
multiverse

Database Fundamentals

Session 2

Session Outline

Data Modelling

Entity Relationship Diagrams

DIY Database Design

Data Warehouses

Data Integration

Types of Data Integration

Data Integration Life Cycle

Rules and Policies

Extract, Transform, Load

Combining Data

Recap

Learning Objectives

- Design and visualise an **entity relationship diagram** for a database
- Understand the principles of **data integration** and how it is used to help our analysis >
- Describe the **ETL process** and how it is used to in data integration

multiverse

Data Modelling

Why use a Data Model?

All data objects required by the database are **accurately represented**

All data objects required by the database are **accurately represented**

A data model helps design the database at the **conceptual, physical and logical levels**

All data objects required by the database are **accurately represented**

A data model helps design the database at the **conceptual, physical and logical levels**

Data model structure helps define **relational tables**, **primary and foreign keys** as well as **stored procedures**

All data objects required by the database are **accurately represented**

A data model helps design the database at the **conceptual, physical and logical levels**

Data model structure helps define **relational tables**, **primary and foreign keys** as well as **stored procedures**

It provides a clear picture of the **base data**

All data objects required by the database are **accurately represented**

A data model helps design the database at the **conceptual, physical and logical levels**

Data model structure helps define **relational tables**, **primary and foreign keys** as well as **stored procedures**

It provides a clear picture of the **base data**

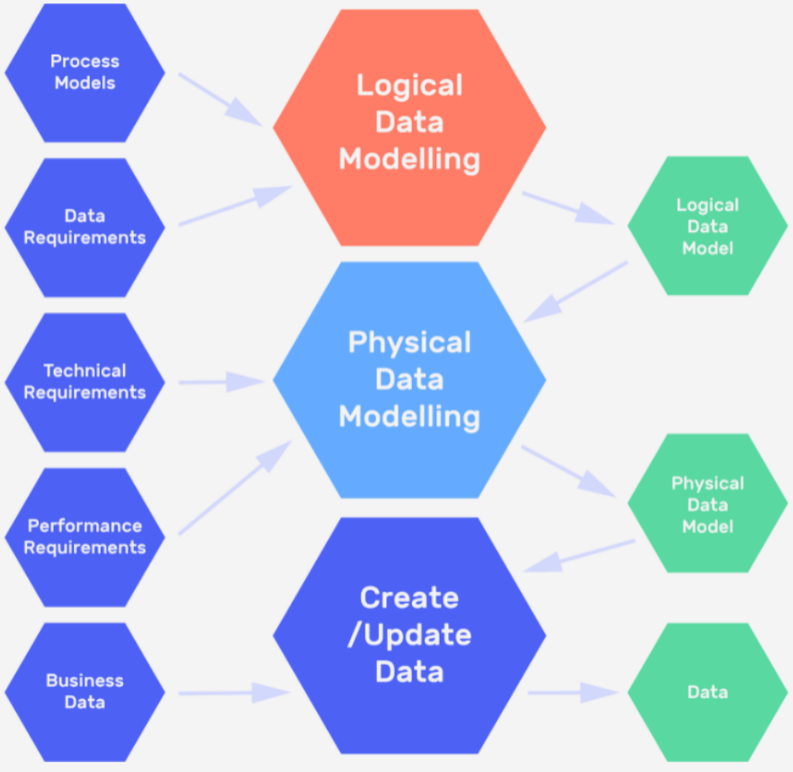
It can help identify **missing or redundant data**

Types of Data Models

Conceptual

Logical

Physical



Types of Data Models

Conceptual

Logical

Physical

An organised view of database concepts and their relationships.
The purpose is to establish:

Types of Data Models

Conceptual

Logical

Physical

An organised view of database concepts and their relationships.
The purpose is to establish:

Entities

Types of Data Models

Conceptual

Logical

Physical

An organised view of database concepts and their relationships.
The purpose is to establish:

Entities

Attributes

Types of Data Models

Conceptual

Logical

Physical

An organised view of database concepts and their relationships.
The purpose is to establish:

Entities

Attributes

Relationships

CUSTOMER
Customer Name Customer Number

Sale

→

PRODUCT
Product Name Product Price

Characteristics of a Conceptual Data Model

Characteristics of a Conceptual Data Model

Offers organisation wide coverage of business concepts

Characteristics of a Conceptual Data Model

Offers organisation wide coverage of business concepts

Designed and developed for business audiences

Characteristics of a Conceptual Data Model

Offers organisation wide coverage of business concepts

Designed and developed for business audiences

Developed independently of hardware specifications like storage capacity, or software specifications like DBMS technology. The focus to represent data as a user would see it in the 'real world'

Types of Data Models

Conceptual

Logical

Physical

Used to define the structure of data elements and set relationships between them. This type of model:

Types of Data Models

Conceptual

Logical

Physical

Used to define the structure of data elements and set relationships between them. This type of model:

Adds further information to the conceptual data model

Types of Data Models

Conceptual

Logical

Physical

Used to define the structure of data elements and set relationships between them. This type of model:

Adds further information to the conceptual data model

Provides a foundation for a physical model (yet retains a generic structure)

Types of Data Models

Conceptual

Logical

Physical

Used to define the structure of data elements and set relationships between them. This type of model:

Adds further information to the conceptual data model

Provides a foundation for a physical model (yet retains a generic structure)

Does not require keys, just need to verify the connector details set for earlier relationships

CUSTOMER	
Customer Name Customer Number	string integer

Sale

→

PRODUCT	
Product Name Product Price	string decimal

Common Data Types

Common Data Types

string/character/varchar

Common Data Types

string/character/varchar

integer

Common Data Types

string/character/varchar

integer

number/float/decimal

Common Data Types

string/character/varchar

integer

number/float/decimal

datetime

Common Data Types

string/character/varchar

integer

number/float/decimal

datetime

boolean

Characteristics of a Logical Data Model

Characteristics of a Logical Data Model

Describes data needs for a single project but could integrate with other logical data models based on the scope of the project

Characteristics of a Logical Data Model

Describes data needs for a single project but could integrate with other logical data models based on the scope of the project

Designed and developed independently from the DBMS

Characteristics of a Logical Data Model

Describes data needs for a single project but could integrate with other logical data models based on the scope of the project

Designed and developed independently from the DBMS

Data attributes will have data types with exact precisions and length

Characteristics of a Logical Data Model

Describes data needs for a single project but could integrate with other logical data models based on the scope of the project

Designed and developed independently from the DBMS

Data attributes will have data types with exact precisions and length

Normalisation processes to the model is applied typically till 3NF

Types of Data Models

Conceptual

Logical

Physical

Describes a database specific implementation of the data model. This type of model:

Types of Data Models

Conceptual

Logical

Physical

Describes a database specific implementation of the data model. This type of model:

Offers database abstraction and helps generate schema through the rich meta-data

Types of Data Models

Conceptual

Logical

Physical

Describes a database specific implementation of the data model. This type of model:

Offers database abstraction and helps generate schema through the rich meta-data

Helps visualise database structure by replicating column keys, constraints and other RDBMS features

CUSTOMER	
Customer Name Customer Number	string integer
Primary Key Customer Number	

Sale

→

PRODUCT	
Product Name Product Price	string decimal
Primary Key Product Name	

Characteristics of a Physical Data Model

Characteristics of a Physical Data Model

Contains relationships between tables

Characteristics of a Physical Data Model

Contains relationships between tables

Developed for a specific version of a DBMS, location, data storage or technology to be used in a project

Characteristics of a Physical Data Model

Contains relationships between tables

Developed for a specific version of a DBMS, location, data storage or technology to be used in a project

Columns should have exact data types, lengths and default values assigned

Characteristics of a Physical Data Model

Contains relationships between tables

Developed for a specific version of a DBMS, location, data storage or technology to be used in a project

Columns should have exact data types, lengths and default values assigned

Primary and foreign keys, views, indices, access profiles and authorisations are defined

multiverse

Entity Relationship Diagrams

An Entitiy Relationship Diagram (ERD) lets you see how different entities (e.g. customers, products) relate to each other in a database.

Components

Entity

Relationship

Attribute

Components

Entity

Relationship

Attribute

A defined object within a database (e.g. customers, products, contractors, etc)

ENTITY
Attribute1 Attribute2 Attribute3

Strong Entitiy

Has a primary key and does not depend on another entity to exist

E.g. student information

Strong Entitiy

Has a primary key and does not depend on another entity to exist

E.g. student information

Weak Entity

Depends on another entity to exist

I.e. the primary key is a foreign key in another table

E.g. student enrolment information

Components

Entity

Relationship

Attribute

Defines how two entities are related to each other

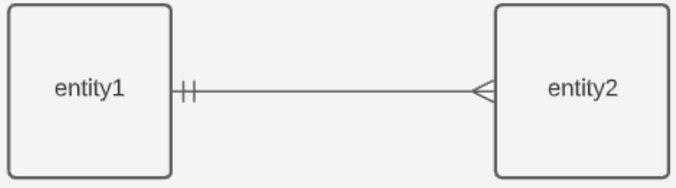
Represented as lines with a "cardinality" that explains the number of instances between two entities

One to One Relationship

One record of an entity is directly related to another record of an entitiy

One to Many Relationship

One record of an entity is related to one or more records of another entity



Many to Many Relationship

Many records of one entity can be related to many records of another entity

One or Zero to Many Relationship

One record of an entity is related to zero, one or more records of another entity

Components

Entity

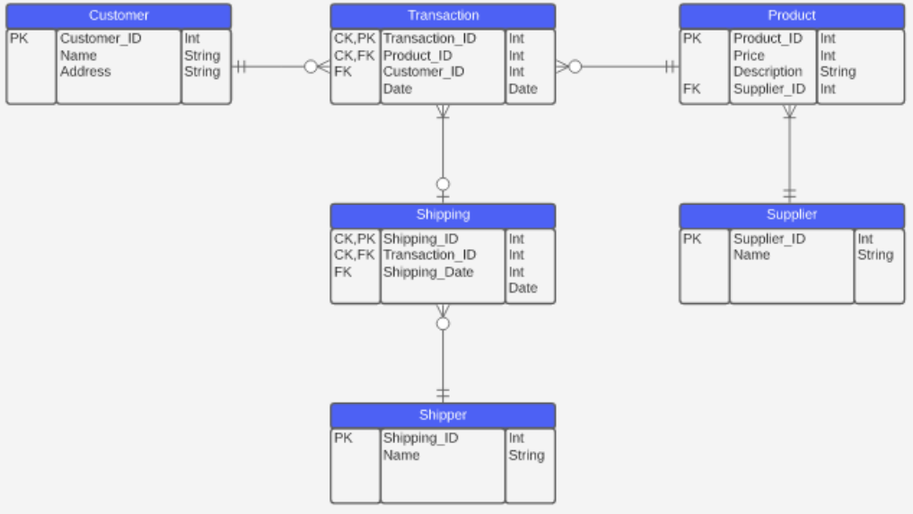
Relationship

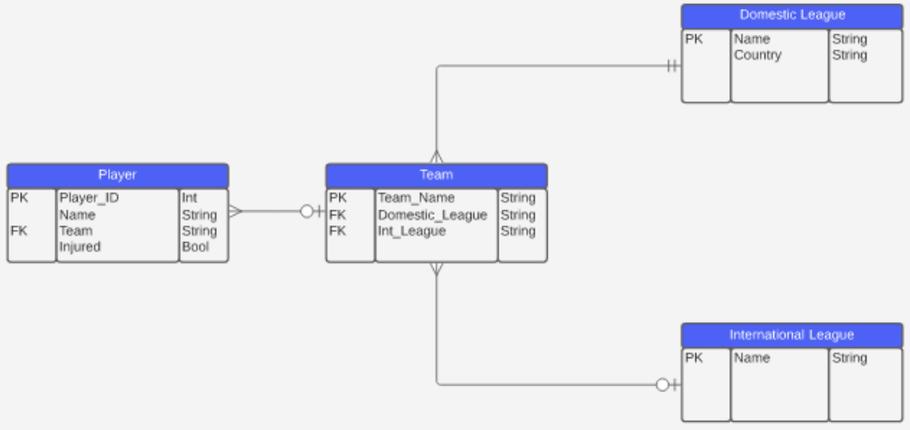
Attribute

A property of an entity

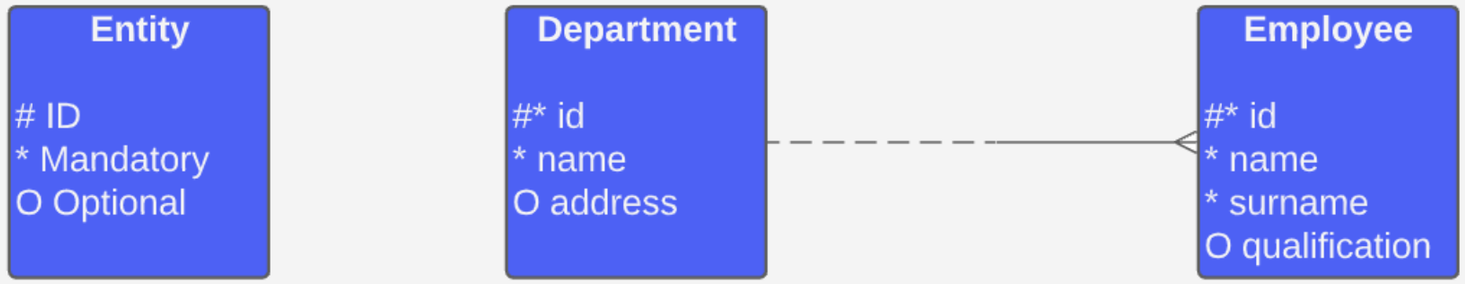
ATTRIBUTE TYPE	DESCRIPTION	EXAMPLE
Simple	Cannot be split into other attributes	First name, surname
Composite	Can be split into other attributes	Name (can be split into forename, middle name and surname)
Derived	Calculated or determined from another attribute	Age of record calculated from creation date


Student
PK
Student_ID
Name
Date of Birth
Address
Int
String
Date
String
Student_course
CK,FK
CK,FK
Student_ID
Course_ID
Enrollement Date
Int
Int
Date
String
Course
PK
Course_ID
Course_Name
Lecturer
Int
String
String


Customer
PK
Customer_ID
Name
Address
Int
String
String
Transaction
CK,PK
CK,FK
FK
Transaction_ID
Product_ID
Customer_ID
Date
Int
Int
Int
Date
Product
PK
FK
Product_ID
Price
Description
Supplier_ID
Int
Int
String
Int
Shipping
CK,PK
CK,FK
FK
Shipping_ID
Transaction_ID
Shipping_Date
Int
Int
Int
Date
Supplier
PK
Supplier_ID
Name
Int
String
Shipper
PK
Shipping_ID
Name
Int
String


Domestic League
PK
Name
Country
String
String
Player
PK
FK
Player_ID
Name
Team
Injured
Int
String
String
Bool
Team
PK
FK
FK
Team_Name
Domestic_League
Int_League
String
String
String
International League
PK
Name
String

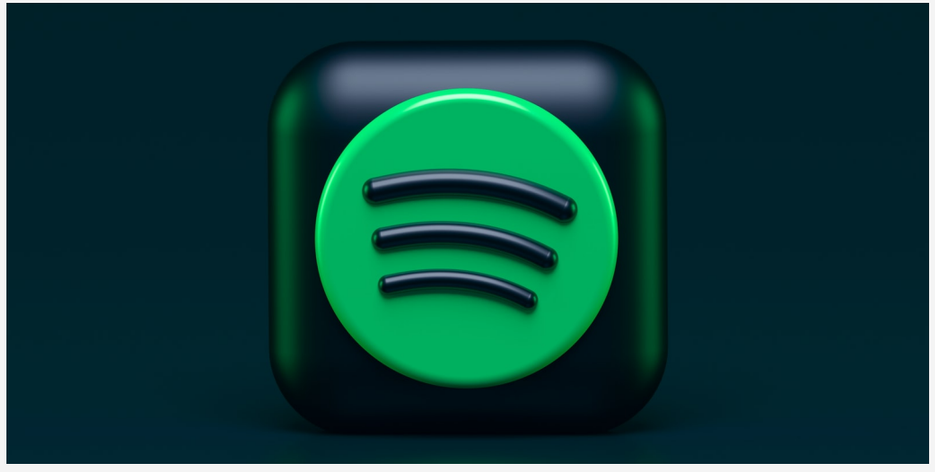
Barker Notation



multiverse

DIY Database Design

Spotify

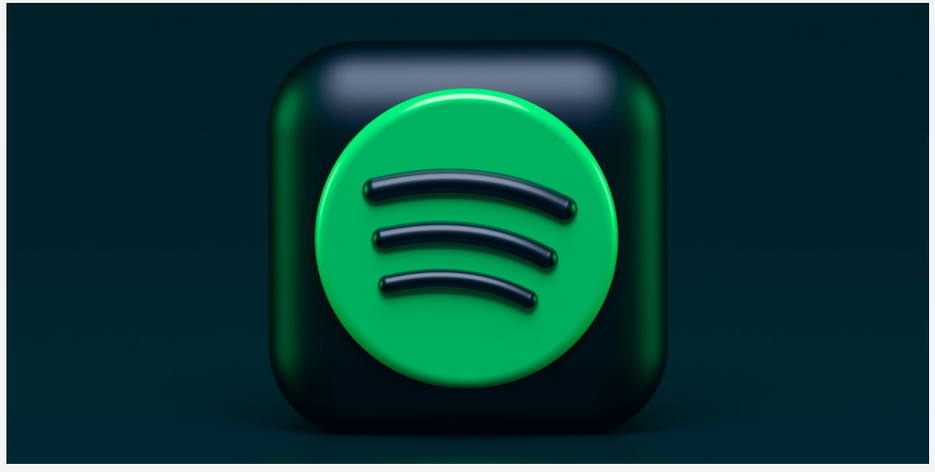


What is its purpose?

What is the user input?

What will the user receive?

Spotify



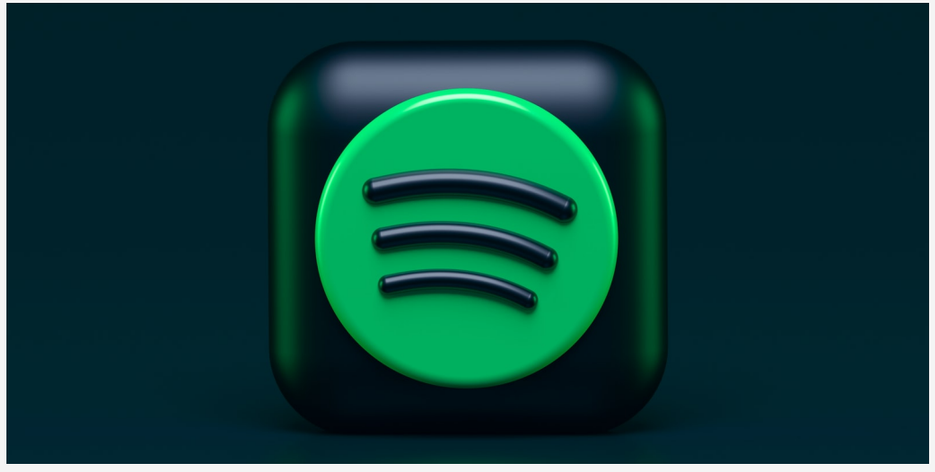
What is its purpose?

Stream music based on artists, albums, playlists or genres

What is the user input?

What will the user receive?

Spotify



What is its purpose?

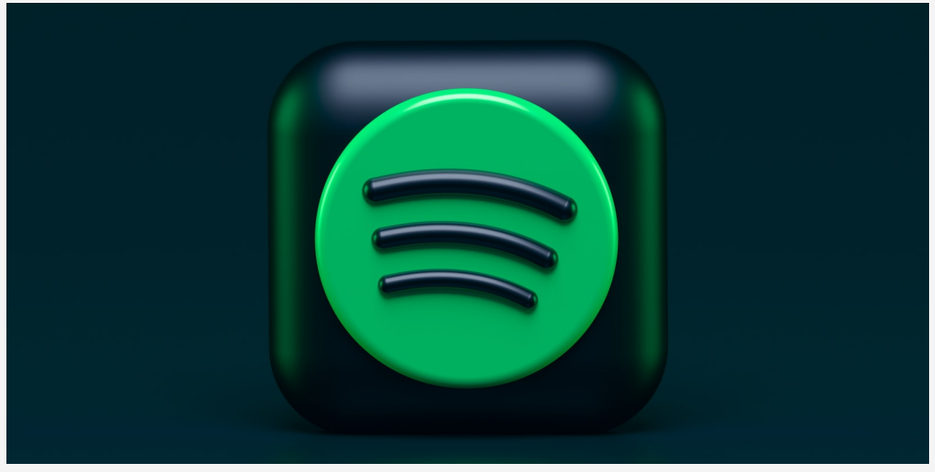
Stream music based on artists, albums, playlists or genres

What is the user input?

Track title, album title, artist name or playlist title

What will the user receive?

Spotify



What is its purpose?

Stream music based on artists, albums, playlists or genres

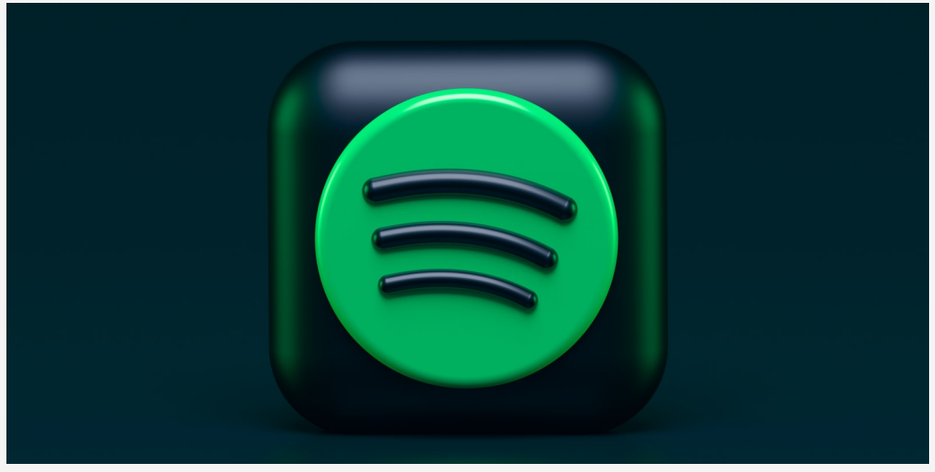
What is the user input?

Track title, album title, artist name or playlist title

What will the user receive?

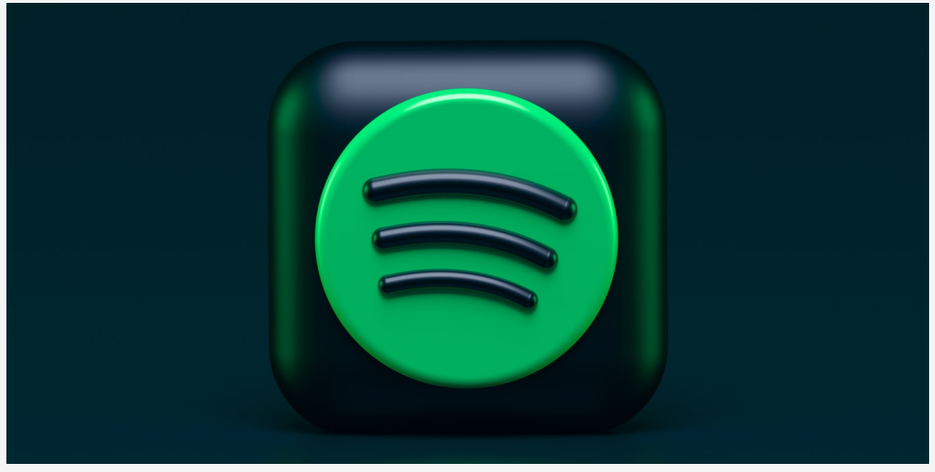
Music content

Spotify



What database tables are required?

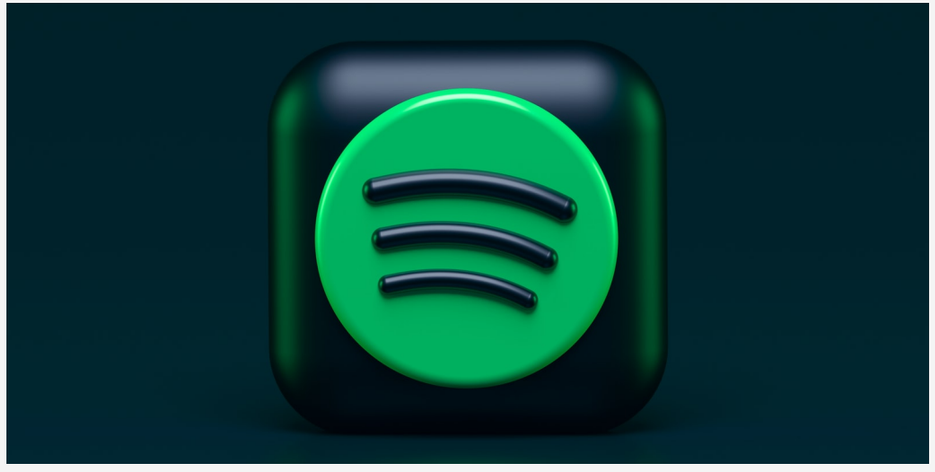
Spotify



What database tables are required?

Artists

Spotify

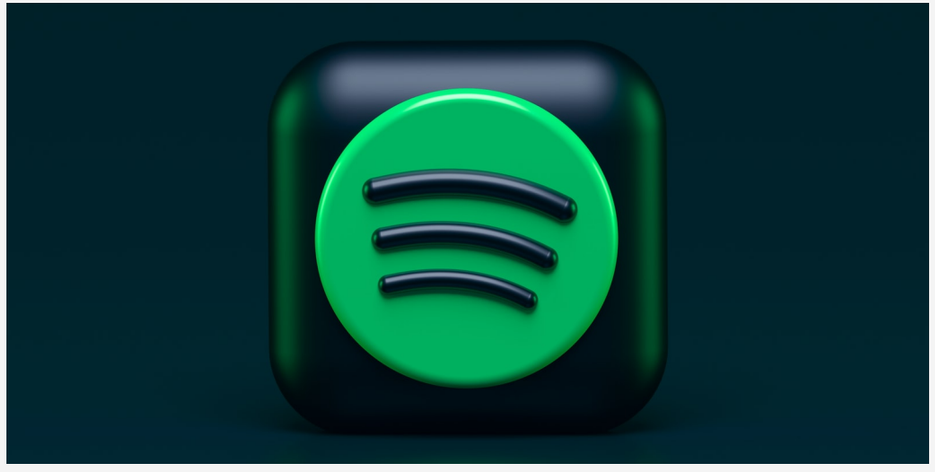


What database tables are required?

Artists

Albums

Spotify



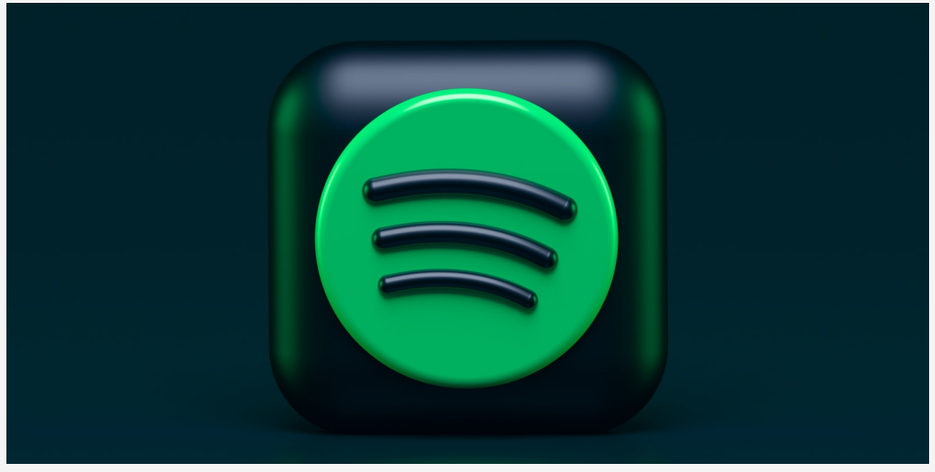
What database tables are required?

Artists

Albums

Playlists

Spotify



What database tables are required?

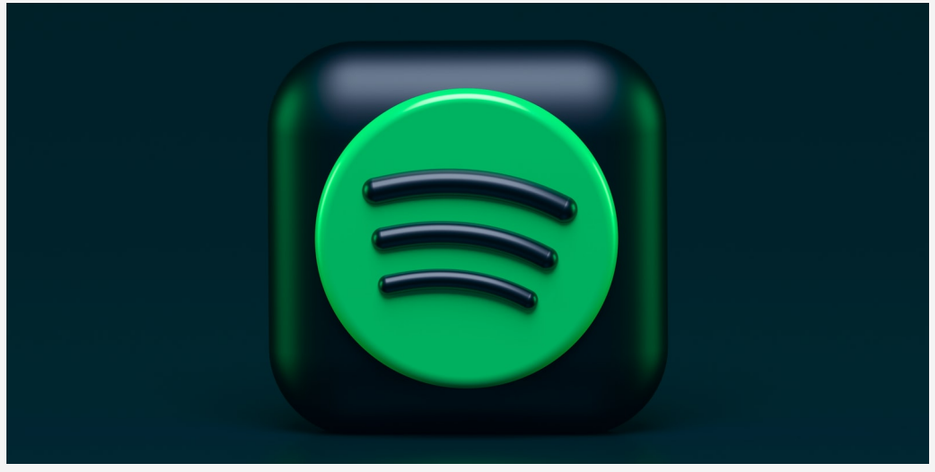
Artists

Albums

Playlists

Tracks

Spotify



What database tables are required?

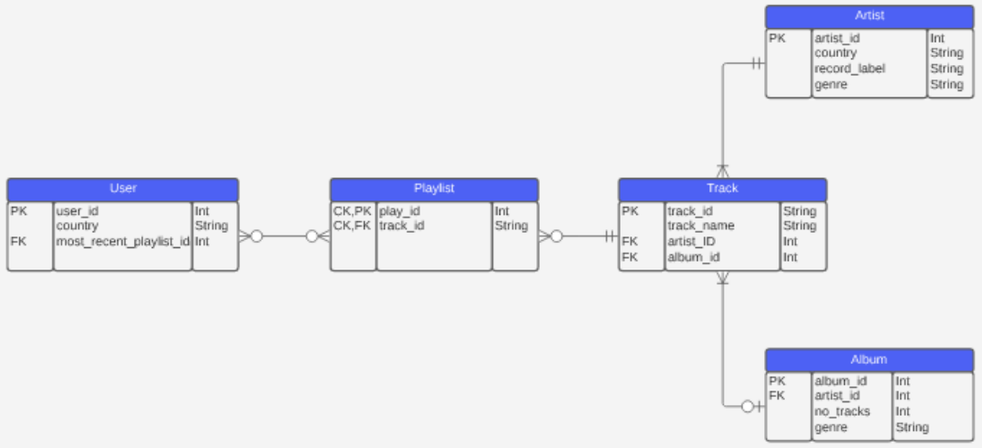
Artists

Albums

Playlists

Tracks

Users

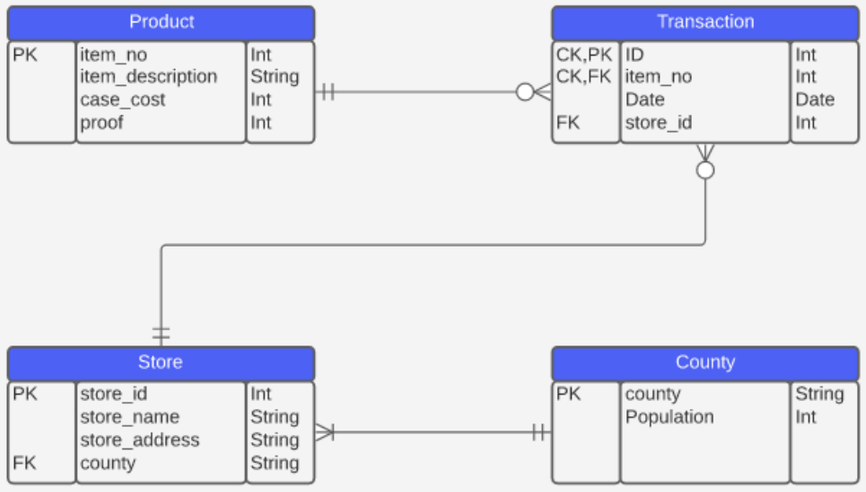

Artist
PK
artist_id
country
record_label
genre
Int
String
String
String
User
PK
FK
user_id
country
most_recent_playlist_id
Int
String
Int
Playlist
CK,PK
CK,FK
play_id
track_id
Int
String
Track
PK
FK
FK
track_id
track_name
artist_ID
album_id
String
String
Int
Int
Album
PK
FK
album_id
artist_id
no_tracks
genre
Int
Int
Int
String

Activity

The state of Iowa wants to design a database that records alcohol sales from all stores in the state.

Using the information on the following slide, design an ERD to show how how the database should bs structured.

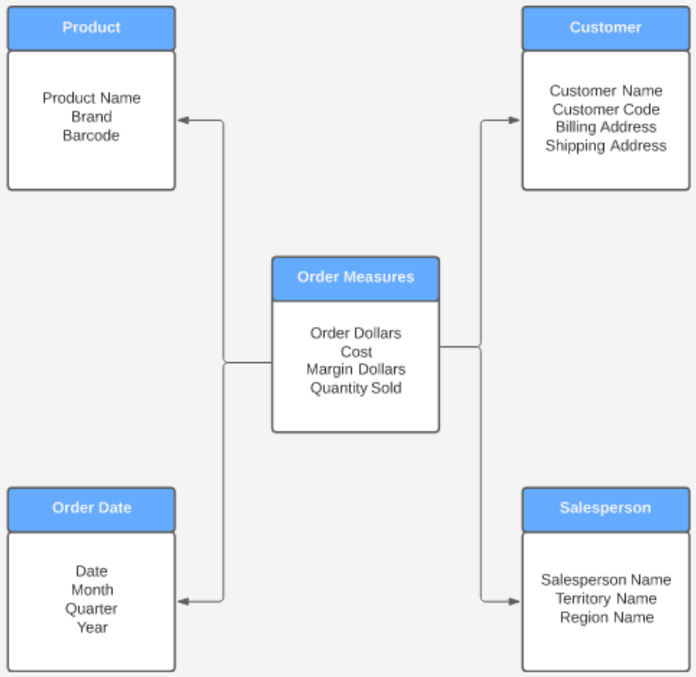
ENTITY	Transaction	Product	Store	County
ATTRIBUTES	ID Date item_no store_id	item_no description case_cost proof	store_id store_name address county	county population


Product
PK
item_no
item_description
case_cost
proof
Int
String
Int
Int
Transaction
CK,PK
CK,FK
FK
ID
item_no
Date
store_id
Int
Int
Date
Int
Store
PK
FK
store_id
store_name
store_address
county
Int
String
String
String
County
PK
county
Population
String
Int

multiverse

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**.


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

Key Features

Key Features

Optimised for low number of complex queries

Key Features

Optimised for low number of complex queries

Contains historical data

Key Features

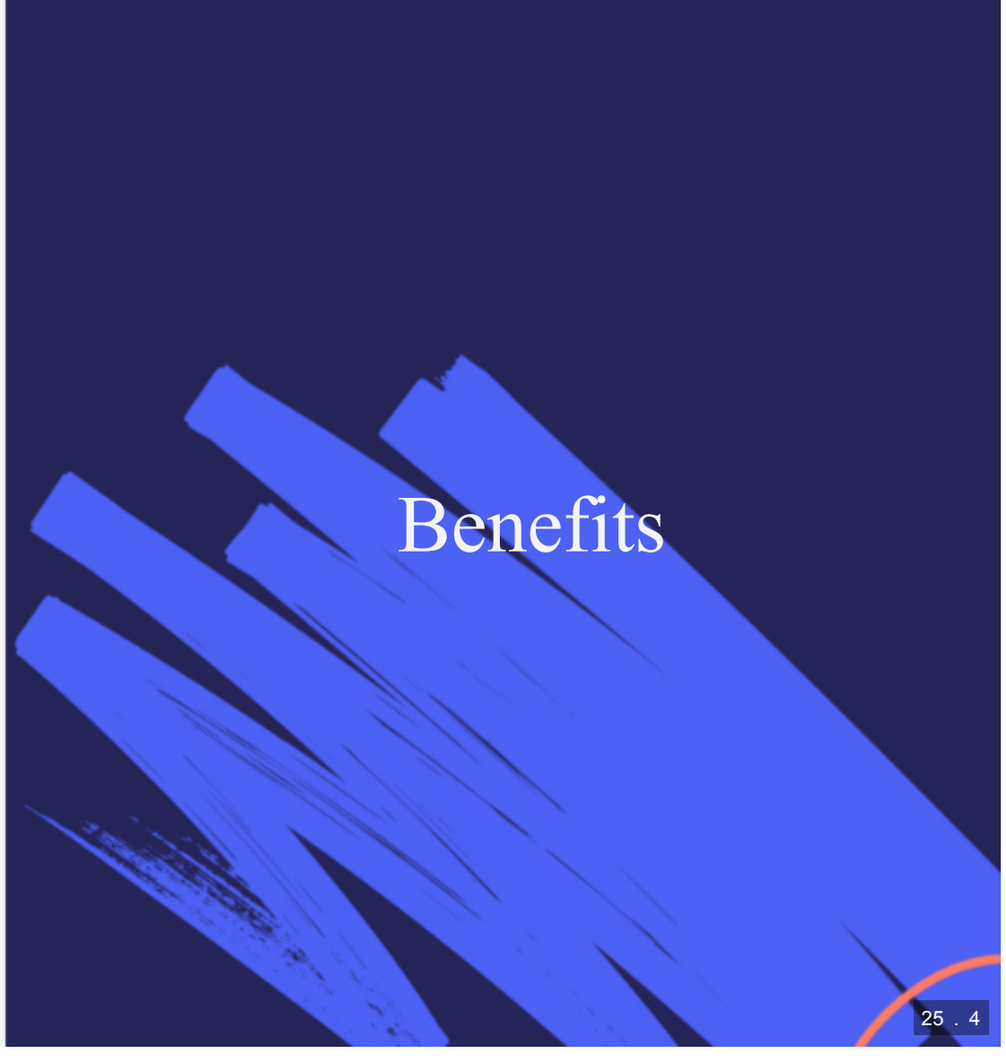
Optimised for low number of complex queries

Contains historical data

Are taken off line when updates are required

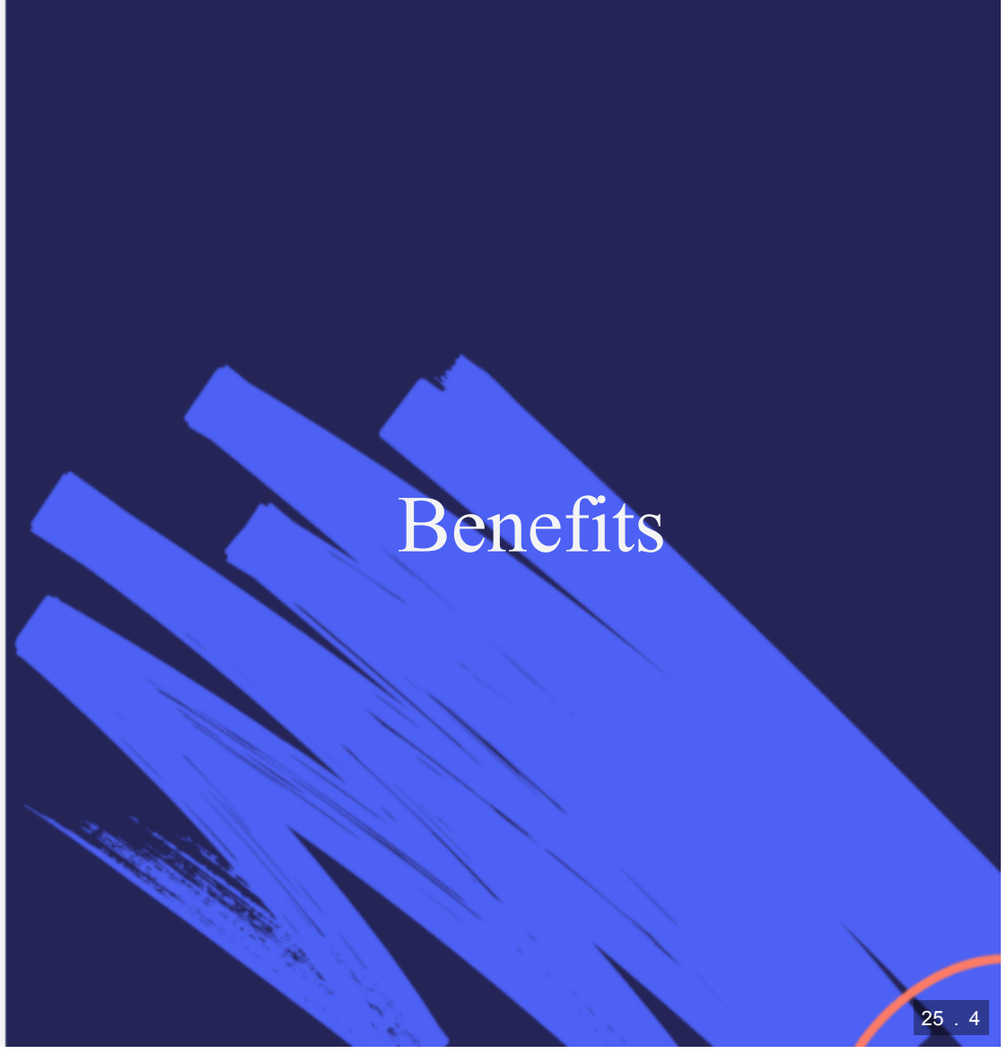
Benefits

Increased availability of data



Increased availability of data

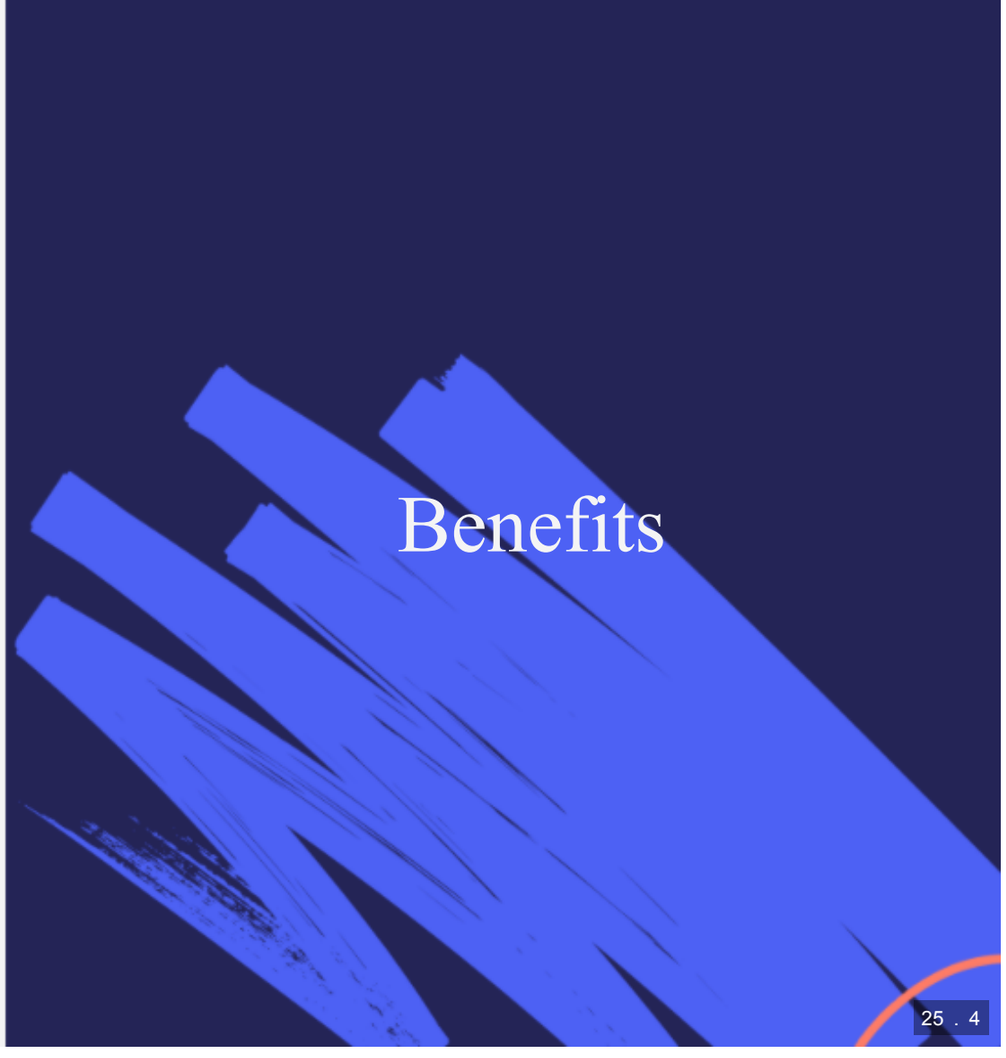
Superior quality of data



Increased availability of data

Superior quality of data

Collaboration opportunities

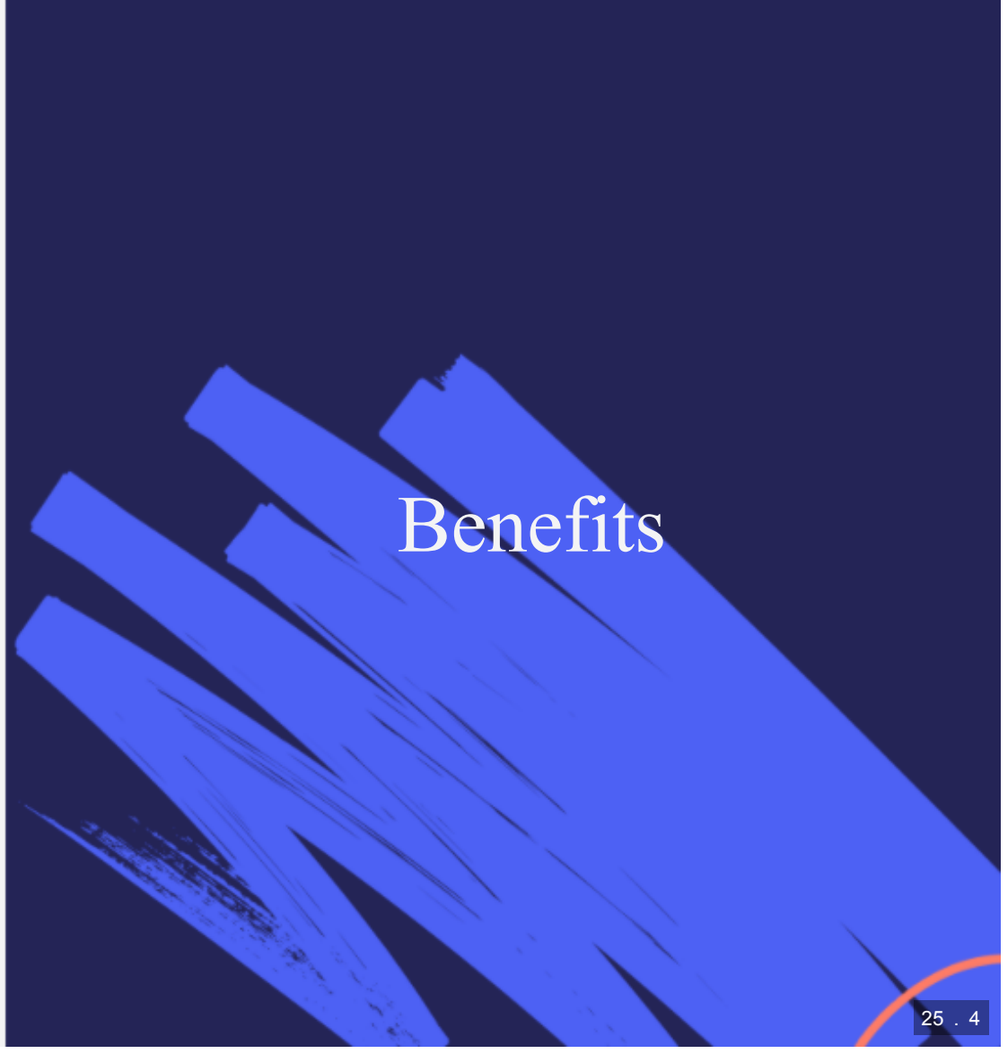


Increased availability of data

Superior quality of data

Collaboration opportunities

Greater insights and improvements



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	Approx 100% uptime
Query Type	Complex queries for in depth analysis	Simple transactional

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)

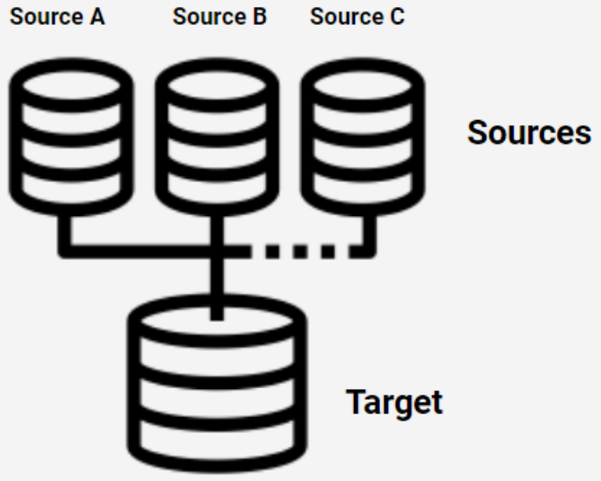
Targeted Adverts

Suggested LinekedIn connections

multiverse

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

Increased availability of data

Benefits of Data Integration

Increased availability of data

Superior data integrity and quality

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

Increased availability of data

Superior data integrity and quality

Collaboration opportunities

Greater insights and improvements

Improved data consistency

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)

multiverse

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

multiverse

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)

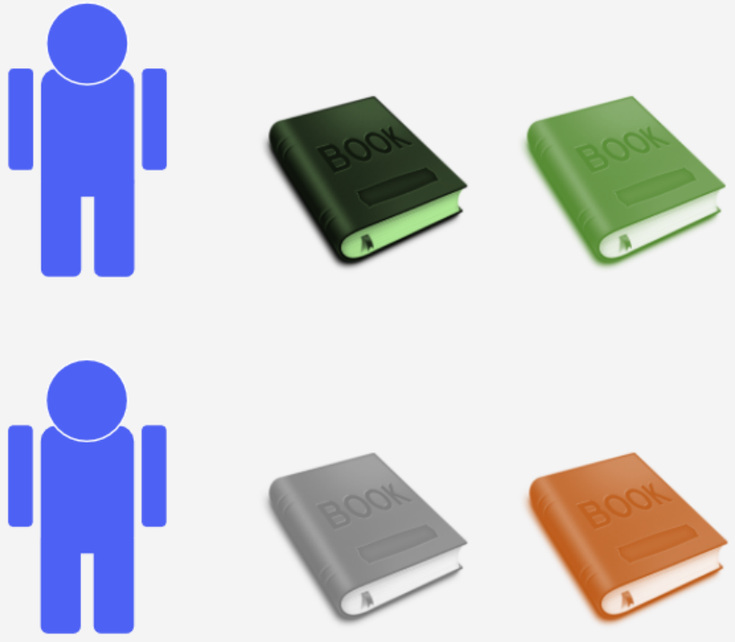
multiverse

Extract, Transform, Load

How would you count the number of occurrences of each word in all the books found in a library using a team of people?

Step 1

Divide the books among the team so every person has an allocation



Step 2

Each person will keep a record of the occurrences of each word in their allocation

WORD	COUNT
Apple	2
Bird	7

WORD	COUNT
Apple	5
Bird	1

Step 3

Finally combine the different records into one unified view which contains each word in the library.

WORD	COUNT
Apple	7
Bird	8


Customer support
Sales
Inventory
Advertising
ADS

Extraction is the process of gathering data from a variety of disparate sources

The extracted data is usually copied from the source, not moved

Validation occurs at this stage to ensure the data is in the correct structure and format, as well as ensuring necessary permissions have been given

The process can be continuous or done in batches

Transformation is the process of ensuring the extracted data is in a consistent format

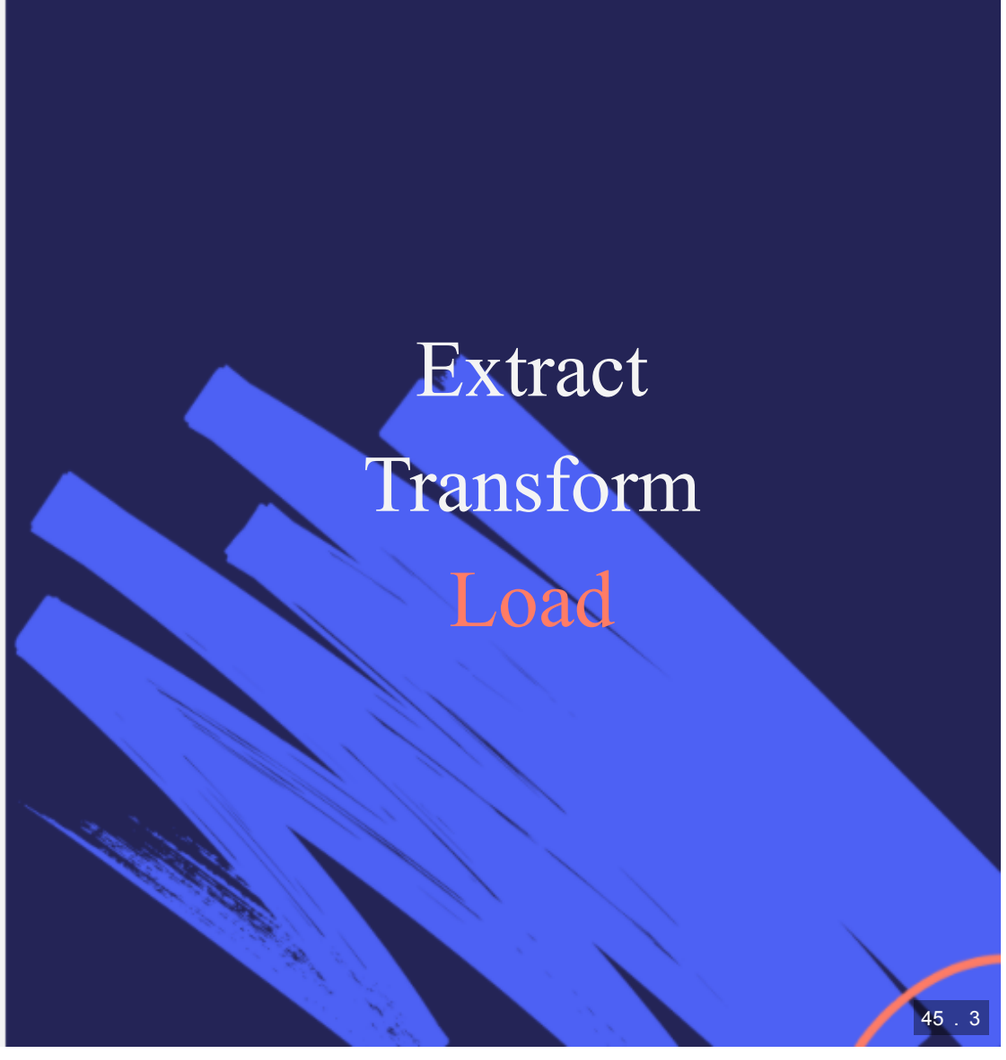
This can include removing null values, changing data types and ensuring field names are the same

As the extracted data is a copy, the original will remian unchanged

Loading is the process of joining the transformed data together into a single unified view (called the target)

Data verification is undertaken post loading to ensure the combined data is accurate and fulfils the necessary business requirements

With 'Big Data' this process is done using parallel processing to manage the large volume of data being written to the system



Benefits

Allows for a unified view of data that is otherwise spread out across an organisation

Ensures data consistency across an organisation allowing for missing data and errors to be identified throughout a pipeline

Encourages collaboration across teams

Better business intelligence and insights for making decisions through greater data availability

Information Structure and Rules

Data integration activities for data warehouses requires that you follow some basic rules:

- Security policies must be specified by organisations providing data sources to **prevent data leakage and unauthorised access**
- **Access layers** (e.g. networks, firewalls, servers, etc) between sources and targets should be properly configured (especially of data is sourced externally)
- Integrated data should be **immutable**- you should not be able to change the data once it is stored in the unified view
- **Validation checks** should be carried out during ETL:
- Source and target table structures and data types should be **consistent**
- **Column names** should be the same as defined by a mapping document

Information Structure and Rules

Data integration activities for data warehouses requires that you follow some basic rules:

- **Verification** is also carried out on the target:
- Verify that the data is **accurate**
- Verify the data is the **'right' data** to be stored in the target
- Verify the data has **not been duplicated**

multiverse

Combining Data

Things to consider

Things to consider

What were the data sources?

Things to consider

What were the data sources?

How did you transform the data into a consistent format?

Things to consider

What were the data sources?

How did you transform the data into a consistent format?

How did you join the data together?

Things to consider

What were the data sources?

How did you transform the data into a consistent format?

How did you join the data together?

How did you verify the accuracy of your integrated data?

Things to consider

What were the data sources?

How did you transform the data into a consistent format?

How did you join the data together?

How did you verify the accuracy of your integrated data?

How did you address security risks?

Things to consider

What were the data sources?

How did you transform the data into a consistent format?

How did you join the data together?

How did you verify the accuracy of your integrated data?

How did you address security risks?

What was the benefit of combining your data?

multiverse

Recap

Learning Objectives

- Design and visualise an **entity relationship diagram** for a database
- Identify starting points for overcoming **project and technical problems**
- Implement **Quality Control** measures and know how to assure the quality of **data uploads** and query output

ASSIGNMENT	
DATABASE DESIGN	
Use a work-related dataset to design your own relational database. You should describe the dataset, follow the normalisation steps and create an Entity Relationship Diagram (ERD).	
Word Count	Max 1500 words
Deadline	3 weeks
Deliverables	Word Document, PowerPoint, Excel File, PDF, Lucid Chart

Additional Resources

Testing Strategies

Dealing with Problems

multiverse

Complete Session Attendance Log and Update Your OTJ