Participant Details			
CRM ID		Module	
Name	Aldo Mema	Delivery Code	245010

Awarding Exercise 1	AO1.1
Evaluate the strengths and limitations of at least 3 methods for assuring data quality *Consider contexts where each might be most appropriate.*	

Data quality is important to establish accurate and reliable information, consistent and good data allows us to make well informed decisions.

Among a number of techniques to ensure data quality such as: cleaning, validation, sampling and consistency. I am evaluating the strength and limitations of 3 methods considering where each might be most appropriate.

Data Cleaning:

Is the process of identifying errors such as duplicates, incompleteness, or incorrect data.

Strengths:

Improves accuracy and reliability of the dataset, enhancing usability for analysis and decision-making reports.

Limitations:

Time consuming, especially for large datasets. Risk of not catching all the errors and accidentally removing valuable data in automated processes.

Data cleaning is applied in every business database, and is particularly appropriate in the context of financial records such as transactions. Keeping records clean is essential in correcting information, removing errors or missing values in the database, preventing frauds, errors or misreporting.

Awarding Exercise 1	AO1.1

Data Consistency:

Very appropriate in supply chain management as an example, consistency ensures that all stakeholders like manufactures, suppliers and distributors access accurate and synchronised up to date information, such as product details, inventories, payments and transports (shippings, invoices).

Strengths:

Ensures that data remains uniform across all the systems, maintaining integrity and compatibility between different sources, avoiding conflicting or contradictory information.

Limitations:

Consistency across systems requires strong governance and standardized rules, adding difficulty across decentralized sources. Difficult to enforce without a solid involvement between all stakeholders, the architecture can cause issues in handling evolving structures.

Data Sampling:

Sampling is the process of subsetting data from a larger dataset into smaller and more manageable pieces. There Are different types of sampling techniques like: systematic sampling, cluster sampling, stratified sampling or simple random sampling. All of these, with their relevant importance, are commonly used across the board.

Strengths:

Sampling enables a better handling of the data, reducing processing time and costs, and avoiding redundancy with overly large datasets. Sampling is key in statistical analysis and when working with big datasets such as large populations.

Limitations:

Risk of bias. Without full representation of the data set, specific categories may be missed, not reflecting the entire base, leading to misleading

Awarding Exercise 1	AO1.1

conclusions.

Sampling is widely used across businesses. In market research and quality control for example, companies will conduct customer satisfaction surveys based on a sampled proportion of their customers base, analysing key details based on their product satisfaction.

Awarding Exercise 2	AO1.2

Explain considerations for creating a data infrastructure solution in line with regulatory requirements.

Include considerations for GDPR and ISO

Creating infrastructure solutions in line with regulatory requirements is mandatory in certain industries and strongly recommended for businesses operating in specific sectors.

Following this general concept, the establishment for the creation of data infrastructures depends on several factors of the business, such as: data use, industry regulations, security requirements and risk management.

GDPR regulations set the rules for organisations on how to handle and store personal information across businesses. With this in mind, the start of the creation of the infrastructure leads the business to establish objectives and requirements considering:

- Type of data needed
- Usage
- Storage
- Collection

Classification is the first step to identify the required business information needed. This data could be personal details (name, addresses, emails), sensitive information (Financial details, health records) and non personal data (Business Analysis, surveys).

Awarding Exercise 2	AO1.2

After the classification there is a need to establish the usage of the data. Usage data examples leads the business to use this information for a certain purpose like:

- Marketing
- Customer services
- Product & Development
- Business Operations

GDPR regulations rule that only necessary data should be collected and personal information must have a clear purpose about what will be the usage.

Some of this data require consent or contractual obligations to legitimate interaction and to comply with the legal obligations. Processing personal or sensitive data determines which and how the information can be included in the storage, encrypted or used by a third-party.

ISO standards (International Organization for Standardization) became relevant within these processes, especially with data management, security and governance. These standards often refer to the GDPR rules. These standards set the rules for: Information security, Protection of personal data, IT Management, Risk Management, Privacy and governance.

These protocols, when implemented correctly, ensure a well rounded approach to manage the infrastructure while addressing all the legal and privacy concerns.

Awarding Exercise 3 | AO2.1

Evaluate the strengths and limitations of the following data capture methods when working with large data sets.

1. Web Scraping
2. API's
3. Manual Entry

Analyse the complexity, time, cost and reliability of each of these methods.

Web Scraping is the process of extracting data from live websites.

Strengths:

- Access Real-Time data
- Cost- effective
- Versatility and Customizable

Web scraping allows for a quick, efficient, and cost-effective extraction of up-to-date data. Information can be collected live from websites and tailored to include only relevant information to the purpose of the scraping.

Limitations:

- Data Quality
- Consistency
- Performance
- Legal Restrictions

While this technique allows for a quick extraction of information, this data can be inconsistent, contain errors or missing values.

Often websites implement limits to prevent scraping, several websites terms and conditions prohibit extraction, leading this process to potential legal risks.

The handling of a large dataset also requires robust infrastructure and technical expertise. Furthermore this type of extraction can lead to incomplete datasets.

Awarding Exercise 3	AO2.1

Api allows different softwares and tools to communicate with each other by using standardised protocols and a set of rules that act as intermediary between data of different systems.

Strength:

- Automation
- Standardization
- Access Control & Security
- Real-Time
- Scalability
- Lower Development costs
- Faster Deployment

Api can be highly effective due to fast deployment and the reduced development time. External Api can reduce maintenance cost for security, and updates lowering in house expenses. Api is ideal for large datasets operations increasing efficiency while reducing manual effort with automations.

The standardised structure characteristics simplifies data processing and integration across different systems, while including solid authentication and reliable security mechanisms to allow only authorised access.

Limitations:

- Data Volume Restrictions
- Dependency on Third Parties
- Extra Costs

While development costs can be low, API usage can be significantly impacted by the volume of the business needs, especially when relying entirely on third parties. External companies charge extra fees based on usage, and can be expensive when dealing with large amounts of data.

A total reliability of third parties can also affect the business control over pricing and the handling of the information which can highly impact costs and the growth.

Awarding Exercise 3	AO2.1

Collecting and recording data manually involves entering information into a database one entry at the time either on paper or computer. Examples for caputerning data manually can be:

- Taking notes
- Paper forms
- Registers
- Ledger Books
- Logbooks

Strengths:

Accountants often verify figures manually by double-checking entries. Manual data entry provides a useful tool for audits in historical tracking, some authorities still require manual financial verification for specific transactions. Cost-effective and simpler compared to complex accounting systems, making it a practical choice for small businesses or those without access to sophisticated softwares.

Limitations:

Manual data entry increases the risk of errors and inaccuracies due to the repetitive workload. It is time-consuming and causes delays because of the manual search and manipulation of records. The process also has a higher risk of human error and inconsistencies due to misentries or typing mistakes. Additionally, manual systems lack real-time updates and make searching for information more difficult.

Awarding Exercise 4	AO2.2

Take the CSV uploaded alongside this document, extract it into a PySpark dataframe

Provide screenshots and an explanation of what you did.

Awarding Exercise 4 AO2.2

1.1 - Running Spark in Jupyter notebook:

To initiate Spark in the environment session and view it correctly there is a need to run this code below in jupyter notebook.

```python
import os
import sys
os.environ["JAVA_HOME"] = "JDK 8"
os.environ["PYSPARK_PYTHON"] = sys.executable
os.environ["PYSPARK_DRIVER_PYTHON"] = sys.executable
```
[13] ✓ 0.0s Python

Fig 1.1

This script configures the operating system environment and ensures compatibility between Java, Python, and Apache Spark.

2.1 - PySPark Dataframe Extraction:

Loading the CSV file dataframe using PySpark offers several advantages, especially when dealing with large datasets. In the figure below the dataframe is loaded in the system.

```python
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("HR_att").getOrCreate()

Hr_Employee = "C:/Users/aldom/Documents/Data_Enginnering/HR-Employee-Attrition.csv"

Attrition_df = spark.read.csv(Hr_Employee, header=True, inferSchema=True)

Attrition_df.show()
```
[16] ✓ 0.9s Python

Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Gender	HourlyRate	JobInvolvement	JobLevel
41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	2	Female	94	3	2
49	No	Travel_Frequently	279	Research & Develo...	8	1	Life Sciences	1	2	3	Male	61	2	2
37	Yes	Travel_Rarely	1373	Research & Develo...	2	2	Other	1	4	4	Male	92	2	1
33	No	Travel_Frequently	1392	Research & Develo...	3	4	Life Sciences	1	5	4	Female	56	3	1
27	No	Travel_Rarely	591	Research & Develo...	2	1	Medical	1	7	1	Male	40	3	1
32	No	Travel_Frequently	1005	Research & Develo...	2	2	Life Sciences	1	8	4	Male	79	3	1
59	No	Travel_Rarely	1324	Research & Develo...	3	3	Medical	1	10	3	Female	81	4	1
30	No	Travel_Rarely	1358	Research & Develo...	24	1	Life Sciences	1	11	4	Male	67	3	1
38	No	Travel_Frequently	216	Research & Develo...	23	3	Life Sciences	1	12	4	Male	44	2	3
36	No	Travel_Rarely	1299	Research & Develo...	27	3	Medical	1	13	3	Male	94	3	2
35	No	Travel_Rarely	809	Research & Develo...	16	3	Medical	1	14	1	Male	84	4	1
29	No	Travel_Rarely	153	Research & Develo...	15	2	Life Sciences	1	15	4	Female	49	2	2
31	No	Travel_Rarely	670	Research & Develo...	26	1	Life Sciences	1	16	1	Male	31	3	1
34	No	Travel_Rarely	1346	Research & Develo...	19	2	Medical	1	18	2	Male	93	3	1
28	Yes	Travel_Rarely	103	Research & Develo...	24	3	Life Sciences	1	19	3	Male	50	2	1
29	No	Travel_Rarely	1389	Research & Develo...	21	4	Life Sciences	1	20	2	Female	51	4	3
32	No	Travel_Rarely	334	Research & Develo...	5	2	Life Sciences	1	21	1	Male	80	4	1
22	No	Non-Travel	1123	Research & Develo...	16	2	Medical	1	22	4	Male	96	4	1
53	No	Travel_Rarely	1219	Sales	2	4	Life Sciences	1	23	1	Female	78	2	4
38	No	Travel_Rarely	371	Research & Develo...	2	3	Life Sciences	1	24	4	Male	45	3	1

only showing top 20 rows

Fig 2.1

I have named this project "HR_att" after importing Spark Session to initialise the system ensuring that my session is viewable for future manipulations.

Line 3 of this, defines the file path of my dataset, which is stored locally in my system drive. Within a dedicated folder for the project, this file contains the data that will be used for the analysis.

Awarding Exercise 4

AO2.2

In line 4 I am reading the file into a Dataframe named: Attrition_df.

This line also tells Spark that the first row of the file contains column names (header=True), and to don't threat as strings the data type in the file dataset (inferSchema=True) determining each column information based on the value (integer,float,date) Without (inferSchema=True), Spark will treat all the data in the CSV as strings.

Line 5 will display the DataFrame.

Awarding Exercise 5

AO4.1

Transform the data into a more streamline and usable format. The transformations required are:

1. Clear redundant columns
2. Rename columns to follow snake case format
3. Drop duplicate entries
4. Remove NaN values
5. Any other transformations you see fit

Explore the data first to get a feel for it. Provide screenshots and an explanation of what you did.

Awarding Exercise 5

AO4.1

Dataset transformation Exercise 5:

1. Clearing Redundant columns by dropping (.drop) from the original file.

Dropped: “EmployeeCount”, “Over 18” and “StandardHours” as they all have single values which make them redundant.

```
#Clearing reduntatd columns

Attrition_df = Attrition_df.drop("EmployeeCount")

Attrition_df = Attrition_df.drop("Over18")

Attrition_df = Attrition_df.drop("StandardHours")

Attrition_df.show()
```

[48] ✓ 2.1s Python

```
+---+---------+-----------------+---------+--------------------+-----------------+---------+-----------------+-------
|Age|Attrition|  business_travel|daily_rate|          Department|distance_from_home|Education|  education_field|employe
+---+---------+-----------------+---------+--------------------+-----------------+---------+-----------------+-------
| 18|       No|       Non-Travel|      287|Research & Develo...|                5|        2|    Life Sciences|
| 18|      Yes|Travel_Frequently|      544|               Sales|                3|        2|          Medical|
| 18|      Yes|Travel_Frequently|     1306|               Sales|                5|        3|        Marketing|
| 18|       No|       Non-Travel|     1124|Research & Develo...|                1|        3|    Life Sciences|
| 18|       No|       Non-Travel|     1431|Research & Develo...|               14|        3|          Medical|
| 18|       No|    Travel_Rarely|      812|               Sales|               10|        3|          Medical|
| 18|      Yes|    Travel_Rarely|      230|Research & Develo...|                3|        3|    Life Sciences|
| 18|      Yes|       Non-Travel|      247|Research & Develo...|                8|        1|          Medical|
| 19|       No|    Travel_Rarely|      645|Research & Develo...|                9|        2|    Life Sciences|
| 19|      Yes|Travel_Frequently|      602|               Sales|                1|        1|Technical Degree|
| 19|      Yes|    Travel_Rarely|      419|               Sales|               21|        3|            Other|
| 19|      Yes|    Travel_Rarely|      303|Research & Develo...|                2|        3|    Life Sciences|
| 19|       No|    Travel_Rarely|      265|Research & Develo...|               25|        3|    Life Sciences|
```

Fig 1.1

2. Renaming column following snake case format.

For this task I have created a dictionary and renamed the columns using the snake case format (Ex. BusinessTravel to “business_trave”).

I made a dictionary to rewrite the *old_name* to the *new_name* with a (for in) loop function.

>>> *for* old_name, new_name *in* Format_dict.**items()**:

>>>Attrition_df = Attrition_df.**withColumnRenamed**(old_name, new_name)

>>> Attrition_df.show()

```python
#colums renaming with snake case format

Format_dict = {

    "BusinessTravel": "business_travel",
    "DailyRate": "daily_rate",
    "DistanceFromHome": "distance_from_home",
    "EducationField": "education_field",
    "EmployeeNumber" : "employee_number",
    "EnvironmentSatisfaction": "environment_satisfaction",
    "HourlyRate": "hourly_rate",
    "JobInvolvement": "job_involvement",
    "JobLevel": "job_level",
    "JobRole": "job_role",
    "JobSatisfaction": "job_satisfaction",
    "MaritalStatus": "marital_status",
    "MonthlyIncome" : "monthly_income",
    "MonthlyRate": "monthly_rate",
    "NumCompaniesWorked": "num_companies_worked",
    "OverTime": "over_time",
    "PercentSalaryHike": "percent_salary_hike",
    "PerformanceRating": "performance_rating",
    "RelationshipSatisfaction": "relationship_satisfaction",
    "StockOptionLevel": "stock_option_level",
    "TotalWorkingYears": "total_working_years",
    "TrainingTimesLastYear": "training_times_last_year",
    "WorkLifeBalance": "work_life_balance",
    "YearsAtCompany": "years_at_company",
    "YearsInCurrentRole": "years_in_current_role",
    "YearsWithCurrManager": "years_with_curr_manager",
}

for old_name, new_name in Format_dict.items():
    Attrition_df = Attrition_df.withColumnRenamed(old_name, new_name)

Attrition_df.show()
```

[5] ✓ 1.0s Python

```
+---+---------+-----------------+----------+--------------------+------------------+---------+----------------+---------------+------------------------+------+-----------+---------------+---------+-----
|Age|Attrition|  business_travel|daily_rate|          Department|distance_from_home|Education| education_field|employee_number|environment_satisfaction|Gender|hourly_rate|job_involvement|job_level|
+---+---------+-----------------+----------+--------------------+------------------+---------+----------------+---------------+------------------------+------+-----------+---------------+---------+-----
| 41|      Yes|    Travel_Rarely|      1102|               Sales|                 1|        2|   Life Sciences|              1|                       2|Female|         94|              3|        2|
| 49|       No|Travel_Frequently|       279|Research & Develo...|                 8|        1|   Life Sciences|              2|                       3|  Male|         61|              2|        2|  Res
| 37|      Yes|    Travel_Rarely|      1373|Research & Develo...|                 2|        2|           Other|              4|                       4|  Male|         92|              2|        1|Labor
| 33|       No|Travel_Frequently|      1392|Research & Develo...|                 3|        4|   Life Sciences|              5|                       4|Female|         56|              3|        1|  Res
```

Fig 2.1

3. Dropping duplicate entries using (.dropDuplicates)

>>> Attrition_df = Attrition_df**.dropDuplicates()**

```python
#dropping  duplicate entries

Attrition_df = Attrition_df.dropDuplicates()

Attrition_df.show()
```

[6] ✓ 4.8s Python

```
+---+---------+-----------------+----------+--------------------+------------------+---------+----------------+---------------+------------------------+------+-----------+---------------+---------+----
|Age|Attrition|  business_travel|daily_rate|          Department|distance_from_home|Education| education_field|employee_number|environment_satisfaction|Gender|hourly_rate|job_involvement|job_level|
+---+---------+-----------------+----------+--------------------+------------------+---------+----------------+---------------+------------------------+------+-----------+---------------+---------+----
| 30|       No|    Travel_Rarely|       288|Research & Develo...|                 2|        3|   Life Sciences|            117|                       3|  Male|         99|              2|        2|Heal
| 33|      Yes|    Travel_Rarely|       813|Research & Develo...|                14|        3|         Medical|            325|                       3|  Male|         58|              3|        1|Labo
| 37|       No|Travel_Frequently|       889|Research & Develo...|                 9|        3|         Medical|            403|                       2|  Male|         53|              3|        1|  Re
| 43|       No|Travel_Frequently|      1001|Research & Develo...|                 9|        5|         Medical|            663|                       4|  Male|         72|              3|        2|Labo
| 34|       No|    Travel_Rarely|       121|Research & Develo...|                 2|        4|         Medical|            804|                       3|Female|         86|              2|        1|  Re
| 33|      Yes|    Travel_Rarely|       118|               Sales|                16|        3|       Marketing|            819|                       1|Female|         69|              3|        2|
| 41|       No|    Travel_Rarely|       263|Research & Develo...|                 6|        3|         Medical|            957|                       4|  Male|         59|              3|        1|Labo
| 38|       No|    Travel_Rarely|      1035|               Sales|                 3|        4|   Life Sciences|           1036|                       2|  Male|         42|              3|        2|
| 53|       No|Travel_Frequently|       124|               Sales|                 2|        3|       Marketing|           1050|                       3|Female|         38|              2|        3|
| 36|       No|    Travel_Rarely|      1157|               Sales|                 2|        4|   Life Sciences|           1556|                       3|  Male|         70|              3|        1|Sale
| 45|       No|    Travel_Rarely|      1015|Research & Develo...|                 5|        5|         Medical|           1611|                       3|Female|         50|              1|        2|Labo
| 45|       No|    Travel_Rarely|      1329|Research & Develo...|                 2|        2|           Other|           1635|                       4|Female|         59|              2|        2|Manu
| 34|       No|       Non-Travel|      1375|               Sales|                10|        3|   Life Sciences|           1774|                       4|  Male|         87|              3|        2|
| 42|       No|    Travel_Rarely|      1128|Research & Develo...|                13|        3|         Medical|           1803|                       2|  Male|         95|              4|        2|Heal
| 57|       No|    Travel_Rarely|       334|Research & Develo...|                24|        2|   Life Sciences|            223|                       3|  Male|         83|              4|        3|Heal
| 40|       No|       Non-Travel|      1151|Research & Develo...|                 9|        5|   Life Sciences|            287|                       4|  Male|         63|              2|        2|Heal
| 21|      Yes|    Travel_Rarely|      1427|Research & Develo...|                18|        1|           Other|            923|                       4|Female|         65|              3|        1|  Re
| 35|       No|    Travel_Rarely|       660|               Sales|                 7|        1|   Life Sciences|           1492|                       4|  Male|         76|              3|        1|Sale
| 34|       No|    Travel_Rarely|       971|               Sales|                 1|        3|Technical Degree|           1535|                       4|  Male|         64|              2|        3|
| 30|       No|Travel_Frequently|      1312|Research & Develo...|                23|        3|   Life Sciences|            159|                       1|  Male|         96|              1|        1|  Re
+---+---------+-----------------+----------+--------------------+------------------+---------+----------------+---------------+------------------------+------+-----------+---------------+---------+----
only showing top 20 rows
```

Fig 3.1

Awarding Exercise 5

AO4.1

4. Removing Nan values using (.dropna)

>>> Attrition_df = Attrition_df.**dropna()**

```
#remove NAN values

Attrition_df = Attrition_df.dropna()

Attrition_df.show()
```

[7] ✓ 2.1s Python

Age	Attrition	business_travel	daily_rate	Department	distance_from_home	Education	education_field	employee_number	environment_satisfaction	Gender	hourly_rate	job_involvement	job_level	
30	No	Travel_Rarely	288	Research & Develo...	2	3	Life Sciences	117	3	Male	99	2	2	Heal
33	Yes	Travel_Rarely	813	Research & Develo...	14	3	Medical	325	3	Male	58	3	1	Labo
37	No	Travel_Frequently	889	Research & Develo...	9	3	Medical	403	2	Male	53	3	1	Re
43	No	Travel_Frequently	1001	Research & Develo...	9	5	Medical	663	4	Male	72	3	2	Labo
34	No	Travel_Rarely	121	Research & Develo...	2	4	Medical	804	3	Female	86	2	1	Re
33	Yes	Travel_Rarely	118	Sales	16	3	Marketing	819	1	Female	69	3	2	
41	No	Travel_Rarely	263	Research & Develo...	6	3	Medical	957	4	Male	59	3	1	Labo
38	No	Travel_Rarely	1035	Sales	3	4	Life Sciences	1036	2	Male	42	3	2	
53	No	Travel_Frequently	124	Sales	2	3	Marketing	1050	3	Female	38	2	3	
36	No	Travel_Rarely	1157	Sales	2	4	Life Sciences	1556	3	Male	70	3	1	Sale
45	No	Travel_Rarely	1015	Research & Develo...	5	5	Medical	1611	3	Female	50	1	2	Labo
45	No	Travel_Rarely	1329	Research & Develo...	2	2	Other	1635	4	Female	59	2	2	Manu
34	No	Non-Travel	1375	Sales	10	3	Life Sciences	1774	4	Male	87	3	2	
42	No	Travel_Rarely	1128	Research & Develo...	13	3	Medical	1803	2	Male	95	4	2	Heal
57	No	Travel_Rarely	334	Research & Develo...	24	2	Life Sciences	223	3	Male	83	4	3	Heal
40	No	Non-Travel	1151	Research & Develo...	9	5	Life Sciences	287	4	Male	63	2	2	Heal
21	Yes	Travel_Rarely	1427	Research & Develo...	18	1	Other	923	4	Female	65	3	1	Re
35	No	Travel_Rarely	660	Sales	7	1	Life Sciences	1492	4	Male	76	3	1	Sale
34	No	Travel_Rarely	971	Sales	1	3	Technical Degree	1535	4	Male	64	2	3	
30	No	Travel_Frequently	1312	Research & Develo...	23	3	Life Sciences	159	1	Male	96	1	1	Re

only showing top 20 rows

Fig 4.1

5. Any other transformations

For the purpose of this exercise I have ordered the dataset by ascending age with (.ordeBy), from younger age to older age:

>>> Attrition_df = Attrition_df.**orderBy("Age", ascending=True)**

```
#More transformation ordering based on age

Attrition_df = Attrition_df.orderBy("Age", ascending=True)

Attrition_df.show()
```

[8] ✓ 2.1s Pytho

Age	Attrition	business_travel	daily_rate	Department	distance_from_home	Education	education_field	employee_number	environment_satisfaction	Gender	hourly_rate	job_involvement	job_level	
18	No	Non-Travel	287	Research & Develo...	5	2	Life Sciences	1012	2	Male	73	3	1	
18	Yes	Travel_Frequently	544	Sales	3	2	Medical	1624	2	Female	70	3	1	Sa
18	Yes	Travel_Frequently	1306	Sales	5	3	Marketing	614	2	Male	69	3	1	Sa
18	No	Non-Travel	1124	Research & Develo...	1	3	Life Sciences	1368	4	Female	97	3	1	La
18	No	Non-Travel	1431	Research & Develo...	14	3	Medical	1839	2	Female	33	3	1	
18	No	Travel_Rarely	812	Sales	10	3	Medical	411	4	Female	69	2	1	Sa
18	Yes	Travel_Rarely	230	Research & Develo...	3	3	Life Sciences	405	3	Male	54	3	1	La
18	Yes	Non-Travel	247	Research & Develo...	8	1	Medical	1156	3	Male	80	3	1	La
19	No	Travel_Rarely	645	Research & Develo...	9	2	Life Sciences	1193	3	Male	54	3	1	
19	Yes	Travel_Frequently	602	Sales	1	1	Technical Degree	235	3	Female	100	1	1	Sa
19	Yes	Travel_Rarely	419	Sales	21	3	Other	959	4	Male	37	2	1	Sa
19	Yes	Travel_Rarely	303	Research & Develo...	2	3	Life Sciences	243	2	Male	47	2	1	La
19	No	Travel_Rarely	265	Research & Develo...	25	3	Life Sciences	1269	2	Female	57	4	1	
19	No	Travel_Rarely	1181	Research & Develo...	3	1	Medical	201	2	Female	79	3	1	La
19	Yes	Travel_Rarely	528	Sales	22	1	Marketing	167	4	Male	50	3	1	Sa
19	Yes	Travel_Rarely	489	Human Resources	2	2	Technical Degree	566	1	Male	52	2	1	
19	Yes	Non-Travel	504	Research & Develo...	10	3	Medical	1248	1	Female	96	2	1	
20	Yes	Travel_Rarely	500	Sales	2	3	Medical	922	3	Female	49	2	1	Sa
20	No	Travel_Rarely	1141	Sales	2	3	Medical	1657	3	Female	31	3	1	Sa
20	No	Travel_Rarely	727	Sales	9	1	Life Sciences	1680	4	Male	54	3	1	Sa

only showing top 20 rows

Fig 5.1

Awarding Exercise 5	AO4.1

Data transformation is crucial to the quality of the dataset removing inconsistent or missing values making it suitable for the analysis while reducing complexity and improving interpretability. With this exercise I ensured my database is actionable for further manipulations and structured without errors for the purpose of any future analysis.

Awarding Exercise 6	AO4.1

Save the cleaned dataframe to your SQL database and csv file and turn it into a view in PySpark. Query the data to retrieve the:

1. Average age of the employees
2. Most popular department
3. The median distance from home
4. Most common level of education

Provide screenshots and an explanation of what you did.

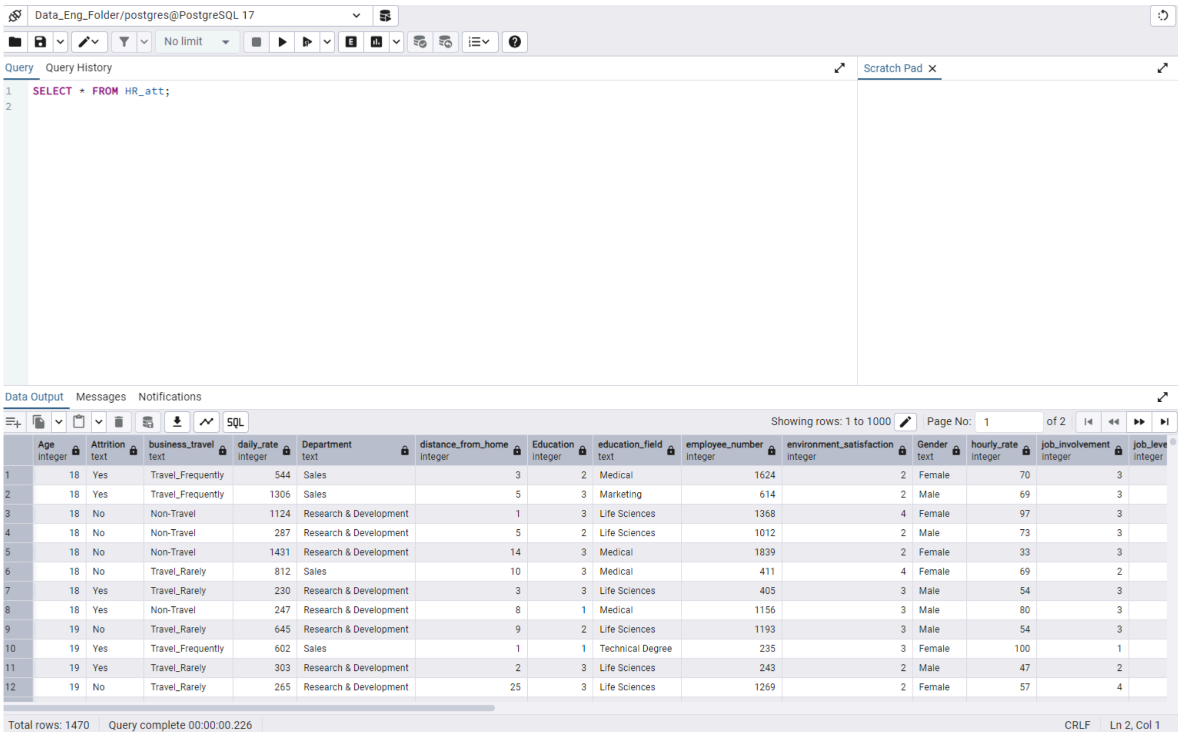
Saving this dataframe in SQL is pictured below showing the coding from Spark to save the dataframe into my SQL Database.

```
#Importing file in postgresql

url = "jdbc:postgresql://localhost:5432/Data_Eng_Folder"
properties = {
    "user": "postgres",
    "password": "[redacted]",
    "driver": "org.postgresql.Driver"
}

# Saving the file in my database

Attrition_df.write.jdbc(url=url, table="HR_att", mode="overwrite", properties=properties)

Attrition_df.show()
```

[52] ✓ 2.9s Python

Awarding Exercise 6

AO4.1



	Age integer	Attrition text	business_travel text	daily_rate integer	Department text	distance_from_home integer	Education integer	education_field text	employee_number integer	environment_satisfaction integer	Gender text	hourly_rate integer	job_involvement integer	job_leve integer
1	18	Yes	Travel_Frequently	544	Sales	3	2	Medical	1624	2	Female	70	3	
2	18	Yes	Travel_Frequently	1306	Sales	5	3	Marketing	614	2	Male	69	3	
3	18	No	Non-Travel	1124	Research & Development	1	3	Life Sciences	1368	4	Female	97	3	
4	18	No	Non-Travel	287	Research & Development	5	2	Life Sciences	1012	2	Male	73	3	
5	18	No	Non-Travel	1431	Research & Development	14	3	Medical	1839	2	Female	33	3	
6	18	No	Travel_Rarely	812	Sales	10	3	Medical	411	4	Female	69	2	
7	18	Yes	Travel_Rarely	230	Research & Development	3	3	Life Sciences	405	3	Male	54	3	
8	18	Yes	Non-Travel	247	Research & Development	8	1	Medical	1156	3	Male	80	3	
9	19	No	Travel_Rarely	645	Research & Development	9	2	Life Sciences	1193	3	Male	54	3	
10	19	Yes	Travel_Frequently	602	Sales	1	1	Technical Degree	235	3	Female	100	1	
11	19	Yes	Travel_Rarely	303	Research & Development	2	3	Life Sciences	243	2	Male	47	2	
12	19	No	Travel_Rarely	265	Research & Development	25	3	Life Sciences	1269	2	Female	57	4	

The above picture shows my database saved and displayed in PostgreSQL.

Awarding Exercise 6

AO4.1

```
import pandas as pd

Hr_csv_saved_df = pd.read_csv('HR-Employee-Attrition.csv')

print(Hr_csv_saved_df.head())
```

[44] ✓ 0.0s

```
...
   Age Attrition     BusinessTravel  DailyRate              Department  \
0   41       Yes      Travel_Rarely       1102                   Sales
1   49        No  Travel_Frequently        279  Research & Development
2   37       Yes      Travel_Rarely       1373  Research & Development
3   33        No  Travel_Frequently       1392  Research & Development
4   27        No      Travel_Rarely        591  Research & Development

   DistanceFromHome  Education EducationField  EmployeeCount  EmployeeNumber  \
0                 1          2  Life Sciences              1               1
1                 8          1  Life Sciences              1               2
2                 2          2          Other              1               4
3                 3          4  Life Sciences              1               5
4                 2          1        Medical              1               7

   ...  RelationshipSatisfaction StandardHours  StockOptionLevel  \
0  ...                         1            80                 0
1  ...                         4            80                 1
2  ...                         2            80                 0
3  ...                         3            80                 0
4  ...                         4            80                 1

   TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  YearsAtCompany  \
0                  8                      0               1               6
1                 10                      3               3              10
2                  7                      3               3               0
...
3                  7                        3                     0
4                  2                        2                     2

[5 rows x 35 columns]
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

To save the CSV file locally I have used panda in the picture above while making a temporary file (named = “attrition_spark_view”) to query the database from spark with SQL.

Average Age of employees

The average age of employees is: 37.

Awarding Exercise 6

AO4.1

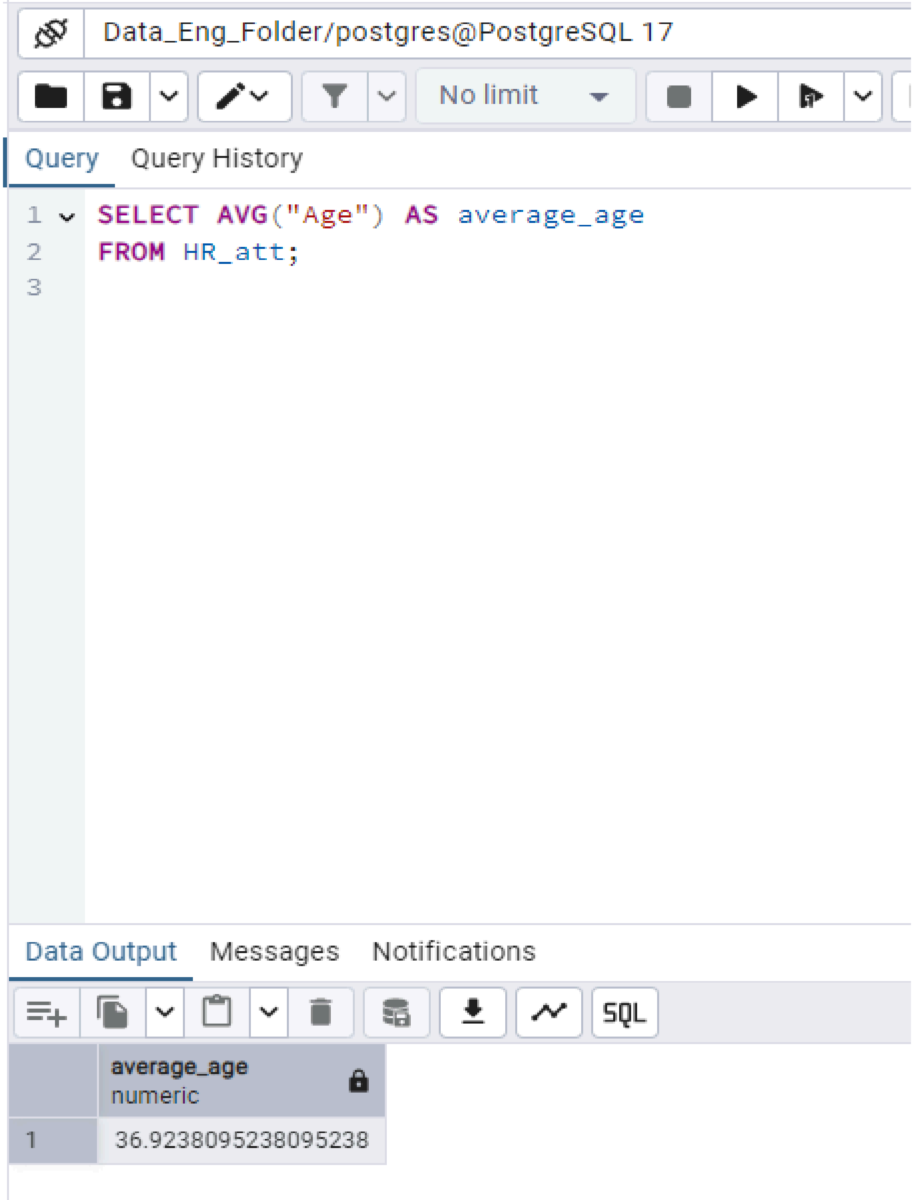
```
Attrition_df.createOrReplaceTempView("attrition_spark_view")

# Avergae age of employees

average_age = spark.sql("SELECT AVG(age) AS avg_age FROM attrition_spark_view")

average_age.show()
```

[45] ✓ 1.4s

```
+------------------+
|           avg_age|
+------------------+
|36.923809523809524|
+------------------+
```

Awarding Exercise 6

AO4.1



Most Popular department

The most popular department is: Research and Development.

Awarding Exercise 6	AO4.1

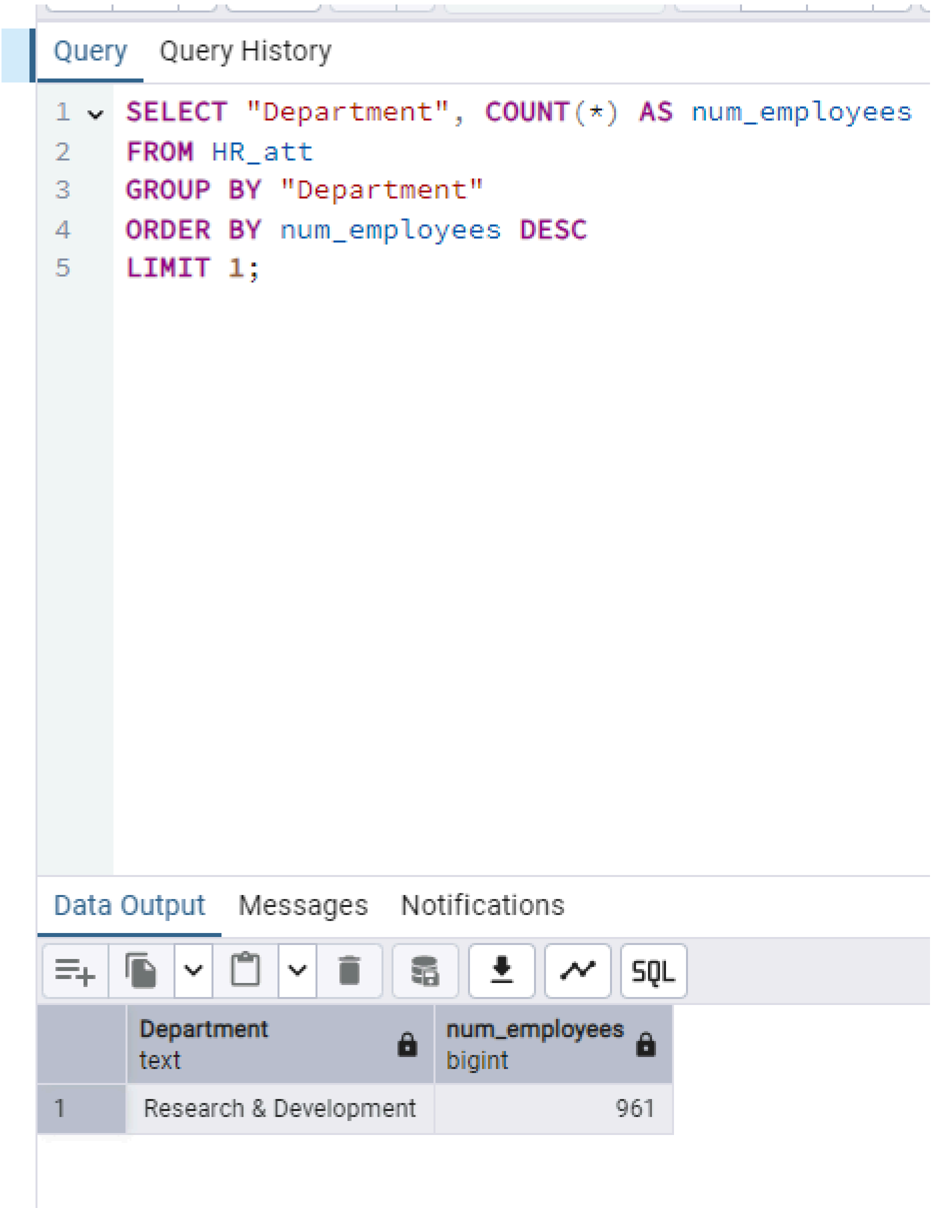
```
#Most popular department

most_popular_department = spark.sql("""
    SELECT Department, COUNT(*) AS count
    FROM attrition_spark_view
    GROUP BY Department
    ORDER BY count DESC
    LIMIT 1
""")
most_popular_department.show()
```

[33] ✓ 1.3s

```
...  +--------------------+-----+
     |          Department|count|
     +--------------------+-----+
     |Research & Develo...|  961|
     +--------------------+-----+
```

Awarding Exercise 6

AO4.1



Median distance from Home

The median distance from home is : 7

Awarding Exercise 6 AO4.1

```
#Finding median distance from home

dist_home = spark.sql("""

    SELECT percentile_approx(distance_from_home, 0.5) AS median_dist
    FROM attrition_spark_view

""")
dist_home.show()
```

[47] ✓ 1.8s

```
+-----------+
|median_dist|
+-----------+
|          7|
+-----------+
```

Most Common level of education

The most common level of education is: Life Sciences

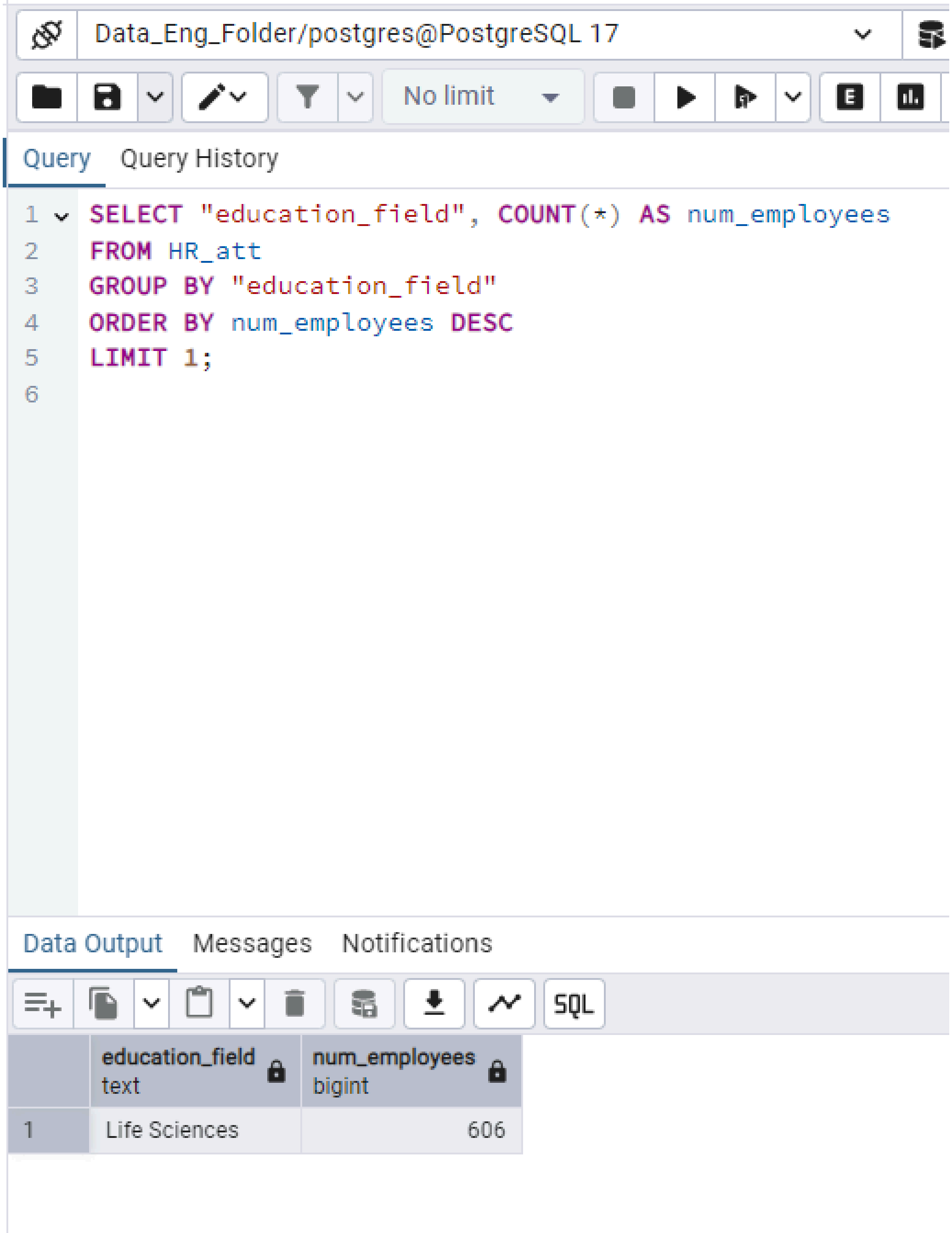
```
# most common education

most_common_education = spark.sql("""

    SELECT education_field, COUNT(*) AS count
    FROM attrition_spark_view
    GROUP BY education_field
    ORDER BY count DESC
    LIMIT 1
""")
most_common_education.show()
```

[34] ✓ 1.8s

```
+---------------+-----+
|education_field|count|
+---------------+-----+
|  Life Sciences|  606|
+---------------+-----+
```

Awarding Exercise 6

AO4.1



I confirm assignments not specified as collaborative are all my own work and do not include any work completed by anyone other than myself.	
Signature	Aldo Mema