

# **7086CEM COURSEWORK**

**STUDENT ID: 10796778**

**STUDENT NAME: SULEIMAN ADAMU YAKUBU**

**MODULE CODE: 7086CEM**

**MODULE LEADER: DR. RACHID ANANE**

I confirm that all work submitted is my own : YES

## PART A: TOOLS AND MACHINERY

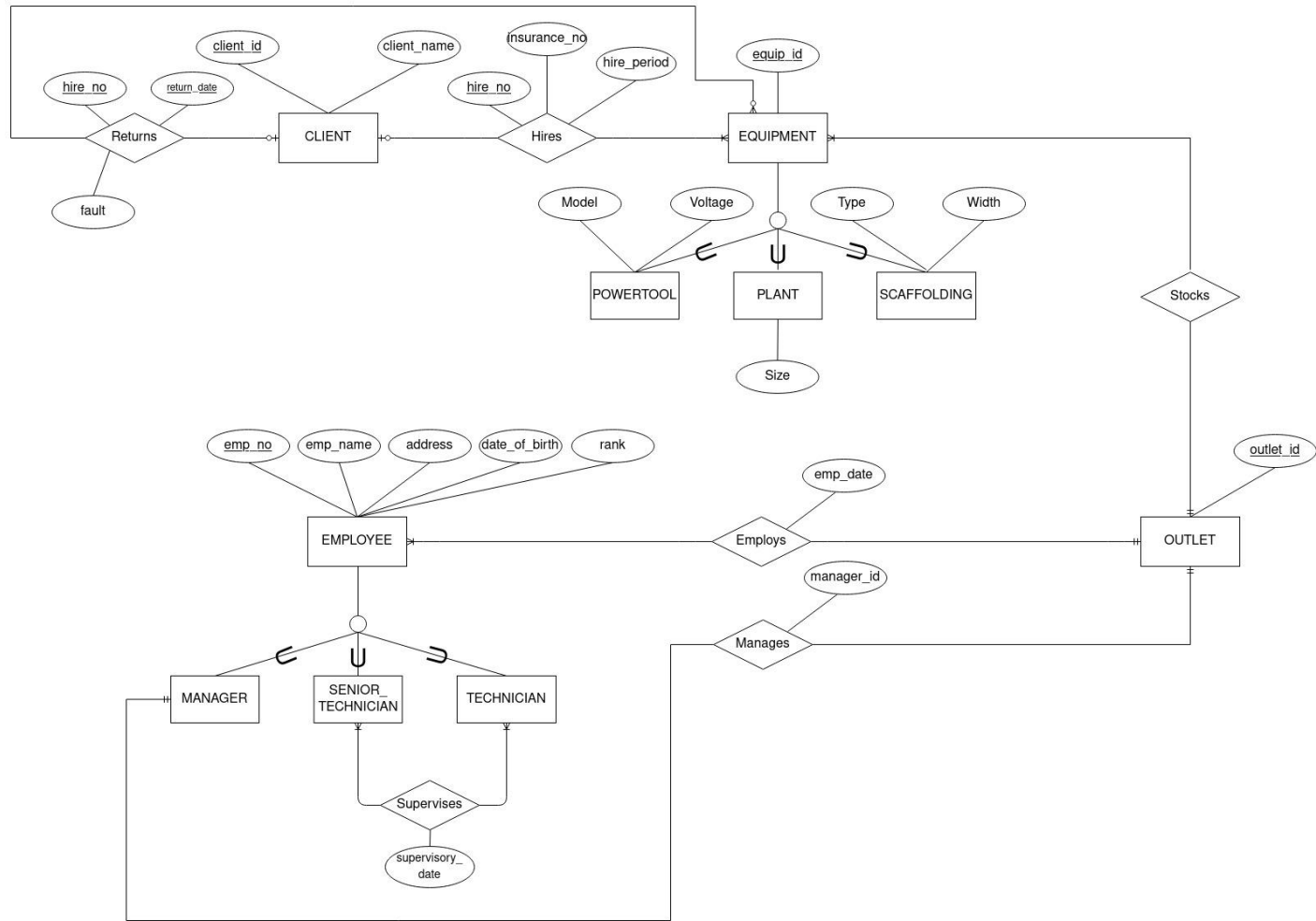


Figure 1: ER Diagram of the scenario

The assumptions made are as follows:

- A manager only manages an outlet and does not supervise technicians.
- A client can hire equipment but may not necessarily return it.
- Managers, Senior Technicians and Technicians are all employees.
- Each outlet is identified by an *outlet\_id*

The schema for the above ER Diagram is as follows:

For clarity, the schema will be split into groups

**Employee:**

Employee (emp\_no, emp\_name, address, date\_of\_birth, \*outlet\_id, rank)

Manager(emp\_no, rank)

Senior\_technician(emp\_no, rank)

Technician(emp\_no, rank)

Supervise(supervisor\_id, supervisee\_id, date)

Employment\_record (emp\_no, outlet\_id, date\_hired)

**Equipment:**

Equipment (equip\_id, tool\_type, \*outlet\_id)

**N.B: The outlet id is added as a foreign key because of the relationship between equipment and outlet**

Powertool (equip\_id, model, voltage)

Plant (equip\_id, model, size)

Scaffolding(equip\_id, type, width)

Hire(hire\_id, client\_no, equip\_id, date, insurance\_no, hire\_period)

Return(hire\_id, return\_date, faults)

**Client:**

Client(clent\_no, client\_name)

**Outlet:**

Outlet(outlet\_id, manager\_id)

## PART B: SQL PROGRAMMING

1. The tables were created using the following SQL commands:

### Author Table

```
CREATE TABLE Author (  
  AuthorID VARCHAR2(4) PRIMARY KEY,  
  Name VARCHAR2(20) NOT NULL  
);
```

### Category Table

```
CREATE TABLE Category (  
  CategoryID VARCHAR2(4) PRIMARY KEY,  
  Type VARCHAR2(20) NOT NULL  
);
```

### Publication Table

```
CREATE TABLE Publication (  
  PubID VARCHAR2(4) PRIMARY KEY,  
  AuthorID VARCHAR2(4) REFERENCES Author,  
  Title VARCHAR2(50) NOT NULL,  
  CatID VARCHAR2(4) REFERENCES Category,  
  PublishedYear INT NOT NULL,  
  Availabilty VARCHAR2(3) NOT NULL  
);
```

### DBUser Table

```
CREATE TABLE DBUser (  
  UserID VARCHAR2(4) PRIMARY KEY,  
  Name VARCHAR2(20) NOT NULL,  
  Email VARCHAR2(25) NOT NULL,  
  Password VARCHAR2(20) NOT NULL  
);
```

## Request Table

```
CREATE TABLE Request (  
  UserID VARCHAR2(4) REFERENCES DBUser,  
  PublicationID VARCHAR2(4) REFERENCES Publication,  
  RequestDate DATE NOT NULL,  
  PRIMARY KEY (UserID, PublicationID, RequestDate)  
);
```

DBUser table was created before the Request table to satisfy Referential Integrity constraints. For simplicity, the commands to create the tables was run together, the output generated is in the figure below:

Table created.

Table created.


Table created.

Table created.

Table created.

*Figure 2: Output of running the create table commands*

The User table as defined in the brief was renamed to DBUser because of naming problems. Oracle SQL does not allow tables to be named User. This is shown in the following figure:



ORA-00903: invalid table name

© 2022 Oracle · Live SQL 22.1.2, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 ·  
Built with ❤️ using Oracle APEX · Privacy · Terms of Use

*Figure 3: Error message when a table is created with the name User*

The tables were then populated with the required values. This is shown below:

#### **Author Table**

```
INSERT INTO Author VALUES('A011', 'Dingle R');  
INSERT INTO Author VALUES('A012', 'Ransom A');  
INSERT INTO Author VALUES('A013', 'Wardale R');  
INSERT INTO Author VALUES('A014', 'Alexander T');  
INSERT INTO Author VALUES('A015', 'Spurrier S');
```

AUTHORID	NAME
A011	Dingle R
A012	Ransom A
A013	Wardale R
A014	Alexander T
A015	Spurrier S

Download CSV  
5 rows selected.

Figure 4: A look at the inserted values in the Author table

## Category Table

```
INSERT INTO Category VALUES('C911', 'Short stories');
INSERT INTO Category VALUES('C912', 'Journal articles');
INSERT INTO Category VALUES('C913', 'Biography');
INSERT INTO Category VALUES('C914', 'Illustrations');
```

CATEGORYID	TYPE
C911	Short stories
C912	Journal articles
C913	Biography
C914	Illustrations

Download CSV  
4 rows selected.

Figure 5: A look at the inserted values in the Category table

## Publication Table

```
INSERT INTO Publication VALUES('P001', 'A011', 'The Blue Treacle', 'C911', 1911, 'No');
INSERT INTO Publication VALUES('P002', 'A012', 'In Aleppo Once', 'C911', 2001, 'Yes');
INSERT INTO Publication VALUES('P003', 'A012', 'Illustrating Arthur Ransome', 'C914', 1973, 'Yes');
INSERT INTO Publication VALUES('P004', 'A012', 'Ransome the Artist', 'C914', 1994, 'Yes');
INSERT INTO Publication VALUES('P005', 'A014', 'Bohemia in London', 'C912', 2008, 'No');
INSERT INTO Publication VALUES('P006', 'A011', 'The Best of Childhood', 'C911', 2002, 'Yes');
INSERT INTO Publication VALUES('P007', 'A015', 'Distilled Enthusiasms', 'C912', 201, 'Yes');
```

PUBID	AUTHORID	TITLE	CATID	PUBLISHEDYEAR	AVAILABILTY
P001	A011	The Blue Treacle	C911	1911	No
P002	A012	In Aleppo Once	C911	2001	Yes
P004	A012	Ransome the Artist	C914	1994	Yes
P005	A014	Bohemia in London	C912	2008	No
P003	A012	Illustrating Arthur Ransome	C914	1973	Yes
P006	A011	The Best of Childhood	C911	2002	Yes
P007	A015	Distilled Enthusiasms	C912	201	Yes

[Download CSV](#)

7 rows selected.

*Figure 6: A look at the inserted values in the Publication table*



## DBUser Table

```
INSERT INTO DBUser VALUES ('U111', 'Kenderine J', 'KenderineJ@hotmail.com', 'Kenj2');
INSERT INTO DBUser VALUES ('U241', 'Wang F', 'WangF@hotmail.com', 'Wanf05');
INSERT INTO DBUser VALUES ('U55', 'Flavel K', 'FlavelK@hotmail.com', 'Flak77');
INSERT INTO DBUser VALUES ('U016', 'Zidane Z', 'ZidaneZ@hotmail.com', 'Zidz13');
INSERT INTO DBUser VALUES ('U121', 'Keita R', 'KeitaR@hotmail.com', 'Keir22');
```

USERID	NAME	EMAIL	PASSWORD
U111	Kenderine J	KenderineJ@hotmail.com	Kenj2
U241	Wang F	WangF@hotmail.com	Wanf05
U55	Flavel K	FlavelK@hotmail.com	Flak77
U016	Zidane Z	ZidaneZ@hotmail.com	Zidz13
U121	Keita R	KeitaR@hotmail.com	Keir22

[Download CSV](#)

5 rows selected.

Figure 7: A look at the inserted values in the DBUser table

## Request Table

```
INSERT INTO Request VALUES ('U016', 'P001', '05-Oct-17');
INSERT INTO Request VALUES ('U241', 'P001', '28-Sep-17');
INSERT INTO Request VALUES ('U55', 'P002', '08-Sep-17');
INSERT INTO Request VALUES ('U016', 'P004', '06-Oct-17');
INSERT INTO Request VALUES ('U121', 'P002', '23-Sep-17');
```

USERID	PUBLICATIONID	REQUESTDATE
U016	P001	05-OCT-17
U016	P004	06-OCT-17
U121	P002	23-SEP-17
U241	P001	28-SEP-17
U55	P002	08-SEP-17

Download CSV  
5 rows selected.

*Figure 8: A look at the inserted values in the Request table*

2. The SQL statements to generate the required results are as follows
  - a. The names and email addresses of users who requested publications of the type “Illustrations”.

```
SELECT Name, Email FROM DBUser WHERE UserID IN
  (SELECT UserID FROM Request WHERE Request.PublicationID IN
    (SELECT PubID from Publication WHERE Publication.CatID IN
      (SELECT CategoryID FROM Category WHERE
Type='Illustrations')));
```

The output is displayed below:

NAME	EMAIL
Zidane Z	ZidaneZ@hotmail.com

Download CSV

*Figure 9: A look at the output of the above query*

- b. Details of publications that were requested during September 2017 in descending order of requested dates.

```
SELECT PubId, Title, PublicationID, RequestDate FROM
Publication, Request
WHERE PubID= PublicationId AND RequestDate BETWEEN '01-Sep-17'
AND '30-Sep-2017'
ORDER BY RequestDate DESC;
```

The output of the query is displayed below:

PUBID	TITLE	PUBLICATIONID	REQUESTDATE
P001	The Blue Treacle	P001	28-SEP-17
P002	In Aleppo Once	P002	23-SEP-17
P002	In Aleppo Once	P002	08-SEP-17

[Download CSV](#)  
3 rows selected.

*Figure 10: A look at the query result*

- c. The names of users who requested more than one publication

```
SELECT Name FROM DBUser WHERE UserID IN
(SELECT UserID FROM Request
HAVING COUNT(UserID) > 1
GROUP BY UserID);
```

NAME
Zidane Z

[Download CSV](#)

*Figure 11: A look at the result of the above query*

- d. The number of publications requested for each of the categories

```
SELECT Publication.CatID, COUNT(Publication.PubID)
FROM Request INNER JOIN Publication
ON Publication.PubID = Request.PublicationID
GROUP BY Publication.CatID;
```

CATID	COUNT(PUBLICATION.PUBID)
C911	4
C914	1

[Download CSV](#)  
2 rows selected.

*Figure 12: A look at the result of the above query*

## PART C: SEQUENTIAL AND PARALLEL PROCESSING

1. Given the schema, a table called FlightInfo was created.

```
CREATE TABLE FlightInfo (  
Year INTEGER NOT NULL,  
Month INTEGER NOT NULL,  
DayOfMonth INTEGER NOT NULL,  
WeekDay INTEGER NOT NULL,  
DepartureTime DATE NOT NULL,  
ActDepartureTime DATE NOT NULL,  
ArrivalTime DATE NOT NULL,  
Carrier VARCHAR2(5),  
FlightNo VARCHAR2(6) PRIMARY KEY,  
DepartureDelay INTEGER NOT NULL,  
ArrivalDelay INTEGER NOT NULL,  
Cancellation VARCHAR2(3) DEFAULT 'No' NOT NULL,  
WeatherDelay INTEGER NOT NULL  
);
```

The table is created with the following assumptions:

- The primary key of the table is the flight number as each flight must have a unique number to identify it
- A carrier can have several flights under it and all flights must belong to one and only one carrier. This means the carrier is not created with a unique constraint.

**NB:** If the carrier is UNIQUE, then that means that each record will have a different carrier, thus no carrier can have more than one delayed flight

With the above assumptions, the following records were created.

Records:

```
{year: "1999", month: "5", day_of_month: "21", day_of_week: "3", departure_time:  
"0015", actual_departure_time: "0014", arrival_time: "0115", carrier: "001",  
flight_number: "BA123", departure_delay: "1", arrival_delay: "0", cancellation: "No",  
weather_delay: "0" }
```

{year: "2001", month: "7", day\_of\_month: "01", day\_of\_week: "2", departure\_time: "1200", actual\_departure\_time: "1200", arrival\_time: "1600 ", carrier: "013", flight\_number: "EA456", departure\_delay: "0", arrival\_delay: "5", cancellation: "No", weather\_delay: "0" }

{year: "2015", month: "12", day\_of\_month: "25", day\_of\_week: "7", departure\_time: "1312", actual\_departure\_time: "1300", arrival\_time: "1700 ", carrier: "045", flight\_number: "GA123", departure\_delay: "12", arrival\_delay: "0", cancellation: "No", weather\_delay: "0" }

{year: "2000", month: "3", day\_of\_month: "4", day\_of\_week: "3", departure\_time: "1400", actual\_departure\_time: "1400", arrival\_time: "1500 ", carrier: "001", flight\_number: "BA125", departure\_delay: "0", arrival\_delay: "5", cancellation: "No", weather\_delay: "0" }

{year: "2003", month: "10", day\_of\_month: "17", day\_of\_week: "6", departure\_time: "1800", actual\_departure\_time: "1800", arrival\_time: "2000 ", carrier: "007", flight\_number: "NA321", departure\_delay: "0", arrival\_delay: "0", cancellation: "No", weather\_delay: "0" }

{year: "2005", month: "6", day\_of\_month: "6", day\_of\_week: "7", departure\_time: "0400", actual\_departure\_time: "0400", arrival\_time: "0430 ", carrier: "001", flight\_number: "BA126", departure\_delay: "0", arrival\_delay: "5", cancellation: "No", weather\_delay: "0" }

{year: "2009", month: "2", day\_of\_month: "14", day\_of\_week: "5", departure\_time: "0015", actual\_departure\_time: "0000", arrival\_time: "0100 ", carrier: "013", flight\_number: "EA876", departure\_delay: "15", arrival\_delay: "15", cancellation: "No", weather\_delay: "0" }

The query to retrieve the delayed flights is as follows:

```
SELECT Carrier, COUNT(FlightNo) FROM FlightInfo
WHERE DepartureDelay > 0 OR ArrivalDelay > 0
GROUP BY Carrier;
```

The records that satisfy the query are then inserted into the mapreduce graphic. The records were abbreviated and only the relevant attributes shown because of size constraints.

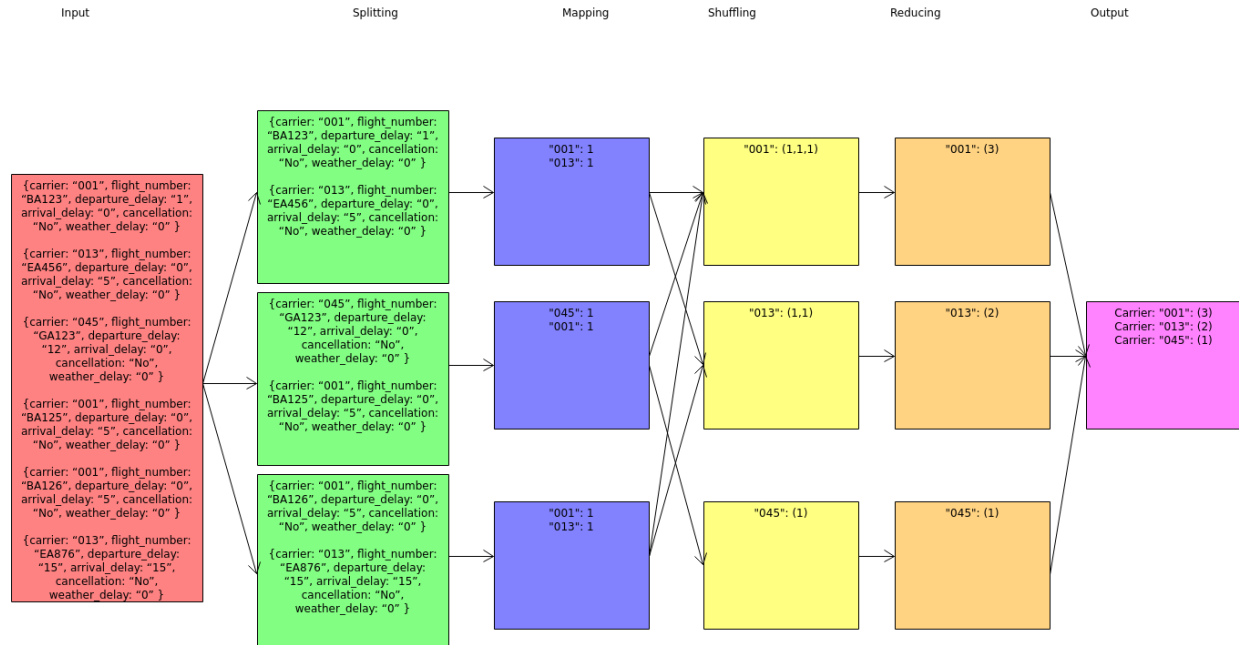


Figure 12: The mapreduce procedure

*Working assumptions: It was assumed that carriers could have multiple flights. So a flight can be identified by the carrier, and flight number. This assumption led us to use the carrier as the key for the Mapreduce operation.*

For the mapreduce procedure, the input file contains all the flights that experienced delays. This file has 6 records. For distributed processing, the file was split into three separate data blocks containing 2 records per block.

The next step in the procedure is the map step. In this step we convert the records to key, value pairs. The key selected for this operation is the carrier as we need to find how many flights were delayed per carrier. The data is stored as key, value pairs with the key being the carrier and the value being the number of times that carrier was delayed in that particular data block. Hence the result of the map function is a list with the carrier, and the number of times it appeared in the data block. The result of the map function on the data blocks is as follows:

- 1st block: ("001", 3) and ("013", 1)
- 2nd block: ("045", 1) and ("001", 1)
- 3rd block: ("001", 1) and ("013", 1)

The shuffle function then groups all the values with the same key together in the same data block. After the shuffle, the data blocks then contain the following lists:

*1st block:* ("001", (1,1,1))

*2nd block:* ("013", (1,1))

*3rd block:* ("045", (1))

The reduce function then sums all the values associated with a single key. The output in the data blocks after the reduce function is the following list:

*1st block:* ("001", 3)

*2nd block:* ("013", 2)

*3rd block:* ("045", 1)

The data blocks are then collected into one file. This file has the count of delayed flights for each carrier.

("001", 3), ("013", 2), ("045", 1)

So, carrier "001" had 3 delayed files, "013" had 2 delayed flights, and "045" had just one delayed flight.



# Data Warehouse

In today's world, data is a valuable resource. Businesses, organisations, or governments can use data to monitor performance, derive insights, and drive decision making. This process makes use of Business Intelligence tools, SQL clients, and a host of other analytical tools, (Amazon.com ) n.d.).

Typically, data is a collection of facts and information about a specific domain of discourse. This data can be quantitative or qualitative in nature. Traditionally, this data is stored in a database, which is an organised collection of data stored in a computer system(Oracle Corporation ) n.d.). In a corporation, a database will usually store transactional information on the day-to-day operations of the corporation. This kind of database is usually referred to as an OLTP (Online Transaction Processing) database (Cardon 2018). This type of data storage is usually optimised for storing data and is not so efficient at analysing it. This is where a data warehouse comes in (Cardon 2018).

A data warehouse is simply a store where an entity (a business, or company) collects all kinds of data typically for analytical purposes. Data such as sales data, employee data, and customer data is aggregated in the data warehouse. This means that the entity does not have to scan around various sources while looking for any data as it is all aggregated in one place. The data warehouse serves as a single point of truth for the company (365 Data Science 2020).

A data warehouse is an OLAP (Online Analytical Processing) data store. It is conceptualised as a layer above one or more OLTP (or even other) databases and aggregates data from them for analysis. The data warehouse then serves as a source for analytics, dashboards, and reporting.

## Architecture of a Data Warehouse

A data warehouse is typically made up of tiers. The top tier is client-facing and is typically used to present results through reporting, analysis, and data mining. The middle tier concerns itself with the analytics. It is made up of the analytical engine that can access and analyse the data. The bottom tier is the data itself. This is the database server where the data is loaded and stored (Amazon.com ) n.d.)

The data warehouse stores data in two ways; data accessed frequently is stored in fast, typically expensive storage such as a Solid State Drive (SSD), while data that is not accessed as frequently lives in cheaper storage (Amazon Web Services n.d.).

## How Data is Stored in a Data Warehouse

The data entered a data warehouse undergoes some processing to make it suitable. This process are termed ETL (Extraction, Transformation, Loading) and is made up of three (3) sub-processes (Fuad 2023: 9):

1. Extraction: the data is extracted from the sources here
2. Transformation: the data is transformed using a series of rules to make it consistent with the data in the warehouse and easier to analyse

Loading: the data is then entered into the data warehouse. This can take place as part of the transformation process or after it.

## Characteristics of a Data Warehouse

For a data store to be considered a data store, it should satisfy certain criteria. These are:

1. It should be subject oriented: The data warehouse contains information around a particular subject and does not contain all company data
2. integrated : a data warehouse usually has multiple sources of information. This means that the format, structure, and characteristic of the data is disparate. The data warehouse creates consistency among the different data from various sources
3. Time-variant: the data warehouse does not just deal with live data, but also historical data. This is because it is used for reporting, forecasting, and analysis.
4. Non-volatile: this means that once data is in a data warehouse, it is stable and cannot be changed or deleted.
5. Summarised: since the data warehouse facilitates reporting and/or analytics, the data is often segmented or aggregated (365 Data Science 2020).

## Benefits of a Data Warehouse

A data warehouse allows joining data from multiple sources for analysis. Thus, it allows an organisation to analyse and extract insights and value from a large amount of variant data (Oracle Corporation n.d.).

## References

- 365 Data Science (2020) *What is a Data Warehouse?* [online] available from  
<[https://www.youtube.com/watch?v=AHR\\_7jFCMeY](https://www.youtube.com/watch?v=AHR_7jFCMeY)> [05/03/2022]
- Amazon.com *What is a Data Warehouse?* [online] available from  
<<https://aws.amazon.com/data-warehouse/>> [05/03/ 2022]
- Cardon, D. (2018) *Database Vs Data Warehouse: A Comparative Review* [online]  
available from  
<https://www.healthcatalyst.com/insights/database-vs-data-warehouse-a-comparative-review/> [05/03/ 2022]
- Fuad, M. (2019) *Lecture 2 -Introduction to Big Data Analytics*. Coventry: Coventry University
- Heller, M. (2021) '*What is a Data Warehouse?* The Source of Business Intelligence'. *InfoWorld.Com* [online] available from  
<<https://www.infoworld.com/article/3629889/what-is-a-data-warehouse-the-source-of-business-intelligence.html>>
- Oracle Corporation (a) *What is a Database?* [online] available from  
<<https://www.oracle.com/uk/database/what-is-database/>> [05/03/ 2022]
- Oracle Corporation (b) *What is a Data Warehouse?* [online] available from  
<<https://www.oracle.com/uk/database/what-is-a-data-warehouse/>> [05/03/ 2022]