERA

Summary: Cloud-native business intelligence for companies of all sizes – easy to use, inexpensive to get started, and more scalable than past data warehouse patterns


Query and Processing
Sources
Ingestion and Transformation
Storage
Historical
Predictive
Output
OLTP Databases via CDC
Applications/ERP (Oracle, Salesforce, Netsuite, ...)
Event Collectors (Segment, Snowplow)
Logs
3rd Party APIs (e.g., Stripe)
File and Object Storage
...
Connectors (Fivetran, Stitch, Matillion)
Data Modeling (dbt, LookML)
Workflow Manager (Airflow, Dagster, Prefect)
Spark Platform (Databricks, EMR)
Python Libs (Pandas, Boto, Dask, Ray, ...)
Batch Query Engine (Hive)
Event Streaming (Confluent/Kafka, Pulsar, AWS Kinesis)
Stream Processing (Databricks/Spark, Confluent/Kafka, Flink)
Data Warehouse (Snowflake, BigQuery, Redshift)
Data Lake
Databricks/ Delta Lake, Iceberg, Hudi, Hive Acid
Parquet, ORC, Avro
S3, GCS, ABS, HDFS
Data Science Platform (Databricks, Domino, Sagemaker, Dataiku, DataRobot, Anaconda, ...)
Data Science and ML Libraries (Pandas, Numpy, R, Dask, Ray, Spark, ... Scikit-learn, Pytorch, TensorFlow, Spark ML, XGBoost, ...)
Ad Hoc Query Engine (Presto, Dremio/ Drill, Impala)
Real-time Analytics (Imply/Druid, Altinity/ Clickhouse, Rockset)
Dashboards (Looker, Superset, Mode, Tableau)
Embedded Analytics (Sisense, Looker, cube.js)
Augmented Analytics (Thoughtspot, Outlier, Anodot, Sisu)
App Frameworks (Plotly Dash, Streamlit)
Custom Apps
...
Metadata Management (Collibra, Alation, Hive, Metastore, DataHub, ...)
Quality and Testing (Great Expectations)
Entitlements and Security (Privacera, Immuta)
Observability (Unravel, Accel Data, Fiddler)

Strengths of this pattern include low up-front investment, speed and ease of getting started, and wide availability of talent. This blueprint is less appropriate for teams that have more complex data needs – including extensive data science, machine learning, or streaming/ low latency applications.

From Udacity lectures



When to use Cloud?

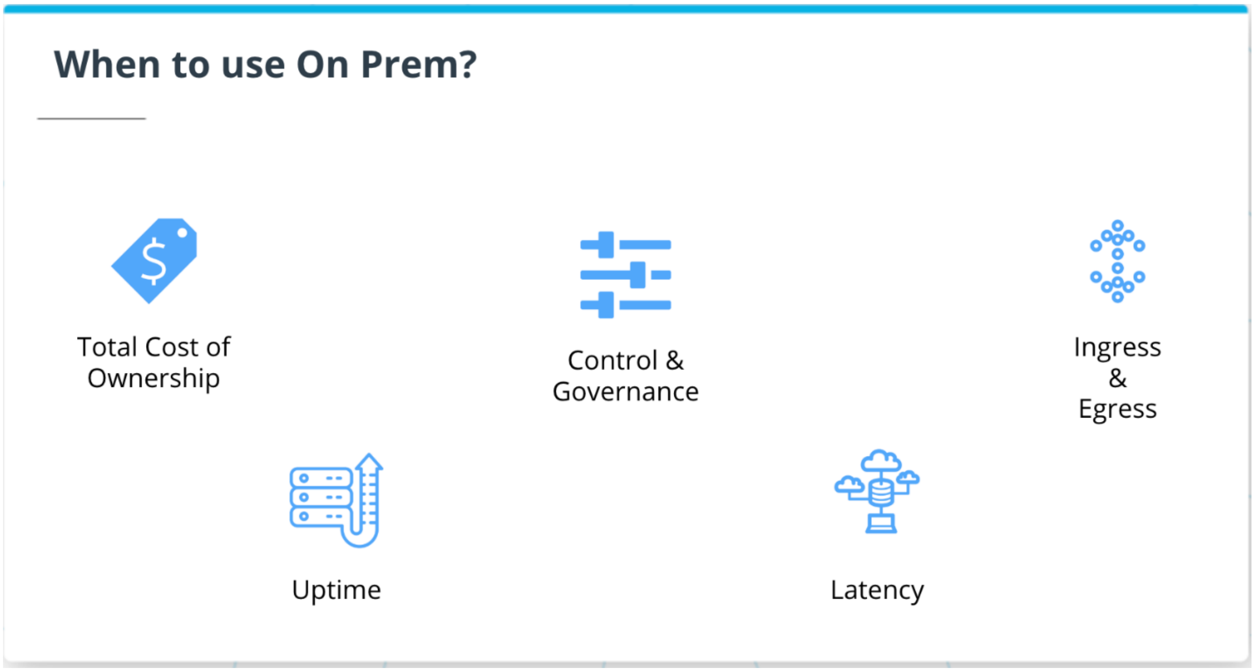
Scaleable & Agile

Low Infrastructure Cost

Support is Critical

Access Anywhere

Security & Compliance



When to use On Prem?

Total Cost of Ownership

Control & Governance

Ingress & Egress

Uptime

Latency

Cloud software advantages

Anywhere and anytime access – You can access your applications anytime and anywhere via a web browser from any device.

Affordable *– Cloud requires no upfront costs, instead you make regular payments which makes it an operating expense (OpEx). While the monthly cost adds up over time, maintenance and support services are included removing the need for annual contracts.*

Predictable costs *– Benefit from predictable monthly payments that cover software licences, upgrades, support and daily back-ups.*

Worry free IT *– Because cloud software is hosted for you, you don't need to worry about the maintenance of your software or the hardware it resides on, compatibility and upgrades are taken care of by the cloud service provider.*

High levels of security *– Data centres employ security measures beyond the affordability of most businesses, therefore your data is often safer in the cloud than on a server in your offices.*

Quick deployment*– Cloud-based software is deployed over the Internet in a matter of hours/days because, compared to on premise applications which needs to be installed on a the physical server and each PC or laptop.*

Scalability *– Cloud technologies provide greater flexibility as you only pay for what you use and can easily scale to meet demand, for example adding and scaling back licences.*

Lower energy costs *– When you move to the cloud, you no longer have to pay to power on-premise servers or to maintain their environment. This significantly reduces the amount you pay on your energy bills.*

The drawbacks

Connectivity – Cloud solutions require reliable internet access for you to remain productive.

Long-term costs – Although requiring a lower upfront investment, cloud applications can be more costly over the course of the system's life cycle, increasing total cost of ownership (TCO).

Less customizable – Cloud software is typically configurable but depending on how it is hosted a cloud solution may not be able to cope with complex development.

On premise deployments

On premise advantages:

Total Cost of Ownership *– Since you are only paying for your user licences once, an on-premise solution can have a lower Total Cost of Ownership (TCO) than a cloud system.*

Complete control *– Your data, hardware and software platforms are all yours. You decide on the configuration, the upgrades and system changes.*

Uptime *– With on-premise systems, you do not rely on internet connectivity or external factors to access your software.*

The drawbacks

Large capital expenditure *– On-premise systems usually require large upfront purchase which means capital expenditure (CapEx) is often required. On top you need to include maintenance costs to ensure support and functionality upgrades.*

Responsibility for maintenance *– With an on premise system, you are responsible for maintaining server hardware and software, data backups, storage and disaster recovery. This can be an issue for smaller companies who have limited budgets and technical resources.*

Longer implementation times *– On-premise implementations take longer due to the time needed to complete installations on servers and each individual computer/laptop.*

Summarize:

Since the service is in MVP status (little capital and specialist) I recommend the cloud solution as we can manage this cost on monthly basis and if the business grows we can still implement the on-premise data architecture.

- We can scale up and down as needed, and pay only for the services/infrastructure used.
- Our teams can focus on innovation and core business functionality instead of focusing on making underlying infrastructure work.

Suggested DWH

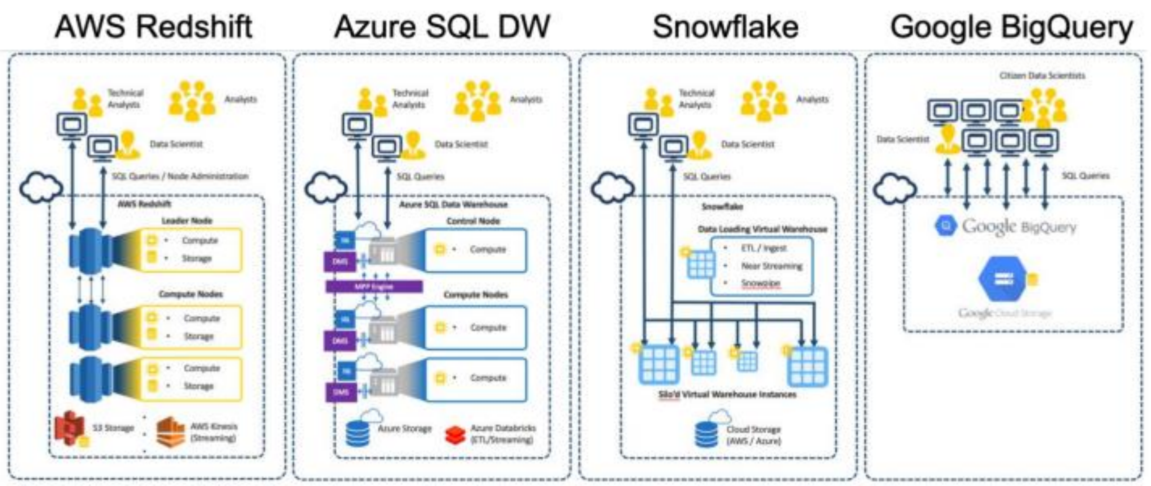
Provide an evidence-based solution as to which DWH product is best for Flyber. Remember to address the factors above.

- ***My suggestion is going with* Google BigQuery**

Reasons:

The Economic Advantages of Google BigQuery versus Alternative Cloud-based EDW Solutions

Figure 3. Functional Comparison of Cloud-based EDW Solutions



Source: Enterprise Strategy Group

Cost

BigQuery eliminates the need to manage virtual EDW nodes as well as the need to monitor, troubleshoot, update, tune, and plan for growth. Google cloud storage is automatically optimized for cost, patching and maintenance is not required, and the support team is well trained and responsive. This leaves administrators more time to focus on other areas of the business.

Scalability

BigQuery scales up or down as needed to meet the changing business demands, enabling organizations to quickly act on new opportunities without the need to plan configuration requirements, pause databases, or spin up dedicated warehouse instances for each organization. The solution also helps to eliminate or reduce the time spent on database administration, ETL management, and new schema modification. BigQuery also is the only solution that provides native AI/ML and supports native integration with many other cloud-based services.

In-house Expertise

At Flyber we are working with Gooogle infrastructure and BigQuery is completely serverless and customers do not have to plan or adjust the supporting infrastructure. With AWS Redshift, the size and quantity of the instances required to handle ingest and workload must be predicted. With Azure SQL DW, users must choose the cDWU rating that best meets their needs. Snowflake users must determine the mix of storage and the size of the virtual data warehouse that best suits the needs of each business unit as well as for a dedicated warehouse used for loading data.

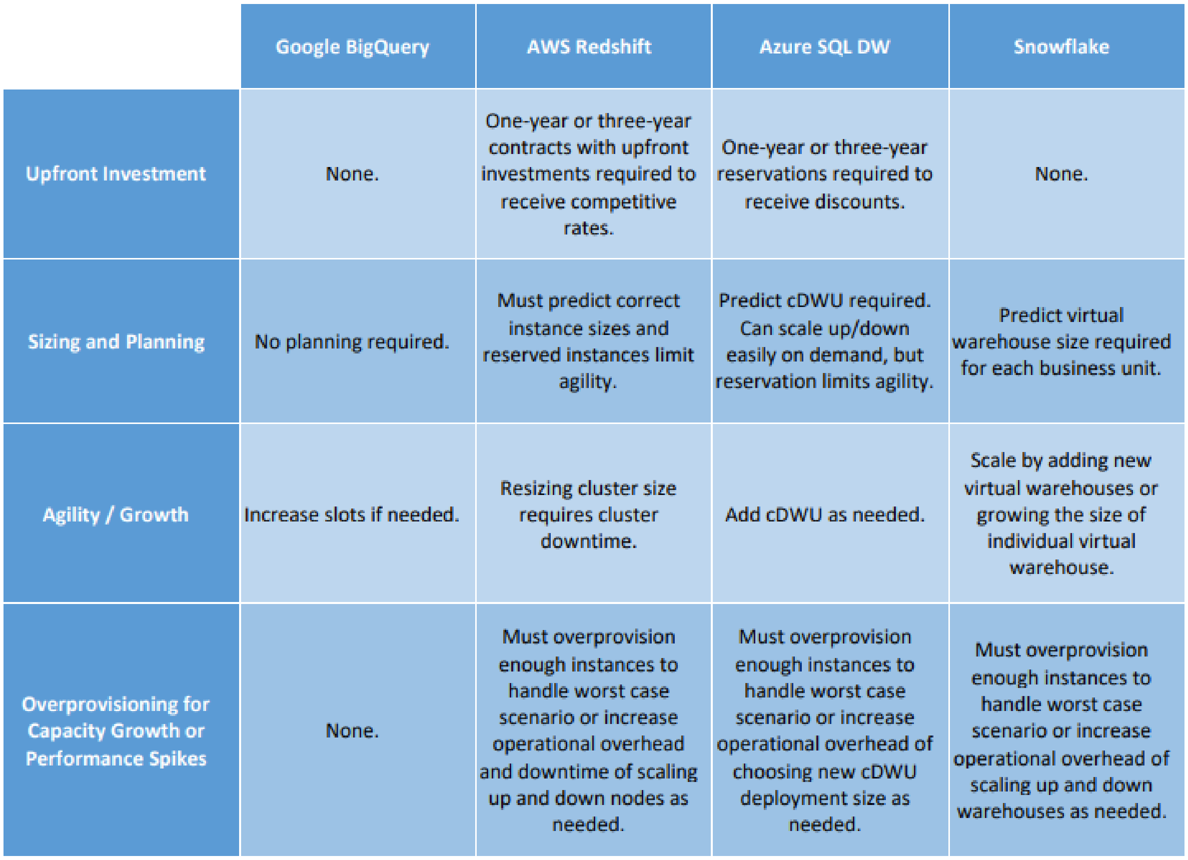
Reliability

Google has SLA: 99.9%

Connectivity

Google have regional coverage but there is option for ***Cloud Interconnect*** *provides low latency, highly available connections that enable you to reliably transfer data between your on-premises and Google Cloud Virtual Private Cloud (VPC) networks.*

Table 1. Comparing Cloud-based EDW Solutions: Upfront Investment, Planning, and Agility



	Google BigQuery	AWS Redshift	Azure SQL DW	Snowflake
Upfront Investment	None.	One-year or three-year contracts with upfront investments required to receive competitive rates.	One-year or three-year reservations required to receive discounts.	None.
Sizing and Planning	No planning required.	Must predict correct instance sizes and reserved instances limit agility.	Predict cDWU required. Can scale up/down easily on demand, but reservation limits agility.	Predict virtual warehouse size required for each business unit.
Agility / Growth	Increase slots if needed.	Resizing cluster size requires cluster downtime.	Add cDWU as needed.	Scale by adding new virtual warehouses or growing the size of individual virtual warehouse.
Overprovisioning for Capacity Growth or Performance Spikes	None.	Must overprovision enough instances to handle worst case scenario or increase operational overhead and downtime of scaling up and down nodes as needed.	Must overprovision enough instances to handle worst case scenario or increase operational overhead of choosing new cDWU deployment size as needed.	Must overprovision enough instances to handle worst case scenario or increase operational overhead of scaling up and down warehouses as needed.

Source: Enterprise Strategy Group

Table 2. Comparing Cloud-based EDW Solutions: Operational Expenses

	Google BigQuery	AWS Redshift	Azure SQL DW	Snowflake
Virtual Node Management	None; completely serverless.	Management node required with manual configuring and scaling of nodes.	Control node required with simple scaling of data warehouse compute units.	Serverless, but may have to manage siloed data warehouses, and a dedicated warehouse to load data is suggested.
Storage Management	Self-optimized storage automatically moves data to cost-effective storage.	Manual storage migration and aging.	Single storage tier.	Must manage data movement and decide between expensive fast storage and economical capacity storage.
EDW / Node Maintenance	Managed by Google in the background with no downtime.	Manual updates of nodes during scheduled downtime.	Updates of nodes during scheduled downtime.	Managed by Snowflake in the background with no downtime.
Enterprise-level Support	Percentage of cloud spend with minimum spend (lesser support levels available).	Percentage of cloud spend with minimum spend (lesser support levels available).	Requires customized quote for premier support (lesser support levels available).	Priced into hourly compute credit cost (lesser support levels available).

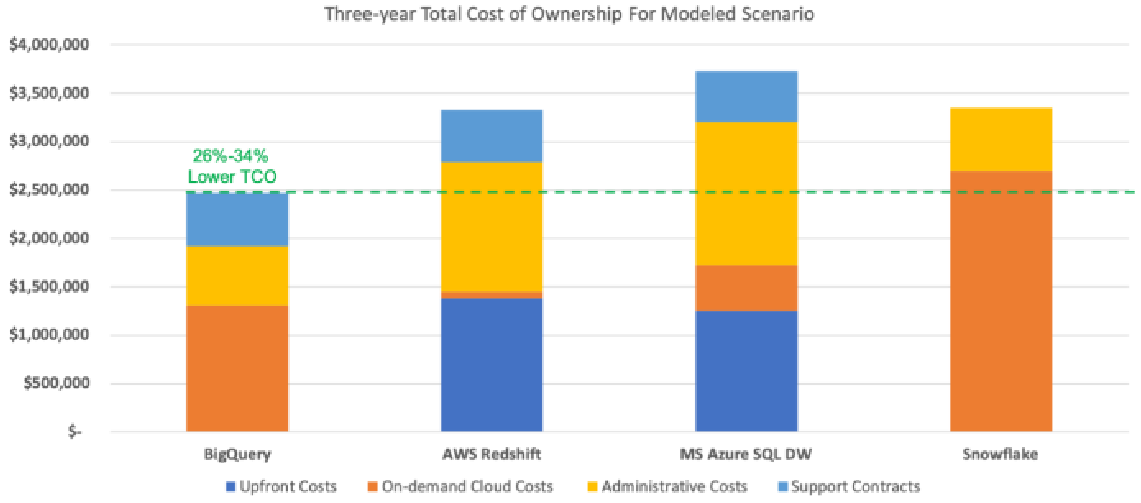
Source: Enterprise Strategy Group

Table 4. Configuration Assumptions Used in ESG Three-year Modeled Scenario

	Google BigQuery	AWS Redshift	Azure SQL DW	Snowflake
Compute / Service Pricing Option	Flat Rate Pricing (Annual Agreement) (paid monthly)	One-year Reserved Instance Pricing (42% Savings) (paid upfront)	One-year Reserved (37% Savings) (paid upfront)	Enterprise+ (AWS US East) ($4.00/Credit)
Compute / Service Configuration	2,000 Fixed Slots	16 x dc2.8xlarge instances	DW6000c	44 Credits (assumed average utilization of 12hrs/day)
Annual Cost of Capital for Upfront Spend	N/A	8%	8%	N/A
Cloud Storage	199.1TB Google Cloud Storage (auto optimized for cost savings)	19.6 TB of usable storage included with instances plus 79.9 TB of S3 storage	99.56 TB of Azure Blob Storage	99.56 TB of Capacity Storage (paid upfront for 43% savings)
Streaming Service / Data Loading	Streaming Inserts	AWS Kinesis	Azure Databricks (Standard tier, 6 DBU)	Large Data Warehouse (8 Credits x 8hrs/day)
Support Level	GCP Business-critical	AWS Enterprise Support	Azure Premier Support	Enterprise+ (Support included in credits)

Source: Enterprise Strategy Group

Figure 4. Estimated Three-year Cloud-based Data Warehouse Solution Total Cost of Ownership (TCO)



Note: Snowflake support contracts included with on-demand cloud costs

Image Appendix

Image 1: Log Growth

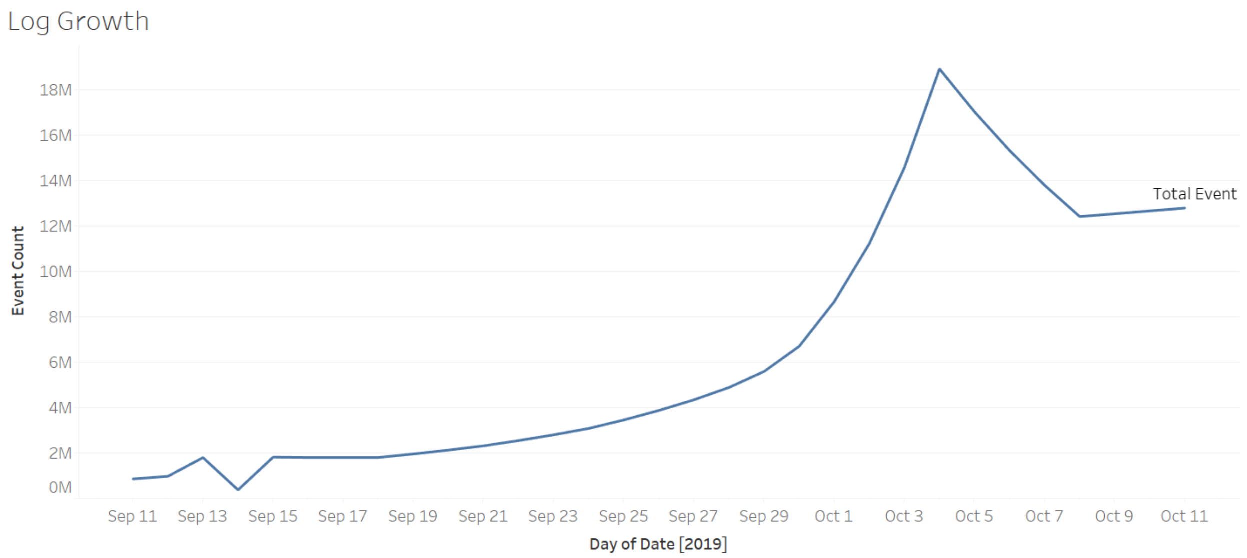


Image 2: Ride Growth

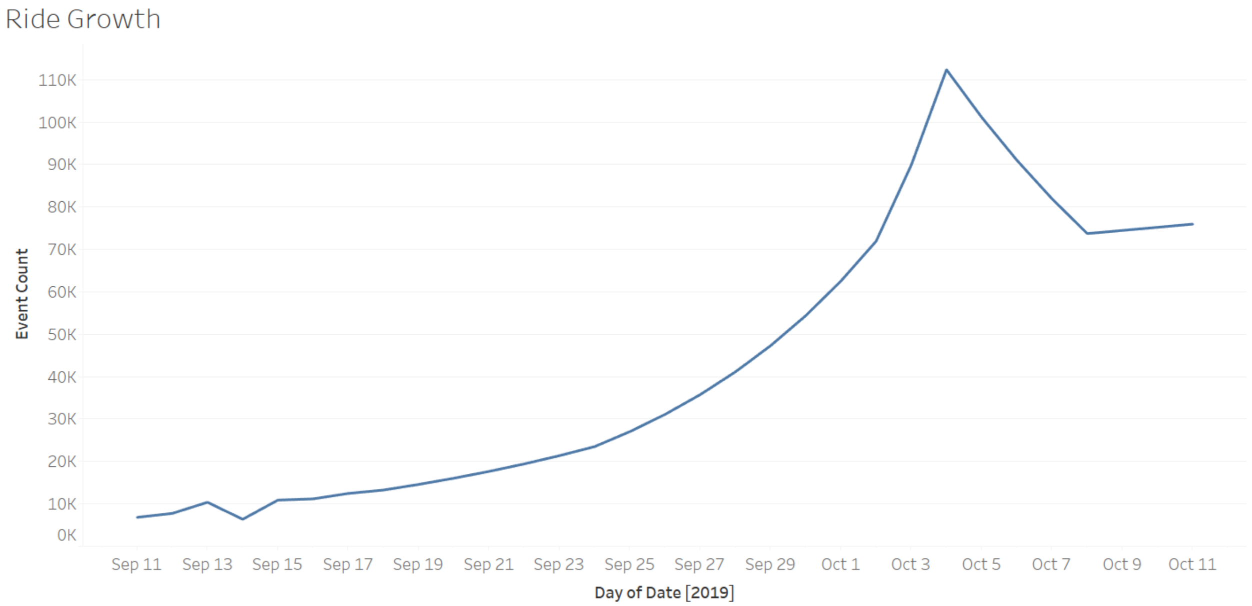


Image 3: Total Event Count

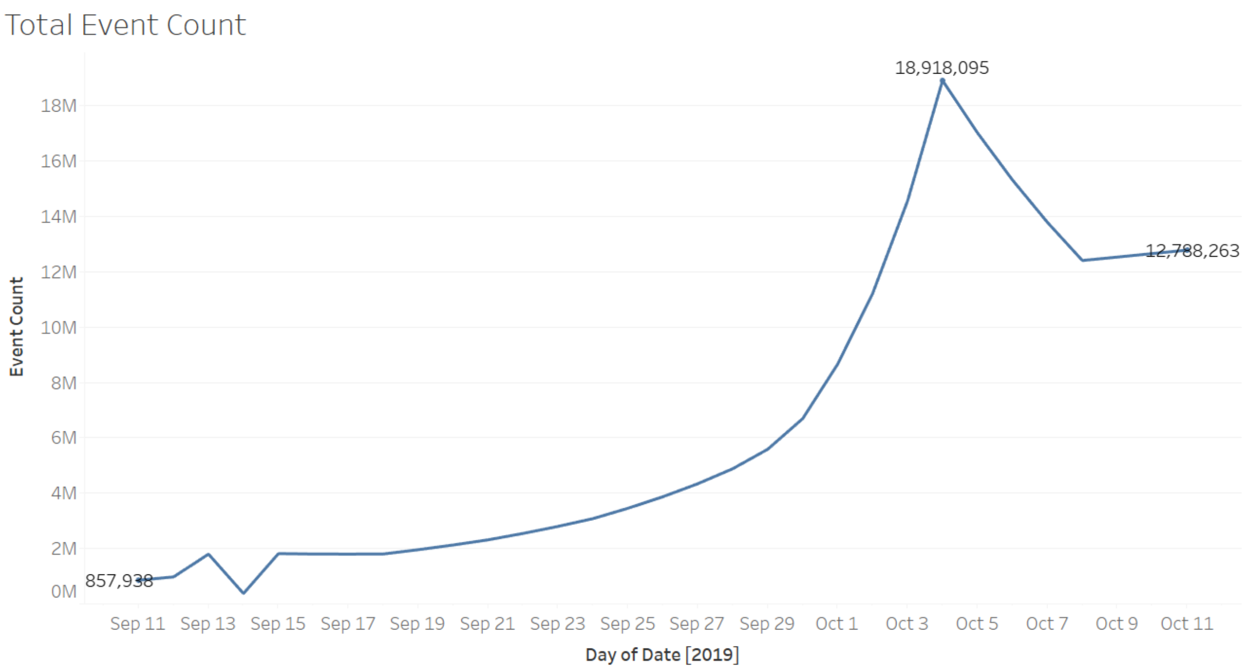
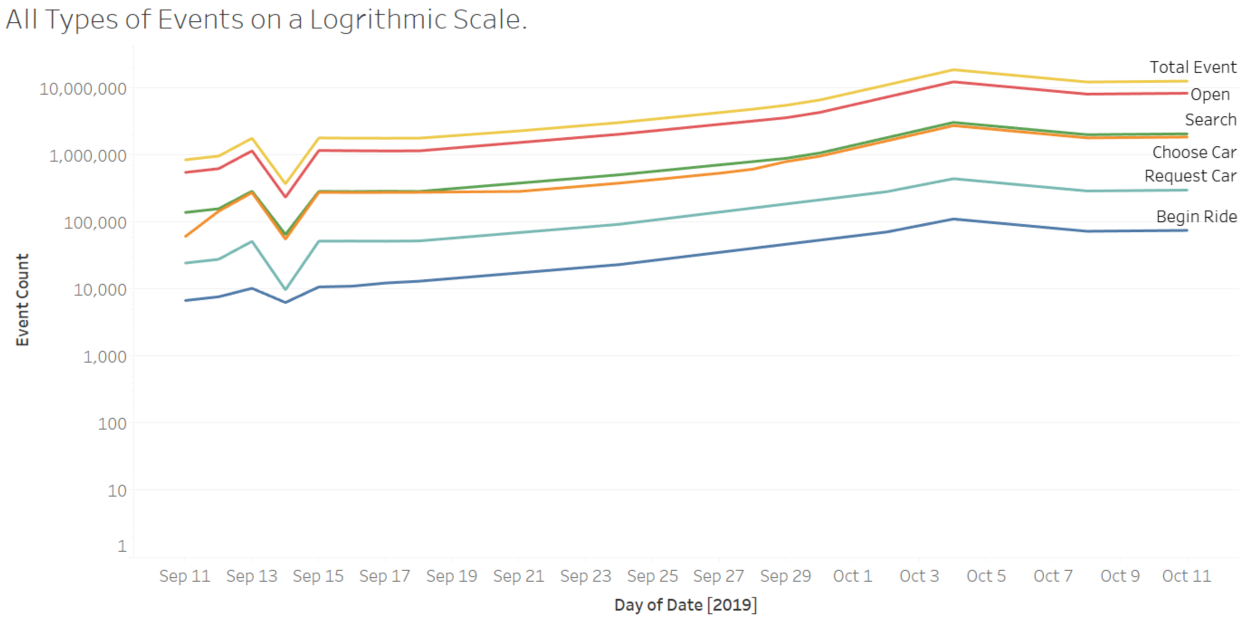


Image 4: All Events Log Scale


All Types of Events on a Logrithmic Scale.
Total Event
Open
Search
Choose Car
Request Car
Begin Ride
Event Count
Sep 11
Sep 13
Sep 15
Sep 17
Sep 19
Sep 21
Sep 23
Sep 25
Sep 27
Sep 29
Oct 1
Oct 3
Oct 5
Oct 7
Oct 9
Oct 11
Day of Date [2019]