Report

(This table is to help locate where each phase is located in our project folder)

Phase	**Where it is located in our project**
Extraction	**Report File - Data Extraction Research:** DB2Asg3Report.docx
Transforming & Data Quality	**Scripts:** A3.ipynb **Analysis - Report File - Data Inspection, Exception Reporting and Data Cleansing Report:** DB2Asg3Report.docx
Data Mining	**Descriptive:** descriptive_a3.ipynb **Predictive:** Predictive_a3.ipynb **Analysis - Report File - Descriptive and Predictive Models:** DB2Asg3Report.docx
Report	**Report File:** DB2Asg3Report.docx

Data Extraction Research

MySQL

2 Different ways to extract data in mysql to a .csv format

Source: https://hevodata.com/learn/mysql-export-to-csv/

1. Using the command line:

Step 1: Go to the table that you would like to export

'USE your_db_name'

Step 2: Select all the data from the table and add the location of the output file. Make sure to use csv extension for the output file.

```
TABLE tableName
INTO OUTFILE 'path/outputFile.csv'
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
ESCAPED BY ''
LINES TERMINATED BY 'n';
```

How to export tables with a timestamp

```
SET @TS = DATE_FORMAT(NOW(),'_%Y_%m_%d_%H_%i_%s');
SET @FOLDER = '/var/lib/sql-files/';
SET @PREFIX = 'employees';
SET @EXT    = '.csv';
SET @CMD = CONCAT("SELECT * FROM tableName INTO OUTFILE
'",@FOLDER,@PREFIX,@TS,@EXT,
"' FIELDS ENCLOSED BY '"
' TERMINATED BY ','
ESCAPED BY '"'",
"LINES TERMINATED BY 'n';");
PREPARE statement FROM @CMD;
EXECUTE statement;
```

Note: the 'INTO OUTFILE' can only be used to export data to a file on the server where MySQL database is running.

How to export with column headers

```
(SELECT 'columnHeading', ...)
UNION
(SELECT column, ...
FROM tableName
INTO OUTFILE 'path-to-file/outputFile.csv''
FIELDS ENCLOSED BY '"'
TERMINATED BY ','
ESCAPED BY '"'
LINES TERMINATED BY 'n')
```

How to handle NULL values

If Null values are not handled, it will appear as ‘N’ on the spreadsheet. To avoid this, run this command:

```
SELECT column, column, IFNULL(column, 'NA')
FROM tableName INTO OUTFILE 'path-to-file/outputFile.csv'
FIELDS ENCLOSED BY '"'
TERMINATED BY ','
ESCAPED BY '"'
LINES TERMINATED BY 'n');
```

This will add the string ‘NA’ to null values.

Important things to consider when using the command line:

File Permissions and Overwriting: Verify that MySQL server has write permissions to the directory when CSV will be created.

Handling Large Datasets: Consider using ‘--quick’ and ‘--compress’ options in the ‘mysql’ command to reduce memory usage and network bandwidth

Escaping Special Characters: If your data contains special characters such as double quotes, you can using the ‘ESCAPED BY’ option in your query.

2. Using mysqldump:

Mysqldump is a tool that is provided by MySQL server and allows users to export their databases and tables to a specified format using a GUI. It can also be used for backup and recovery purposes.

When exporting MySQL data to CSV format, all that has to be done is to add the following command in a command terminal:

```
mysqldump -u [username] -p -t -T/path/to/directory [database] [tableName]
--fields-terminated-by=,
```

This will create a copy of the specified table with the file name being the same as the table name but will be a .txt file. The .txt file created will be located in the path where you defined it in the -T section.

MongoDB

To extract data from a MongoDB database system into a .csv format, there are several methods available. Below are two common approaches along with their respective sources:

Using mongoexport Command Line Tool:

Source: https://www.mongodb.com/docs/database-tools/mongoexport/
MongoDB provides a utility called mongoexport that allows users to export data from a collection to various formats, including CSV. Here's a summary of how to use it:

```
mongoexport --db your_database --collection your_collection --type=csv --fields field1,field2,...
--out output.csv
```

Replace your_database with the name of your MongoDB database.
Replace your_collection with the name of your MongoDB collection.
Replace field1,field2,... with the fields you want to export.
Specify the output file name with --out.

Using Programming Language Libraries (e.g., Python with PyMongo):

Source: https://pymongo.readthedocs.io/en/stable/tutorial.html
You can use libraries like PyMongo (Python driver for MongoDB) to connect to a MongoDB database and extract data programmatically. Then, you can manipulate the data and save it to a CSV file. Below is a simplified example:

Python

```
from pymongo import MongoClient
import csv
```

```
# Connect to MongoDB
client = MongoClient('mongodb://localhost:27017/')

# Access the database and collection
db = client['your_database']
collection = db['your_collection']

# Query data from collection
cursor = collection.find({}, {'_id': False})

# Write data to CSV file
with open('output.csv', 'w', newline='') as csvfile:
    fieldnames = ['field1', 'field2', ...]
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()
    for document in cursor:
        writer.writerow(document)
```

Replace 'your_database' and 'your_collection' with the actual database and collection names.
Specify the fields you want to export in the fieldnames list.
These methods provide efficient ways to extract data from MongoDB into CSV format, depending on your specific requirements and preferences.

Data Inspection, Exception Reporting and Data Cleansing Report

Department ID Uniqueness Exceptions:

Department ID Uniqueness

	index	Department_ID	Department_Name	DOE
0	1	IDEPT5528	Biosciences and Bioengineering	6/28/1943
1	11	IDEPT1825	Mechanical Engineering	9/21/1971
2	15	IDEPT3868	Center for Learning and Teaching (PPCCLT)	3/26/1982
3	21	IDEPT5528	Sanitation and Digital Gaming	None
4	24	IDEPT7005	Centre of Studies in Resources Engineering (CSRE)	8/22/1966
5	25	IDEPT7005	Centre of Studies in Craft Engineering (CSCE)	8/22/1966
6	27	IDEPT9009	Centre for the Study of Ecology in Mars	7/9/2025
7	35	IDEPT3868	Center for Learning and Teaching (PPCCLT)	3/26/1982
8	39	IDEPT9009	Laser Technology Enhancements	None
9	45	IDEPT1825	Materials Strength Testing	9/21/1971

Department Name Uniqueness Exceptions:

Department ID Name

	index	Department_ID	Department_Name	DOE
0	15	IDEPT3868	Center for Learning and Teaching (PPCCLT)	3/26/1982
1	35	IDEPT3868	Center for Learning and Teaching (PPCCLT)	3/26/1982

DOE year less than 1900:

```
...   Department DOE Validation
         index Department_ID           Department_Name        DOE
      0     31     IDEPT1677  ABC-EDS Research Academy  7/10/1849
```

Department Missing Values:

```
...   Department Missing Values
         index Department_ID                Department_Name   DOE
      0     21     IDEPT5528  Sanitation and Digital Gaming  None
      1     39     IDEPT9009  Laser Technology Enhancements  None
```

Student Counseling Department Admission Missing Values:

```
...   Student Counceling Information Missing Values
         index   Student_ID       DOA        DOB Department_Choices  \
      0    298  SID20135073  7/1/2013  12/7/1995               None

        Department_Admission
      0                 None
```

Student Counseling Department Admission Does not Exists:

```
...   Student Counceling Information Missing Values
         index   Student_ID       DOA        DOB Department_Choices  \
      0    298  SID20135073  7/1/2013  12/7/1995               None

        Department_Admission
      0                 None
```

Student and Paper Exception:

```
  209607  209609  SID20189989        Sem_8  SEMI0083259    Paper 6   73.0
  209608  209608  SID20189989        Sem_8  SEMI0086600    Paper 6   87.0
```

Cleaned Data

Marks not in 0-100 range:

```
...   marks
       index   Student_ID Semster_Name     Paper_ID Paper_Name  Marks  \
    0    328  SID20131189        Sem_1  SEMI0015910    Paper 4  -49.0
    1    551  SID20131231        Sem_1  SEMI0016208    Paper 5 -100.0
    2 172218  SID20179280        Sem_4  SEMI0044518    Paper 6    NaN
    3 209593  SID20189989        Sem_6  SEMI0064181    Paper 4    NaN

       Effort_Hours
    0           0.0
    1          14.0
    2           NaN
    3           6.0
```

Effort Hours not in min and max range:

```
...   hours
       index   Student_ID Semster_Name     Paper_ID Paper_Name  Marks  \
    0  59635  SID20147406        Sem_6  SEMI0067259    Paper 2   78.0

       Effort_Hours
    0          -3.0
```

Missing Values:

```
...   Missing Values
       index   Student_ID Semster_Name     Paper_ID Paper_Name  Marks  \
    0    125  SID20131171        Sem_3  SEMI0031818       None   87.0
    1 172218  SID20179280        Sem_4  SEMI0044518    Paper 6    NaN
    2 209593  SID20189989        Sem_6  SEMI0064181    Paper 4    NaN

       Effort_Hours
    0          11.0
    1           NaN
    2           6.0
```

To clean the data for interpreting and analysis, we first wrote a query that returned marks between 0-100 and no null values, then we wrote a query that returned effort hours with positive integers and no decimals (just in case). The final query returned data with no null values in any of the columns. Finally, we combined all the queries into one which gave us the clean dataset.

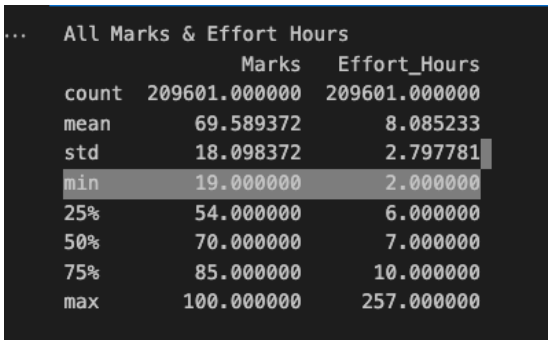
Clean Dataset:

Data cleaning query

index	Student_ID	Semster_Name	Paper_ID	Paper_Name	Marks	Effort_Hours
0	SID20131143	Sem_1	SEMI0012995	Paper 1	44.0	5.0
1	SID20131143	Sem_1	SEMI0015183	Paper 2	74.0	8.0
2	SID20131143	Sem_1	SEMI0018371	Paper 3	80.0	8.0
3	SID20131143	Sem_1	SEMI0015910	Paper 4	44.0	5.0
4	SID20131143	Sem_1	SEMI0016208	Paper 5	95.0	12.0
5	SID20131143	Sem_1	SEMI0017431	Paper 6	61.0	7.0
6	SID20131143	Sem_1	SEMI0014130	Paper 7	90.0	11.0
7	SID20131143	Sem_2	SEMI0024747	Paper 1	92.0	12.0
8	SID20131143	Sem_2	SEMI0025909	Paper 2	57.0	6.0
9	SID20131143	Sem_2	SEMI0022443	Paper 3	91.0	12.0
10	SID20131143	Sem_2	SEMI0025077	Paper 4	84.0	10.0
11	SID20131143	Sem_2	SEMI0029604	Paper 5	80.0	8.0
12	SID20131143	Sem_2	SEMI0029061	Paper 6	66.0	7.0
13	SID20131143	Sem_2	SEMI0022256	Paper 7	54.0	6.0
14	SID20131143	Sem_3	SEMI0037924	Paper 1	76.0	8.0
15	SID20131143	Sem_3	SEMI0034580	Paper 2	83.0	10.0
16	SID20131143	Sem_3	SEMI0033576	Paper 3	41.0	5.0
17	SID20131143	Sem_3	SEMI0031818	Paper 4	80.0	8.0
18	SID20131143	Sem_3	SEMI0039951	Paper 5	69.0	7.0
19	SID20131143	Sem_3	SEMI0039037	Paper 6	73.0	8.0
20	SID20131143	Sem_3	SEMI0035955	Paper 7	66.0	7.0
21	SID20131143	Sem_4	SEMI0044637	Paper 1	78.0	8.0
22	SID20131143	Sem_4	SEMI0048491	Paper 2	83.0	10.0
...						
209606	SID20189989	Sem_8	SEMI0088030	Paper 4	49.0	5.0
209607	SID20189989	Sem_8	SEMI0081794	Paper 5	47.0	257.0
209608	SID20189989	Sem_8	SEMI0086600	Paper 6	87.0	11.0
209609	SID20189989	Sem_8	SEMI0083259	Paper 6	73.0	8.0

Descriptive and Predictive Models

Descriptive:



All Marks & Effort Hours

	Marks	Effort_Hours
count	209601.000000	209601.000000
mean	69.589372	8.085233
std	18.098372	2.797781
min	19.000000	2.000000
25%	54.000000	6.000000
50%	70.000000	7.000000
75%	85.000000	10.000000
max	100.000000	257.000000

When looking at the results based on the descriptive statistics:

1) **Amount of Data:**
    a) After going through the Data Inspection, exception, and data cleansing, this dataset contains 209, 601 records.
2) **Marks Column**
    a) Mean: The average score across all 209, 601 students is 69.6%
    b) Standard Deviation: There is a good amount of variation in the marks based on the STD being 18.1
    c) Min: The lowest mark received is 19%
    d) Quartiles:
        i) 25% of students scored **below** 54%
        ii) 25% of students scored **above** 85%
        iii) 50% of students scored **above** 70%
        iv) 50% of students scored **below** 70%
    e) Max: Highest score is 100%
3) **Effort Hours Column**
    a) Mean: The average hours of effort is around 8.08 hours
    b) Standard Deviation: There is a good amount of variation in the marks based on the STD being 2.78 hours
    c) Min: The lowest amount of effort hours is 2 hours
    d) Quartiles:
        i) 25% of students spent 6 hours or less
        ii) 50% of students spent 7 hours or more
        iii) 75% of students spent 10 hours or less
    e) Max: Maximum effort recorded is 257 hours

Predictive:

Student	Predicted Score in next paper	Department
SID20131151	81.91546178813809	IDEPT6347
SID20149500	82.2542794737644	IDEPT4308
SID20182516	82.1694259983626	IDEPT3062

When looking at the result based on our predictive model:

1) **Predicted Scores:**
    a) The three students have predicted scores that are quite close to each other from 81.92 to 82.25.
    b) Our analysis based on the predictive model is that these 3 students should receive a high mark if 10 hours of effort is put into writing their paper.
2) **Score Precision:**
    a) Our group decided to keep the decimals as the predicted scores for each student were fairly close to each other.
3) **Different Department:**
    a) Since the students are in different departments and are predicted to get similar scores, we believe that the standards for how papers are graded are fairly similar across the different departments.
    b) The model is general enough to be used across different departments
4) **How this model can better be improved:**
    a) Our group found the predicted score eerily similar to one another raising the question of whether this predictive model was implemented correctly. After testing other students, we came to the conclusion that this model was working. Our analysis on perhaps improving this model is to include more variables other than the Effort Hours and Marks. A new column on the Student Performance Table where it stores how difficult a paper is, can help further analyze the predicted score.