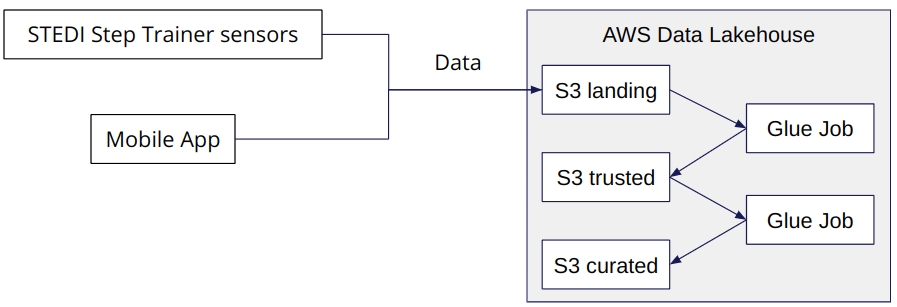
Project Instructions

[**Lesson**](https://learn.udacity.com/nd222-ent-wgu-data-eng-d609?version=1.0.5&partKey=cd12441&lessonKey=c99ad3cc-0c43-42af-9eb3-19e0d36c6348&conceptKey=932e8502-8e63-4e86-80f1-a5323a842180&tab=lesson)[Cloud Resources](https://learn.udacity.com/nd222-ent-wgu-data-eng-d609?version=1.0.5&partKey=cd12441&lessonKey=c99ad3cc-0c43-42af-9eb3-19e0d36c6348&conceptKey=932e8502-8e63-4e86-80f1-a5323a842180&tab=cloud-resources)

Using AWS Glue, AWS S3, Python, and Spark, create or generate Python scripts to build a lakehouse solution in AWS that satisfies these requirements from the STEDI data scientists.

Refer to the flowchart below to better understand the workflow.



A flowchart displaying the workflow.

Requirements

To simulate the data coming from the various sources, you will need to create your own S3 directories for customer\_landing, step\_trainer\_landing, and accelerometer\_landing zones, and copy the data there as a starting point.

* You have decided you want to get a feel for the data you are dealing with in a semi-structured format, so you decide to create **three Glue tables** for the three landing zones. Share your customer\_landing.sql, accelerometer\_landing.sql, and step\_trainer\_landing.sql scripts in git.
* Query those tables using Athena, and take a screenshot of each one showing the resulting data. Name the screenshots customer\_landing(.png,.jpeg, etc.), accelerometer\_landing(.png,.jpeg, etc.), step\_trainer\_landing (.png, .jpeg, etc.).

The Data Science team has done some preliminary data analysis and determined that the **Accelerometer Records** each match one of the **Customer Records**. They would like you to create 2 AWS Glue Jobs that do the following:

1. Sanitize the Customer data from the Website (Landing Zone) and only store the Customer Records who agreed to share their data for research purposes (Trusted Zone) - creating a Glue Table called **customer\_trusted**.
2. Sanitize the Accelerometer data from the Mobile App (Landing Zone) - and only store Accelerometer Readings from customers who agreed to share their data for research purposes (Trusted Zone) - creating a Glue Table called **accelerometer\_trusted**.
3. You need to verify your Glue job is successful and only contains Customer Records from people who agreed to share their data. Query your Glue customer\_trusted table with Athena and take a screenshot of the data. Name the screenshot customer\_trusted(.png,.jpeg, etc.).

Data Scientists have discovered a data quality issue with the Customer Data. The serial number should be a unique identifier for the STEDI Step Trainer they purchased. However, there was a defect in the fulfillment website, and it used the same 30 serial numbers over and over again for millions of customers! Most customers have not received their Step Trainers yet, but those who have, are submitting Step Trainer data over the IoT network (Landing Zone). *The data from the Step Trainer Records has the correct serial numbers.*

The problem is that because of this serial number bug in the fulfillment data (Landing Zone), we don’t know which customer the Step Trainer Records data belongs to.

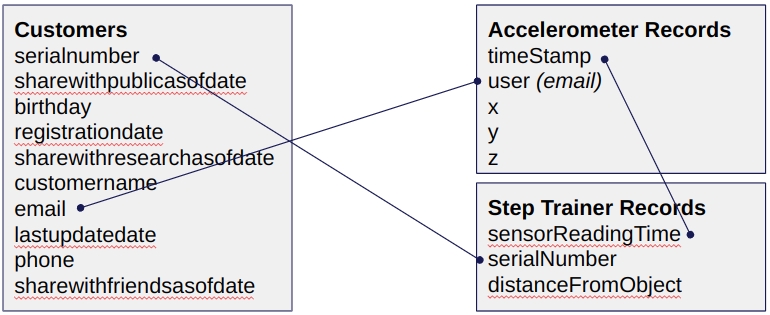
The Data Science team would like you to write a Glue job that does the following:

1. Sanitize the Customer data (Trusted Zone) and create a Glue Table (Curated Zone) that only includes customers who have accelerometer data *and* have agreed to share their data for research called **customers\_curated**.

Finally, you need to create two Glue Studio jobs that do the following tasks:

1. Read the Step Trainer IoT data stream (S3) and populate a Trusted Zone Glue Table called **step\_trainer\_trusted** that contains the Step Trainer Records data for customers who have accelerometer data and have agreed to share their data for research (customers\_curated).
2. Create an aggregated table that has each of the Step Trainer Readings, and the associated accelerometer reading data for the same timestamp, but only for customers who have agreed to share their data, and make a glue table called **machine\_learning\_curated.**

Refer to the relationship diagram below to understand the desired state.



A diagram displaying the relationship between entities.

Check your work!

After each stage of your project, check if the row count in the produced table is correct. You should have the following number of rows in each table:

* Landing
  + Customer: 956
  + Accelerometer: 81273
  + Step Trainer: 28680
* Trusted
  + Customer: 482
  + Accelerometer: 40981
  + Step Trainer: 14460
* Curated
  + Customer: 482
  + Machine Learning: 43681

**Hint:** Use Transform - SQL Query nodes whenever you can. Other node types may give you unexpected results.

For example, rather than a Join node, you may use a SQL node that has two parents, then join them through a SQL query.

Task List

* Python script using Spark that sanitizes the Customer data from the website (Landing Zone) and only stores the Customer Records who agreed to share their data for research purposed creating a Glue Table called customer\_trusted
* A python script using Spark that sanitizes the Accelerometer data from the Mobile App and only stores Readings from customers who agreed to share their data for research purposes creating a Glue Table called accelerometer\_trusted.
* A python script using Spark that sanitizes the customer data and creates a glue table that only includes customers who have accelerometer data and have acgreed to share their data for research called customers)curated
* A python script using Spark that reads the Step Trainer IoT data stream (S3) and populates a Trusted Zone Glue Table called step\_trainer\_trusted containing the Step Trainer Records data for customers who have accelerometer data and have agreed to share their data for research(customers\_curated)
* A python script using spark that creates an aggregated table that has each of the step training readings and the associated accelerometer reading data for the same timestamp but only for customers who have agreed to share their data and populates a glue table called machine\_learning\_curated
* Customer landing.sql, accelerometer\_landing.sql and your step\_trainer\_landing.sql script along with screenshots customer\_landing.(png,jpg,etc), accelerometer\_landing(image) and step\_trainer\_landing(image)
* Check the rubrix and ensure all criteria is met

Rubric

Use this project rubric to understand and assess the project criteria.

Landing Zone

| **Criteria** | **Submission Requirements** |
| --- | --- |
| Use Glue Studio to ingest data from an S3 bucket | **customer\_landing\_to\_trusted.py**, **accelerometer\_landing\_to\_trusted.py**, and **step\_trainer\_trusted.py** Glue jobs have a node that connects to S3 bucket for customer, accelerometer, and step trainer landing zones. |
| Manually create a Glue Table using Glue Console from JSON data | SQL DDL scripts **customer\_landing.sql**, **accelerometer\_landing.sql**, and **step\_trainer\_landing.sql** include all of the JSON fields in the data input files and are appropriately typed (not everything is a string). |
| Use Athena to query the Landing Zone. | Include screenshots showing various queries run on Athena, along with their results:   * Count of customer\_landing: 956 rows * The customer\_landing data contains multiple rows with a blank shareWithResearchAsOfDate. * Count of accelerometer\_landing: 81273 rows * Count of step\_trainer\_landing: 28680 rows |

Trusted Zone

| **Criteria** | **Submission Requirements** |
| --- | --- |
| Configure Glue Studio to dynamically update a Glue Table schema from JSON data | Glue Job Python code shows that the option to dynamically infer and update schema is enabled.  To do this, set the *Create a table in the Data Catalog and, on subsequent runs, update the schema and add new partitions* option to True. |
| Use Athena to query Trusted Glue Tables | Include screenshots showing various queries run on Athena, along with their results:   * Count of customer\_trusted: 482 rows   + The resulting customer\_trusted data has no rows where shareWithResearchAsOfDate is blank. * Count of accelerometer\_trusted: 40981 rows * Count of step\_trainer\_trusted: 14460 rows   However, **if you are following the stand-out suggestions**, your row counts should be as follows:   * Count of customer\_trusted: 482 rows   + The resulting customer\_trusted data has no rows where shareWithResearchAsOfDate is blank. * Count of accelerometer\_trusted: 32025 rows * Count of step\_trainer\_trusted: 14460 rows |
| Filter protected PII with Spark in Glue Jobs | **customer\_landing\_to\_trusted.py** has a node that drops rows that do not have data in the sharedWithResearchAsOfDate column.  Hints:   * **Transform - SQL Query** node often gives more consistent outputs than other node types. * Glue Jobs do not replace any file. Delete your S3 files and Athena table whenever you update your visual ETLs. |
| Join Privacy tables with Glue Jobs | **accelerometer\_landing\_to\_trusted.py** has a node that inner joins the customer\_trusted data with the accelerometer\_landing data by emails. The produced table should have only columns from the accelerometer table. |

Curated Zone

| **Criteria** | **Submission Requirements** |
| --- | --- |
| Write a Glue Job to join trusted data | **customer\_trusted\_to\_curated.py** has a node that inner joins the customer\_trusted data with the accelerometer\_trusted data by emails. The produced table should have only columns from the customer table. |
| Write a Glue Job to create curated data | **step\_trainer\_trusted.py** has a node that inner joins the step\_trainer\_landing data with the customer\_curated data by serial numbers  **machine\_learning\_curated.py** has a node that inner joins the step\_trainer\_trusted data with the accelerometer\_trusted data by sensor reading time and timestamps  Hints:   * **Data Source - S3 bucket** node sometimes extracted incomplete data. Use the **Data Source - Data Catalog** node when that's the case. * Use the Data Preview feature with at least 500 rows to ensure the number of customer-curated rows is correct. Click "Start data preview session", then click the gear next to the "Filter" text box to update the number of rows * As before, the **Transform - SQL Query** node often gives more consistent outputs than any other node type. Tip - replace the JOIN node with it. * The step\_trainer\_trusted may take about 8 minutes to run. |
| Use Athena to query Curated Glue Tables | Include screenshots showing various queries run on Athena, along with their results:   * Count of customer\_curated: 482 rows * Count of machine\_learning\_curated: 43681 rows   However, **if you are following the stand-out suggestions**, your row counts should be as follows:   * Count of customer\_curated: 464 rows * Count of machine\_learning\_curated: 34437 rows   Hint: If you get unexpected results, consider using the **Transform - SQL Query** node rather than Glue-provided nodes. |

Suggestions to Make Your Project Stand Out

Consider these additions to your project to make it stand out!

1. When creating the Glue Job to join data from the accelerometer readings and the customer table, filter out any readings that were prior to the research consent date. This will ensure consent was in place at the time that data was gathered. This helps in the case that in the future the customer revokes consent. We can be sure that the data we used for research was used when consent was in place for that particular data.
2. Anonymize the final curated table so that it is not subject to GDPR or other privacy regulations, in case a customer requests deletion of PII, we will not be in violation by retaining PII data --remove email, and any other personally identifying information up front.