multiverse

multiverse

Advanced Skills

Session 1

Session Outline

Data Warehouse

Data Integration

Types of Data Integration

Data Integration Life Cycle

Data Profiling

Data Integration Techniques

Rules and Policies

ETL

Security

Automating the Process

Licenses vs Coding

Master Data Management

Testing Strategies

Recap

Learning Objectives

- **Understand** concepts of Data Integration and ETL techniques
- **Explain** the difference between **Data Integration** and **Data Migration**
- **Explore** different testing strategies for Data Integration

Data Warehouse

'Data Warehousing' is a practice in data management whereby data is copied from various operational systems into a persistant data store in a consistent format to be used for analysis, decision making and reporting.

Online Transactional Processing (OLTP)

OLTP provides transaction orientated applications, administering day to day transcations of an organisation. For example:

Online Transactional Processing (OLTP)

OLTP provides transaction orientated applications, administering day to day transcations of an organisation. For example:

Supermarkets

Online Transactional Processing (OLTP)

OLTP provides transaction orientated applications, administering day to day transcations of an organisation. For example:

Supermarkets

Online banking

Online Transactional Processing (OLTP)

OLTP provides transaction orientated applications, administering day to day transcations of an organisation. For example:

Supermarkets

Online banking

Airline ticket booking

Online Transactional Processing (OLTP)

OLTP provides transaction orientated applications, administering day to day transcations of an organisation. For example:

Supermarkets

Online banking

Airline ticket booking

Adding items to a shopping cart

Online Analytical Processing (OLAP)

OLAP consists of data analytics tools that are used for making business decisions. It provides an environment to leverage insights from multiple database systems at one time. For example:

Online Analytical Processing (OLAP)

OLAP consists of data analytics tools that are used for making business decisions. It provides an environment to leverage insights from multiple database systems at one time. For example:

Recommendation algorithms (e.g. Spotify suggested, Amazon products)

Online Analytical Processing (OLAP)

OLAP consists of data analytics tools that are used for making business decisions. It provides an environment to leverage insights from multiple database systems at one time. For example:

Recommendation algorithms (e.g. Spotify suggested, Amazon products)

Virtual assisstants (e.g. Alexa, Siri)

Online Analytical Processing (OLAP)

OLAP consists of data analytics tools that are used for making business decisions. It provides an environment to leverage insights from multiple database systems at one time. For example:

Recommendation algorithms (e.g. Spotify suggested, Amazon products)

Virtual assisstants (e.g. Alexa, Siri)

Targeted Adverts

Online Analytical Processing (OLAP)

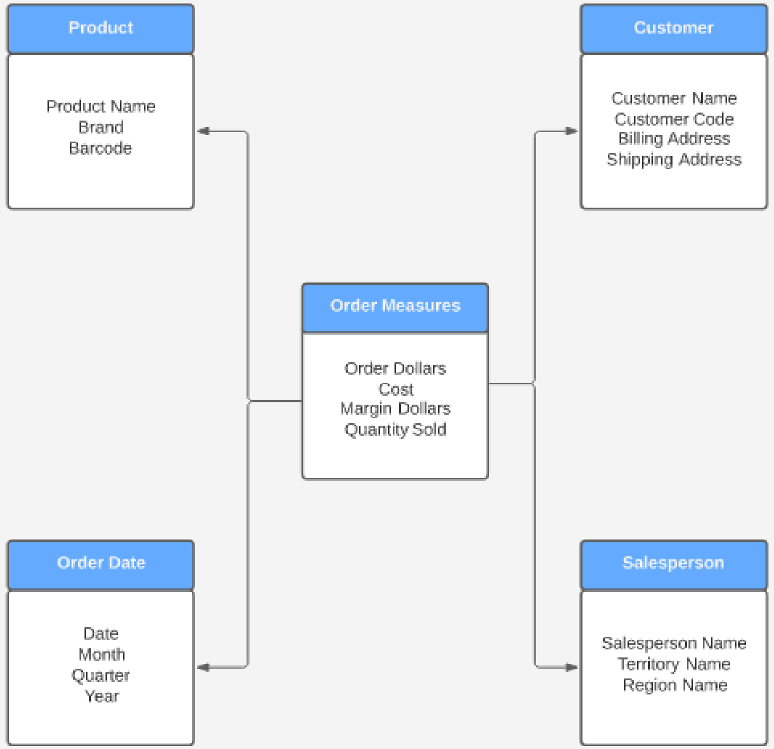
OLAP consists of data analytics tools that are used for making business decisions. It provides an environment to leverage insights from multiple database systems at one time. For example:

Recommendation algorithms (e.g. Spotify suggested, Amazon products)

Virtual assisstants (e.g. Alexa, Siri)

Targeted Adverts

Suggested LinekedIn connections



Product

Product Name
Brand
Barcode

Customer

Customer Name
Customer Code
Billing Address
Shipping Address

Order Measures

Order Dollars
Cost
Margin Dollars
Quantity Sold

Order Date

Date
Month
Quarter
Year

Salesperson

Salesperson Name
Territory Name
Region Name

Data Warehouse

Data Warehouse

VS

Data Warehouse

vs

Database

Data Warehouse vs *Database*

Type of Processing

Data Warehouse vs *Database*

Type of Processing

OLAP

Data Warehouse vs ***Database***

Type of Processing

OLAP

OLTP

Data Warehouse vs *Database*

Data Structure

Data Warehouse vs *Database*

Data Structure

Denormalised table containing repeated data

Data Warehouse vs ***Database***

Data Structure

Denormalised table containing repeated data

Highly normalised with different tables

Data Warehouse vs *Database*

Optimised For

Data Warehouse vs *Database*

Optimised For

Rapid execution of queries on large complex datasets

Data Warehouse vs *Database*

Optimised For

Rapid execution of queries on large complex datasets

Updating, deleting and modifying data

Data Warehouse vs *Database*

Data Timeline

Data Warehouse vs ***Database***

Data Timeline

Historical Data

Data Warehouse vs *Database*

Data Timeline

Historical Data

Current real-time data

Data Warehouse vs *Database*

Uptime

Data Warehouse vs *Database*

Uptime

Regular downtime to allow batch upload

Data Warehouse vs *Database*

Uptime

Regular downtime to allow batch upload

Approx 100%

Data Warehouse vs *Database*

Query Type

Data Warehouse vs *Database*

Query Type

Complex queries for in depth analysis

Data Warehouse vs ***Database***

Query Type

Complex queries for in depth analysis

Simple transactional queries

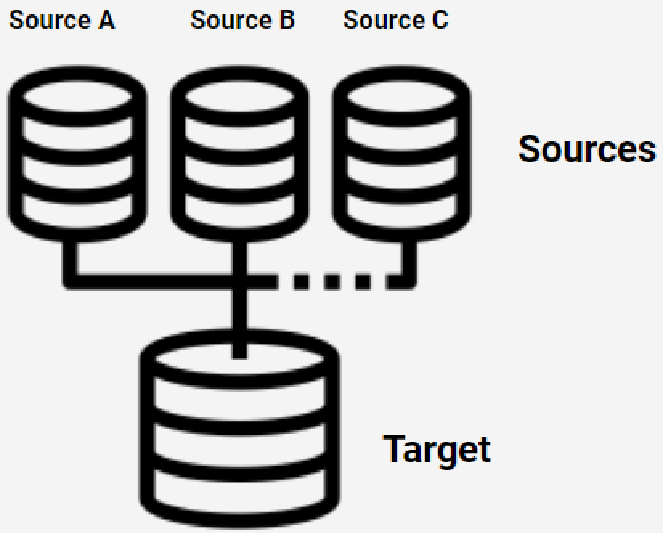
Data Warehouse vs Database

	DATA WAREHOUSE	DATABASE
Processing	OLAP	OLTP
Structure	Denormalised table containing repeated data	Highly normalised with different tables
Optimisation	Rapidly executing low number of complex queries on large multi-dimensional datasets	Updating, deleting and modifying data
Timeline	Historical data	Current real-time data
Uptime (SLA)	Regular downtime to allow batch uploads	Appox. 100% uptime
Query Type	Complex queries for in depth analysis	Simple transactional

Data Warehouses are examples of Data Integration products

Data Integration

Data Integration is the process of collecting data from a variety of sources into a single target.



Data Integration Sources

Data Integration Sources

Data Integration Sources

Text Files

Data Integration Sources

Text Files

Databases

Data Integration Sources

Text Files

Databases

Spreadsheets

Data Integration Sources

Text Files

Databases

Spreadsheets

Applications

Benefits of Data Integration

Benefits of Data Integration

Benefits of Data Integration

Increased availability of data

Benefits of Data Integration

Increased availability of data

Superior data integrity and quality

Benefits of Data Integration

Increased availability of data

Superior data integrity and quality

Collaboration opportunities

Benefits of Data Integration

Increased availability of data

Superior data integrity and quality

Collaboration opportunities

Greater insights and improvements

Benefits of Data Integration

Increased availability of data

Superior data integrity and quality

Collaboration opportunities

Greater insights and improvements

Improved data consistency

Data Integration vs Data Migration

Data Integration vs Data Migration

Data Integration is the process of collecting data from a variety of sources into a unified view for analysis and making data driven business decisions.

Data Integration vs Data Migration

Data Integration is the process of collecting data from a variety of sources into a unified view for analysis and making data driven business decisions.

Data Migration is when the data is simply moved from one source to another.

Data Integration vs Data Migration

Data Integration is the process of collecting data from a variety of sources into a unified view for analysis and making data driven business decisions.

Data Migration is when the data is simply moved from one source to another.

Companies will typically migrate data when implementing a new system or merging to a new environment.

multiverse

Types of Data Integration

Batch

Real-time

Batch

Data transfered from source to target in groups periodically

Real-time

Batch

Data transfered from source to target in groups periodically

Data formats and layouts must be consistent between source and target

Real-time

Batch

Data transfered from source to target in groups periodically

Data formats and layouts must be consistent between source and target

Source and target are **'asynchronus'** (source doesn't wait for target to process data)

Real-time

Batch

Data transfered from source to target in groups periodically

Data formats and layouts must be consistent between source and target

Source and target are **'asynchronus'** (source doesn't wait for target to process data)

Real-time

Data transfered from source to target instantly

Batch

Data transfered from source to target in groups periodically

Data formats and layouts must be consistent between source and target

Source and target are **'asynchronus'** (source doesn't wait for target to process data)

Real-time

Data transfered from source to target instantly

Involved a much smaller amount of data and used when it is necessary to complete a single transaction

Batch

Data transfered from source to target in groups periodically

Data formats and layouts must be consistent between source and target

Source and target are **'asynchronus'** (source doesn't wait for target to process data)

Real-time

Data transfered from source to target instantly

Involved a much smaller amount of data and used when it is necessary to complete a single transaction

Source and target are **'synchronus'** (changes in source are reflected in target)

Data Integration Life Cycle

1. Scoping
2. Profiling
3. Design
4. Testing
5. Implementation

Technical Requirements

Business Requirements

Data Requirements

Operational Requirements

1. Scoping
2. Profiling
3. Design
4. Testing
5. Implementation

Understand our data

- Duplicates
- Null values
- Format
- Data Types
- Values

1. Scoping
2. Profiling
3. Design
4. Testing
5. Implementation

Decide on the architecture of the data warehouse using business, technical and operational metadata

1. Scoping
2. Profiling
3. Design
4. Testing
5. Implementation

Validation and verification of coding interface

Test the process works

User Acceptance Testing (UAT)

Technical Acceptance Testing (TAT)

Performance Stress Testing (PST)

1. Scoping
2. Profiling
3. Design
4. Testing
5. Implementation

Implement the process at an operational level

Data Profiling

Data Profiling is the process of reviewing and analysing data to be used in an extract to understand the format and content.

Uses of Data Profiling

Uses of Data Profiling

Develop metadata and documentation

Uses of Data Profiling

Develop metadata and documentation

Report data formats, uniqueness, consistency, correctness and null values

Uses of Data Profiling

Develop metadata and documentation

Report data formats, uniqueness, consistency, correctness and null values

Compare field names across data stores/tables

Uses of Data Profiling

Develop metadata and documentation

Report data formats, uniqueness, consistency, correctness and null values

Compare field names across data stores/tables

Can be difficult to arrange if it involves personal or sensitive information

DATASET NAME	FORMAT	DATASET TYPE
My_dataset	RDBMS	Reference

Author: Multiverse; Last editied: 01/03/2021

FIELD NAME	DATA TYPE	COUNT	NULL VALUES	% NULLS	MAXIMUM VALUE	MINIMUM VALUE
customer_surname	string	1501	0	0%	zabini	abbots

Activity

- Open the products Jupyter Notebook
- Using pandas, profile the data for:
  - Data Format
  - Field Names
  - Field Data Types
  - Summary Statistics of the data
  - Information on Null values
  - Any other information you think is necessary
- Create a text document to show this information

Data Integration Techniques

Data Integration Techniques

Manual Data Integration

Middleware Data Integration

Application Based Integration

Uniform Access Integration

Common Storage Integration

Manual Data Integration

Whole process (e.g. data collection and cleaning, connecting sources) done manually by a human

Best for one-time instances

Manual Data Integration

Benefits

Reduced Costs

Greater Freedom

Drawbacks

Difficulty Scaling

Greater Room for Error

Less Access

Middleware Data Integration

Using softwares that connect applications and transfers between them and databases (no coding)

Acts an interpreter between systems and enacts an automatic transfer

Examples include Microsoft Dynamic CRM, SAP and Sage

Middleware Data Integration

Benefits

Fast and Efficient

Scalable

Time Saving

Drawbacks

Less Access

Limited Functionality

Uniform Access Integration

Also known as "Virtual Integration"

Data is allowed to stay in its original location when being accessed

Provides a unified view quickly to both customers and across platforms

Uniform Access Integration

Benefits

Simplified View of Data

Easy Access

Lower Storage Requirements

Drawbacks

Data Management can be Difficult

Data Integrity could be Compromised

Common Storage Integration (Data Warehouse)

Similar to uniform access except it creates and stores a copy of the data

One of the most popular integration methods

Common Storage Integration (Data Warehouse)

Benefits

Reduced Burden

Cleaner Data Appearance

Enhanced Data Analytics

Drawbacks

Increased Storage Costs

Higher Maintenance Costs

Rules and Policies

You must specifiy security policies (e.g. who has access?)

Data integrated should be immutable (unchanging)

Validation checks should be carried out during the process

- Validate the source and target table structure and data types
- Validate the column names against a mapping document

Verification is also carried out on the Data Warehouse

- Verify the data is accurate
- Verify the data is correct
- Verify the data has not been duplicated in the Data Warehouse

If you are wanting to use Business Data...

Get Permission from the Data Owner!

Data owners are given the right to decide who can have access to enterprise data.

The process involved may be something like this:

Data owners are given the right to decide who can have access to enterprise data.

The process involved may be something like this:

A person (staff member, contractor, supplier, etc) requests access to information

Data owners are given the right to decide who can have access to enterprise data.

The process involved may be something like this:

A person (staff member, contractor, supplier, etc) requests access to information

A business resource (Data owner, manager) will review the request

Data owners are given the right to decide who can have access to enterprise data.

The process involved may be something like this:

A person (staff member, contractor, supplier, etc) requests access to information

A business resource (Data owner, manager) will review the request

A techinical resources (usually a DBA) physically grants permission to an application, database or other data store containing the data.

Data owners are given the right to decide who can have access to enterprise data.

The process involved may be something like this:

A person (staff member, contractor, supplier, etc) requests access to information

A business resource (Data owner, manager) will review the request

A techinical resources (usually a DBA) physically grants permission to an application, database or other data store containing the data.

Often the permission follows a CRUD schema (create, read, update, delete)

ETL

ETL

E

T

L

Extract

E

T

L

Extract

Transform

E

T

L

Extract

Transform

Load

A process of Data Integration from Multiple Sources

It allows business the ability to gather data from multiple sources and consolidate into a single, centralised location

This can be hard coded or using a licensed product

Extract

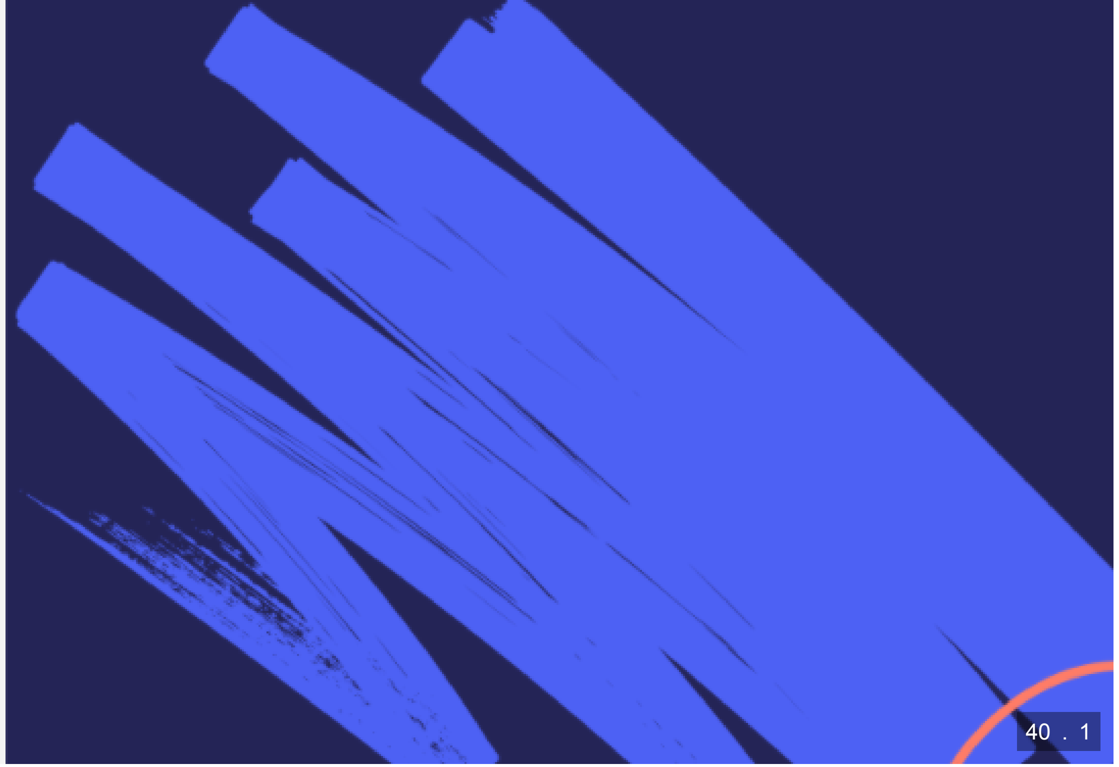
Extract

Data is accessed from the source

Extract

Data is accessed from the source

For this stage to be effective, a basic understanding of the data is required

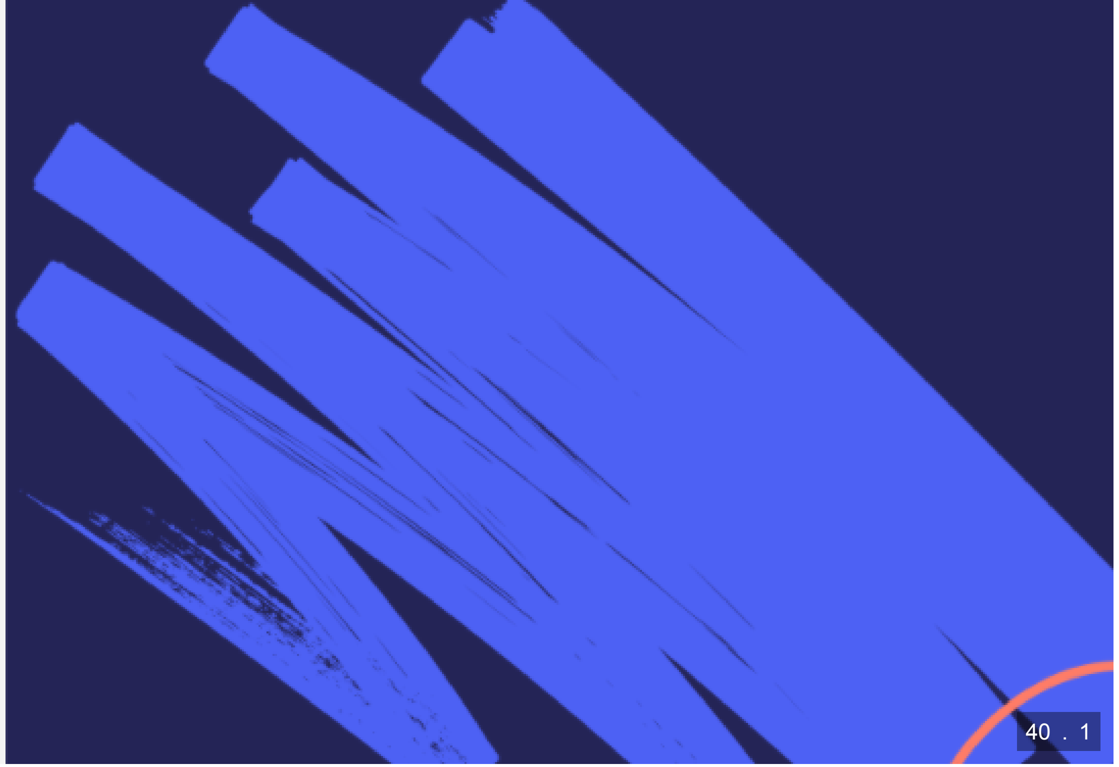


Extract

Data is accessed from the source

For this stage to be effective, a basic understanding of the data is required

There are two methods:



Extract

Data is accessed from the source

For this stage to be effective, a basic understanding of the data is required

There are two methods:

Extract

Data is accessed from the source

For this stage to be effective, a basic understanding of the data is required

There are two methods:

- Current system sends out a copy

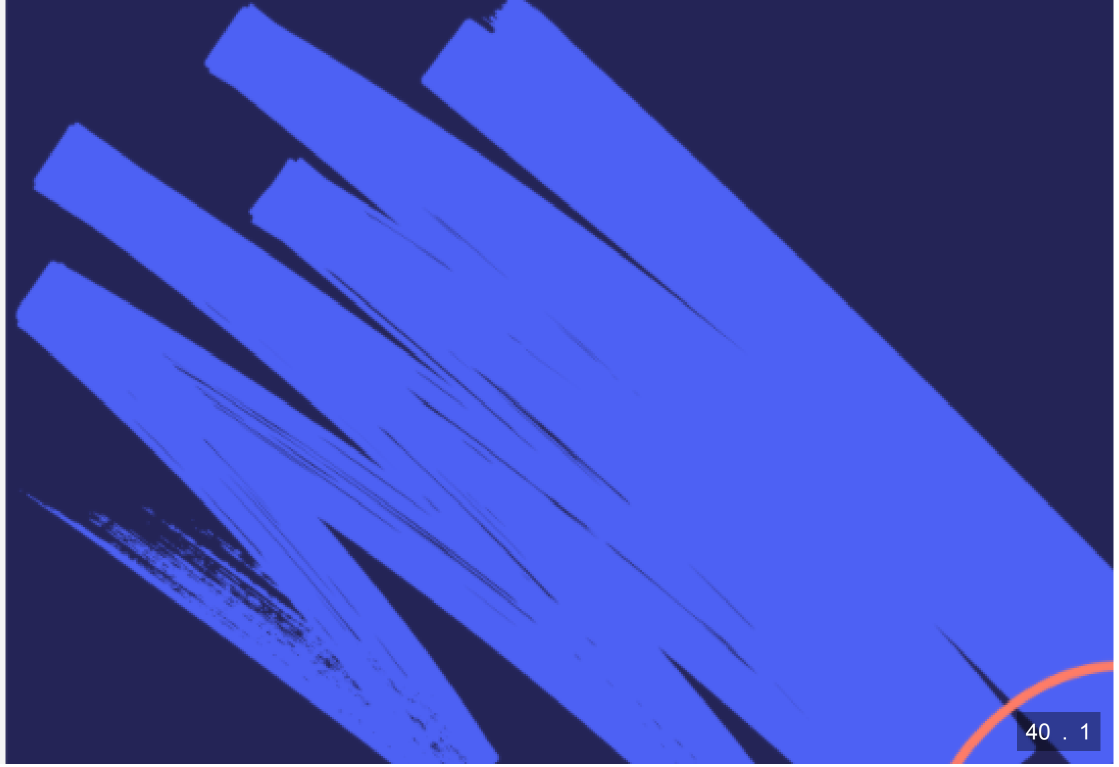
Extract

Data is accessed from the source

For this stage to be effective, a basic understanding of the data is required

There are two methods:

- Current system sends out a copy
- Other system comes in and grabs the data



Extract

Data is accessed from the source

For this stage to be effective, a basic understanding of the data is required

There are two methods:

- Current system sends out a copy
- Other system comes in and grabs the data

Commonly done with SQL queries if data is in databases

Extraction

Can be hard-coded or tool-based

For hard-coding in python:

- CSV Reader (python or R)
- Pandas

```
import pandas as pd

data1=pd.read_csv('dataset1.csv')

data2=pd.read_json('dataset1.json')

data1.info()
data2.info()
```

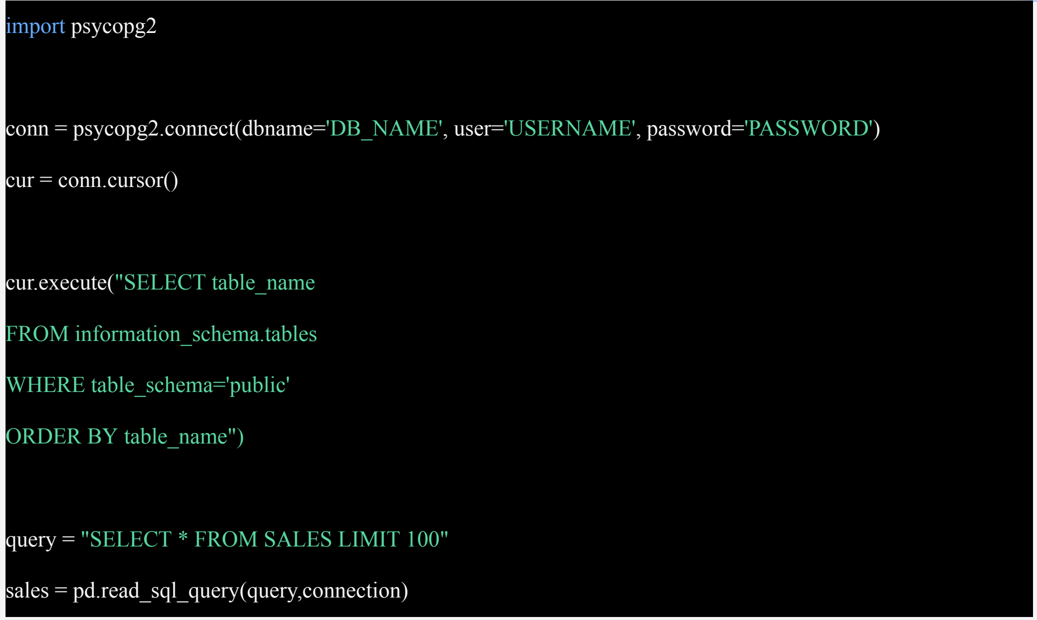
Pandas is not just a data analysis library but can also be used for extraction

Extraction

You can connect to a SQL server to extract data

This can be done in python using:

- Psycopg2 (for postgreSQL)
- SQLAlchemy
- SQLite3



```
import psycopg2


conn = psycopg2.connect(dbname='DB_NAME', user='USERNAME', password='PASSWORD')

cur = conn.cursor()


cur.execute("SELECT table_name

FROM information_schema.tables

WHERE table_schema='public'

ORDER BY table_name")


query = "SELECT * FROM SALES LIMIT 100"

sales = pd.read_sql_query(query,connection)
```

Activity

Open Jupyter Notebook ETL_python

Complete Section 1: Extraction

Transform

Transform

Transform the data to be compatible with the target data structure

Transform

Transform the data to be compatible with the target data structure

Sometimes simple, sometimes near on impossible

Transform

Transform the data to be compatible with the target data structure

Sometimes simple, sometimes near on impossible

Requires detailed requirements elicitation

Transformations could include:

Transformations could include:

Transformations could include:

- Mapping field from source to target

Transformations could include:

- Mapping field from source to target
- String manipulation and manual data standardisation

Transformations could include:

- Mapping field from source to target
- String manipulation and manual data standardisation
- Aggregation and normalisation

Transformations could include:

- Mapping field from source to target
- String manipulation and manual data standardisation
- Aggregation and normalisation
- Calculations

Transformations could include:

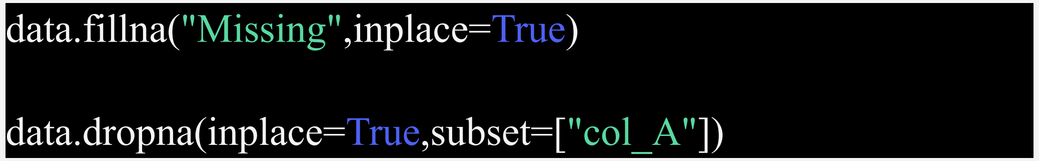
- Mapping field from source to target
- String manipulation and manual data standardisation
- Aggregation and normalisation
- Calculations
- Dealing with duplicate values

Transformations could include:

- Mapping field from source to target
- String manipulation and manual data standardisation
- Aggregation and normalisation
- Calculations
- Dealing with duplicate values
- Data validation

Transformation

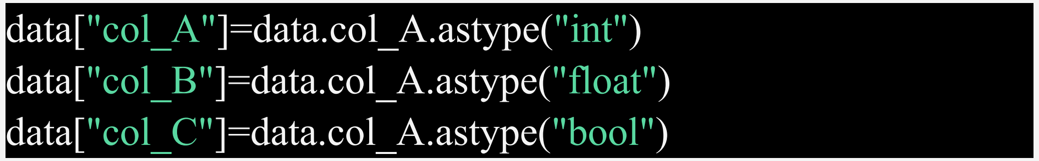
Null Values



```
data.fillna("Missing",inplace=True)

data.dropna(inplace=True,subset=["col_A"])
```

Convert Datatypes



```
data["col_A"]=data.col_A.astype("int")
data["col_B"]=data.col_A.astype("float")
data["col_C"]=data.col_A.astype("bool")
```

Transformation

Deduplication

```
# check for duplicates

data.duplicated

# remove duplicates

data.drop_duplicates(inplace=True)
```

Rename Fields

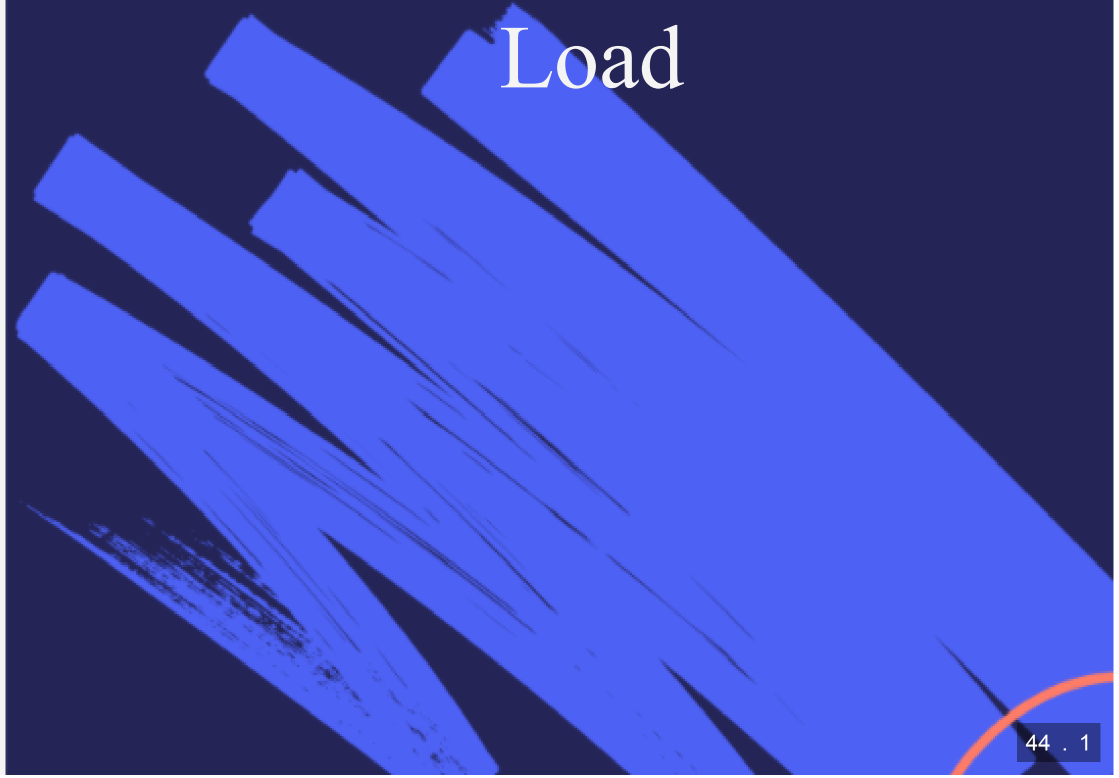
```
data.rename(columns={"col_A":"Col_A"})
```

Activity

Open Jupyter Notebook ETL_python

Complete Section 2: Transformation

Load



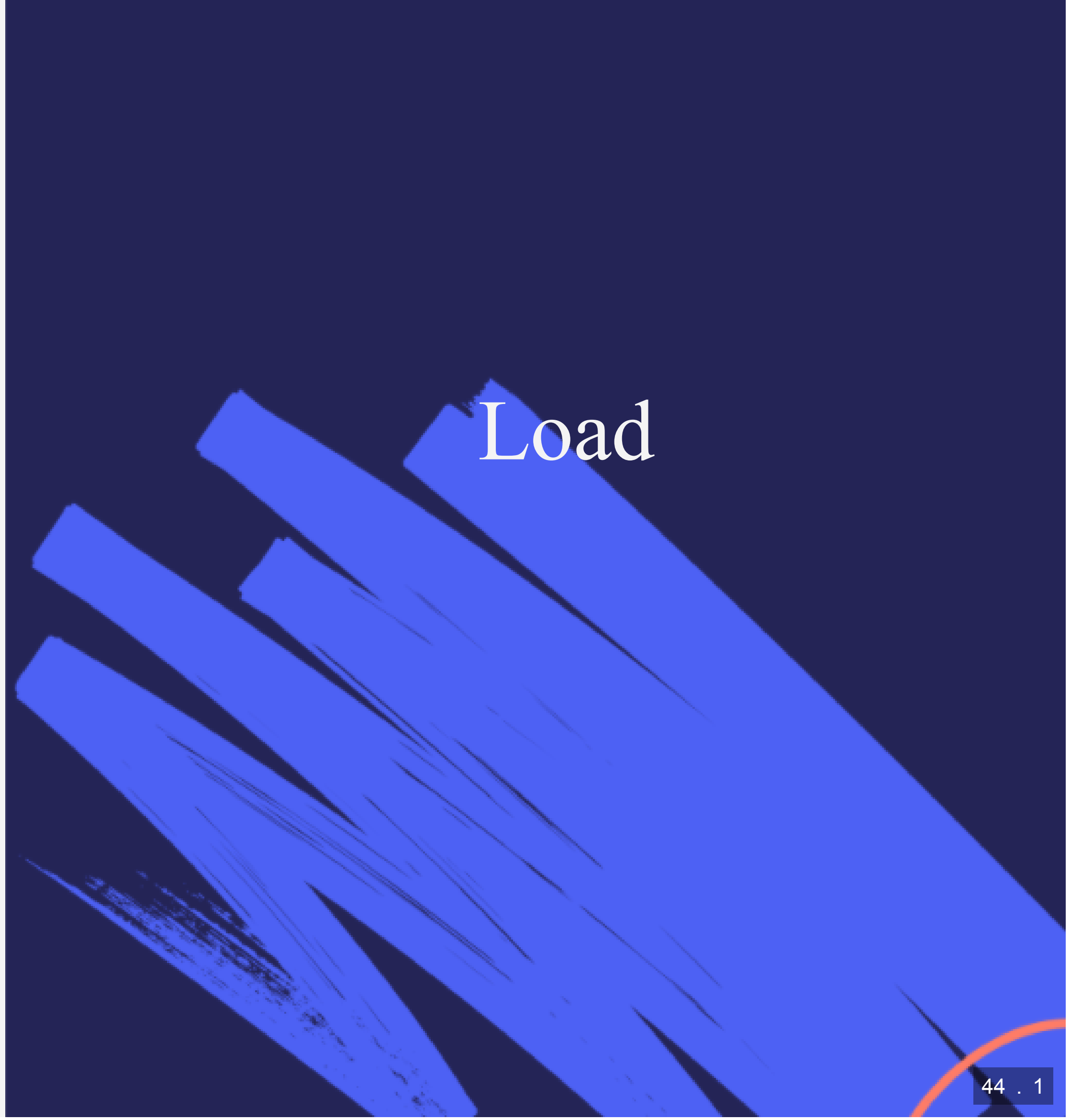
Load

Load the data into the target data structure

Load

Load the data into the target data structure

Either write code to insert data or make use of application code that already exists



Load

Load the data into the target data structure

Either write code to insert data or make use of application code that already exists

Examples include loading into a database or Data Warehouse

Load

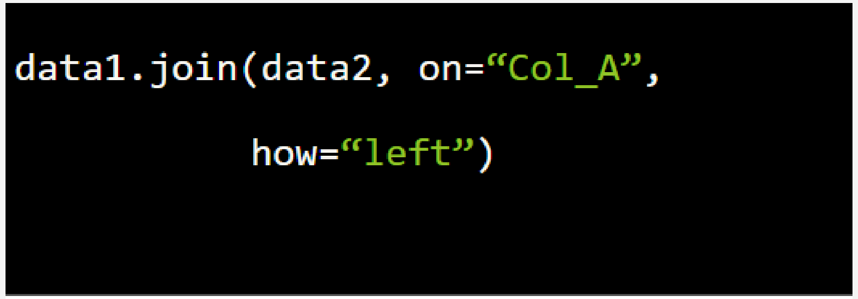
Load the data into the target data structure

Either write code to insert data or make use of application code that already exists

Examples include loading into a database or Data Warehouse

Could involve joining all extracted data into a single table

Loading



```
data1.join(data2, on=“Col_A”,
           how=“left”)
```

Activity

Open Jupyter Notebook ETL_python

Complete Section 3: Loading

Security

Some tips to help you run your ETL processes more securely:

Some tips to help you run your ETL processes more securely:

Some tips to help you run your ETL processes more securely:

- Download the data onto a secure server

Some tips to help you run your ETL processes more securely:

- Download the data onto a secure server
- Run ETL processes on local files or business/enterprise databases

Some tips to help you run your ETL processes more securely:

- Download the data onto a secure server
- Run ETL processes on local files or business/enterprise databases
- If the data owner has not given you the necessary permissions to write data to the target you will need to hand your script to the development team to implement

Automating the Process

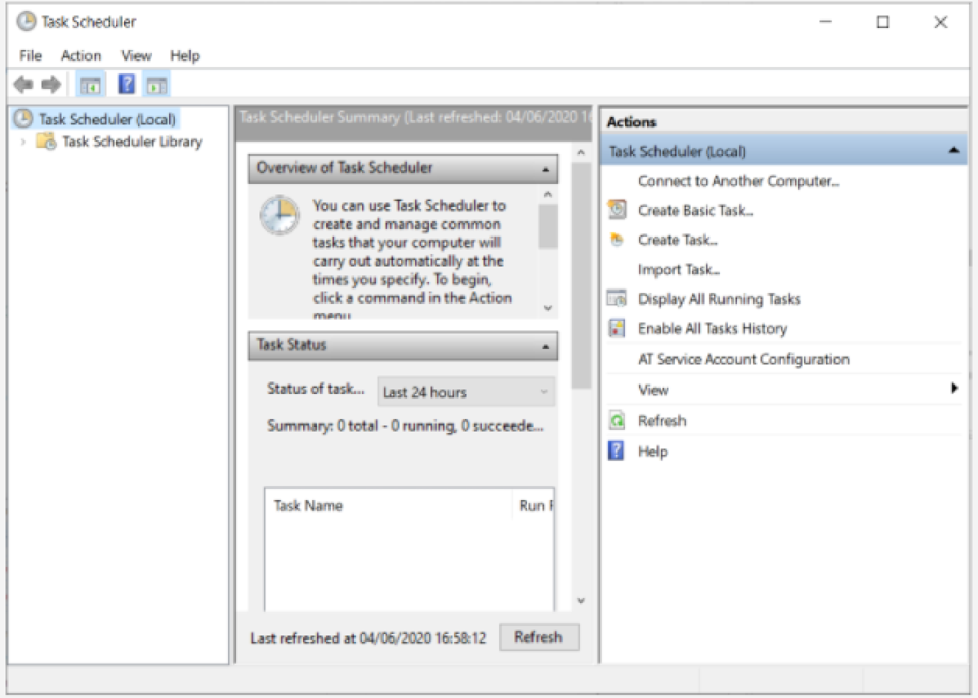
If your ETL process is unlikely to be a one off then it may be more efficient to automate the process.

You will need to assess when new data becomes available to determine how often your scripts need to run.

Task Scheduling

Microsoft has a 'Task Scheduler' which can create batch files

To help with performance, scripts should be run out of hours to ensure performance is not slowed down



Task Scheduler Summary (Last refreshed: 04/06/2020 1

You can use Task Scheduler to create and manage common tasks that your computer will carry out automatically at the times you specify. To begin, click a command in the Action menu.

Status of task... Last 24 hours

Summary: 0 total - 0 running, 0 succeede...

Last refreshed at 04/06/2020 16:58:12

Licenses vs Coding

Advantages

Advantages

License

- Company may already have license
- Friendly GUI
- Supports various databases and formats
- Customer support and good documentation
- Easy scalability for larger datasets

Advantages

License

- Company may already have license
- Friendly GUI
- Supports various databases and formats
- Customer support and good documentation
- Easy scalability for larger datasets

Coding

- Easy to create if database is small
- Easy to install

Disadvantages

Disadvantages

License

- License costs
- Steep learning curve

Disadvantages

License

- License costs
- Steep learning curve

Coding

- Challenging to create (especially if schema changes frequently)
- Developing scripts is time consuming
- Issues around scaling to larger datasets
- Requires programming expertise

Data Integration Softwares

Data Integration Softwares

zapier

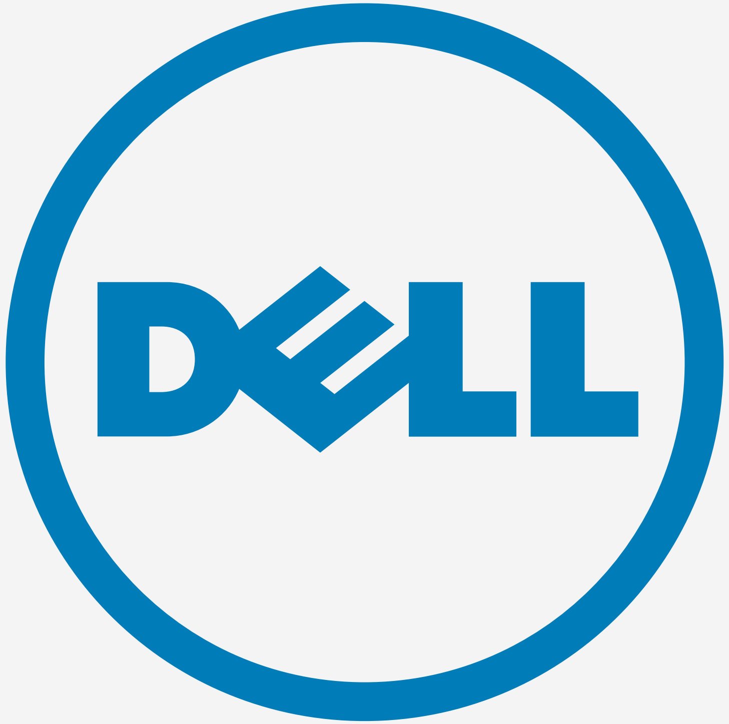
Data Integration Softwares

IFTTT

Data Integration Softwares

automate.io

Data Integration Softwares



Master Data Management

Important data about items in an organisation is called Master Data

This includes customer and product information as well as organisational structure

In business, master data management (MDM) comprises the processes, governance, policies, standards and tools that consistently define and manage the critical data of an organisation to provide a single point of reference.

Benefits

Benefits

Redundancy Elimination

Benefits

Redundancy Elimination

Master Data Edits

Benefits

Redundancy Elimination

Master Data Edits

Data Consistency

Benefits

Redundancy Elimination

Master Data Edits

Data Consistency

Access Based on Role

Testing Strategies

Why do we need testing strategies?

To ensure that unified data sets are:

To ensure that unified data sets are:

To ensure that unified data sets are:

- Correct

To ensure that unified data sets are:

- Correct
- Complete

To ensure that unified data sets are:

- Correct
- Complete
- Up to Date

Types of Testing Strategies

Types of Testing Strategies

Technical Acceptance Testing (TAT)

Types of Testing Strategies

Technical Acceptance Testing (TAT)

User Acceptance Testing (UAT)

Types of Testing Strategies

Technical Acceptance Testing (TAT)

User Acceptance Testing (UAT)

Performance Stress Testing (PST)

Technical Acceptance Testing (TAT)

Testing scripts to ensure they produce the correct output

Can be done manually or by automation

There are three strategies:

Technical Acceptance Testing (TAT)

Testing scripts to ensure they produce the correct output

Can be done manually or by automation

There are three strategies:

Technical Acceptance Testing (TAT)

Testing scripts to ensure they produce the correct output

Can be done manually or by automation

There are three strategies:

- Unit tests

Technical Acceptance Testing (TAT)

Testing scripts to ensure they produce the correct output

Can be done manually or by automation

There are three strategies:

- Unit tests
- Integration tests

Technical Acceptance Testing (TAT)

Testing scripts to ensure they produce the correct output

Can be done manually or by automation

There are three strategies:

- Unit tests
- Integration tests
- Functional tests

TAT: Unit Tests

Testing individual functions or lines of code

Uses python library unittest

Naming convention `test_xxx.py`

Run on your command line: `python -m unittest`

```
import unittest
def fun(x):
  return x+1

class MyTest(unittest.TestCase):
  def test(self):
    self.assertEqual(fun(3),4)
```

Activity

- Open a file in Jupyter Notebook and write a function that sums of a list of numbers
- Write a test case for the function in the same script and name it `test_sum.py`
- Write a new function in a different file that averages a list of numbers
- Write a test case for the function in the same script and name it `test_average.py`
- Open a terminal and run `python -m unittest`

TAT: Integration Tests

Integration tests verify that different modules or services used by your application work well together.

For example, it can be testing the interaction with a database, i.e. are you able to write queries?

TAT: Functional Tests

These focus on the business requirements of an application. They only verify the output of an action and do not check the intermediate states of the system when performing that action.

User Acceptance Testing (UAT)

Formal tests to verify if a report or system statisfies its business requirements

Can be done manually or by automation

User Acceptance Testing (UAT)

Answers questions like:

User Acceptance Testing (UAT)

Answers questions like:

User Acceptance Testing (UAT)

Answers questions like:

- Does the report meet the original requirements?

User Acceptance Testing (UAT)

Answers questions like:

- Does the report meet the original requirements?
- Does the report produce sensible information?

User Acceptance Testing (UAT)

Answers questions like:

- Does the report meet the original requirements?
- Does the report produce sensible information?
- Is the design and layout acceptable?

Performance Stress Testing (PST)

Focusses on validating performance characteristics of the product such as scalability and reliability

They check the behaviours of the system when it is under a significant load

Performance Stress Testing (PST)

Tests are based on non functional requirements:

Performance Stress Testing (PST)

Tests are based on non functional requirements:

Performance Stress Testing (PST)

Tests are based on non functional requirements:

- Scalability

Performance Stress Testing (PST)

Tests are based on non functional requirements:

- Scalability
- Reliability

Performance Stress Testing (PST)

Tests are based on non functional requirements:

- Scalability
- Reliability
- Stability

Performance Stress Testing (PST)

Tests are based on non functional requirements:

- Scalability
- Reliability
- Stability
- Availability

Recap

Learning Objectives

- **Understand** concepts of Data Integration and ETL techniques
- **Explain** the difference between **Data Integration** and **Data Migration**
- Explore different testing strategies for Data Integration

multiverse

Complete Session Attendance Log and Update Your OTJ