**DATA MODELING**

Developing a data model that shows a visual representation of data elements and the relationship between them.

A data model provides a visual representation of data elements within the database.

A data model shows how data is stored, accessed, updated and queried in your database.

# Course requirements

The Professional Certificates create opportunities so that anyone, regardless of education, background or experience, can learn high-quality skills to land a high-growth career. There is no degree or experience required to get started.

In order to succeed in this course, you'll need to make sure that you've completed the previous courses and installed the required software. The previous courses that you must complete include:

* [Introduction to databases](https://www.coursera.org/learn/introduction-to-databases/home/week/1)
* [Database structures and management with MySQL](https://www.coursera.org/learn/database-structures-and-management-with-mysql/home/week/1)
* [Advanced MySQL topics](https://www.coursera.org/learn/advanced-mysql-topics/home/week/1)
* [Programming in Python](https://www.coursera.org/learn/programming-in-python/home/week/1)
* [Database clients](https://www.coursera.org/learn/database-clients/home/week/1)

**Data model example**

## **Overview**

A data model provides a visual representation of data elements and illustrates how they relate to one another. It provides database engineers with an understanding of how data in a database is stored, accessed, updated and queried.

There are several types of data models that can be used to design a database system. Which model you choose should depend on its suitability in terms of supporting the needs of the business.

**Conceptual data model**

data model presents a high-level overview of the database system through a visual representation of its entities and their relationships.

**Dimensional data model**

A dimensional data model consists of two types of tables: fact tables and dimensional tables.

## **The Entity Relationship Model**

**E**ntity Relationship Entity Relationship Entity

Orders

Place

Contain

Clients

Entity

Can use the entity relationship model to assign attributes to the tables, or entities, in your database.

The entity relationship model is typically represented in an ER diagram that displays a database’s entities and the different relationships between them, where each entity has a set of related attributes.

This model also depicts different types of multiplicities including one-to-one, one-to-many, and many-to-many.

The following ER Diagram illustrates a surgery appointment booking system. It consists of four entities: Doctor, Surgery, Appointments and Patients.

Each entity has a set of attributes, and all entities are connected via foreign keys. According to this diagram:

* A patient can book one or many appointments.
* Multiple appointments could be assigned to each doctor.
* Multiple appointments take place at a specific surgery.

The entity relationship data model supports the surgery’s data requirements, and the ER diagram presents the database structure in an easy-to-use fashion, so you can use it to document your database and communicate its structure with your stakeholders.

You can also build this diagram in an integrated development tool such as MySQL Workbench and implement the database model directly in MySQL.

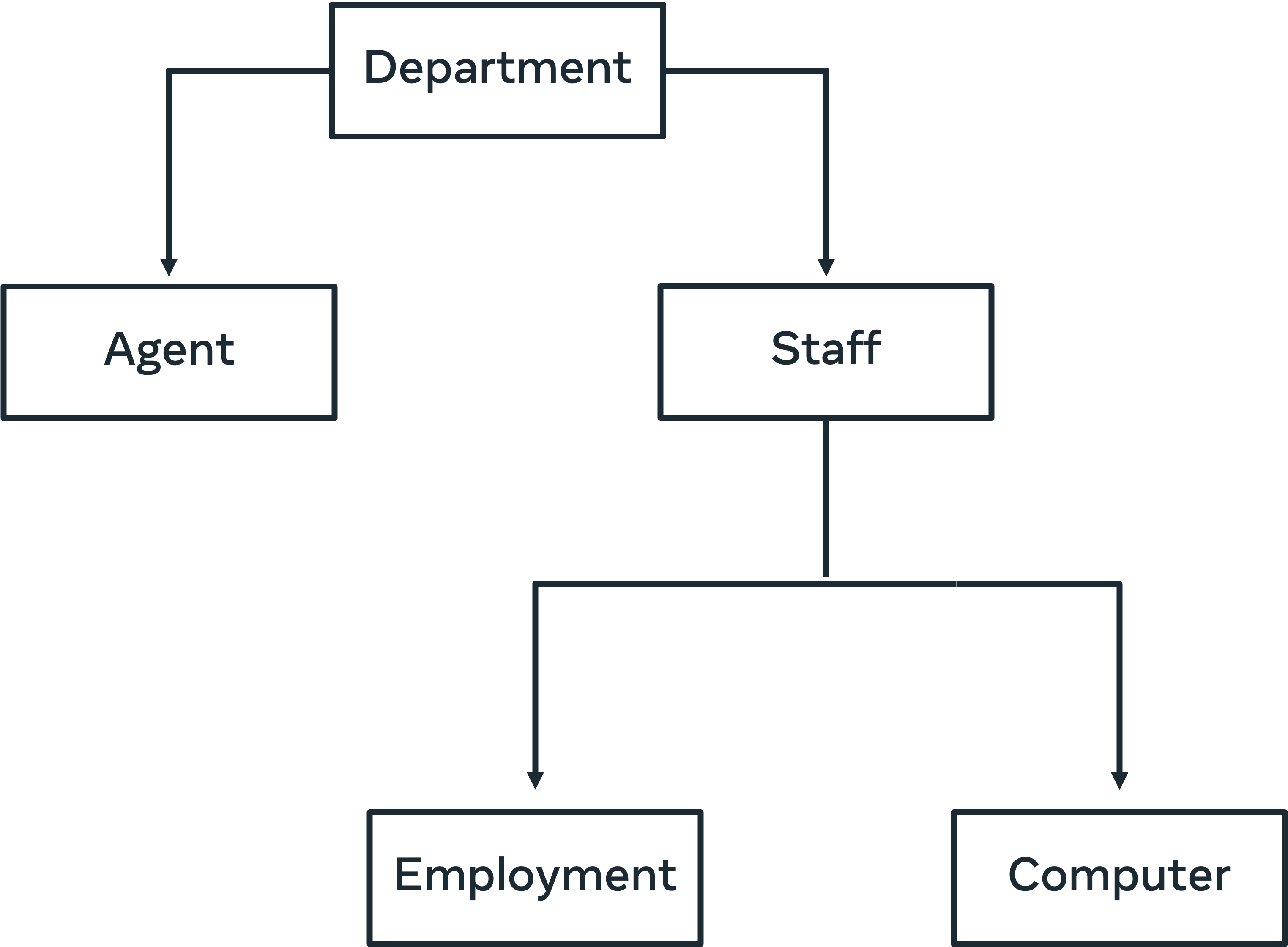
## **The hierarchical data model**

The hierarchical data model is organized in a tree-like or a parent-child structure, where each record of data has one parent node and can also have several child nodes.

With this model, you can easily find data or add and delete information. However, this model only permits one-to-many relationships between nodes, as each child node can have only one parent node.

The following data model illustrates the database structure of an insurance company. In this diagram the insurance company is structured as follows:

* Department is a root node,
* The Agent and Staff nodes are linked to the parent Department root node.
* And the Computer and Employment nodes are linked to the parent Staff node.

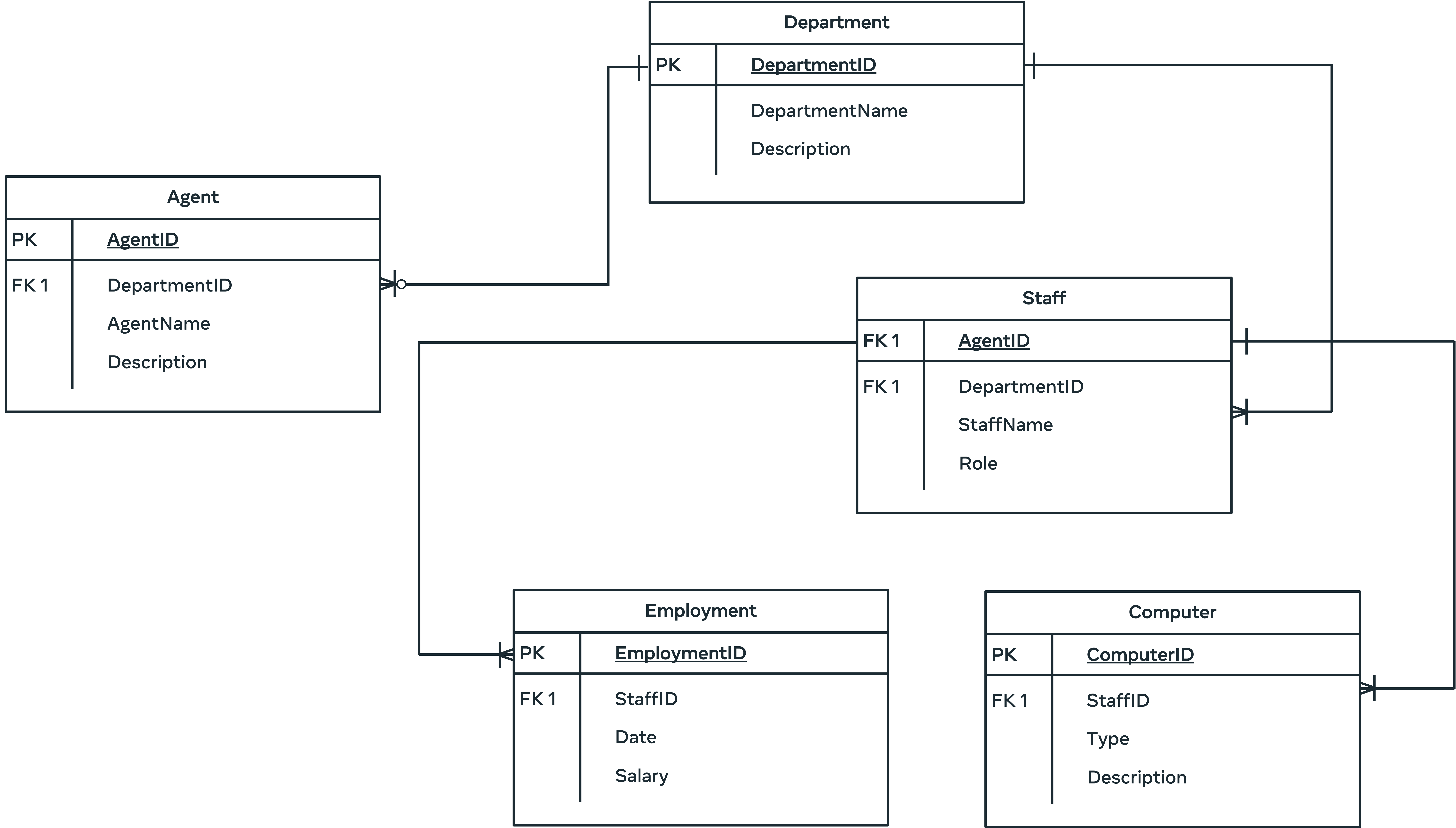


You may have noticed that a key advantage of this model is the simplicity of its structure. Also, since each record has only one parent, you can navigate through the database quickly and efficiently. However, you should be aware that, in this model, if you delete a parent node then the corresponding child records are also deleted.

In addition, this model does not support many-to-many relationships, which may be required in some cases (such as in the previous surgery scenario).

It’s also difficult to re-organize the structure of a database to incorporate new requirements. Doing so may disrupt the existing parent-child relationships.

You can use ER diagram to represent a more detailed level of this model as shown below.



The Object-Oriented data model

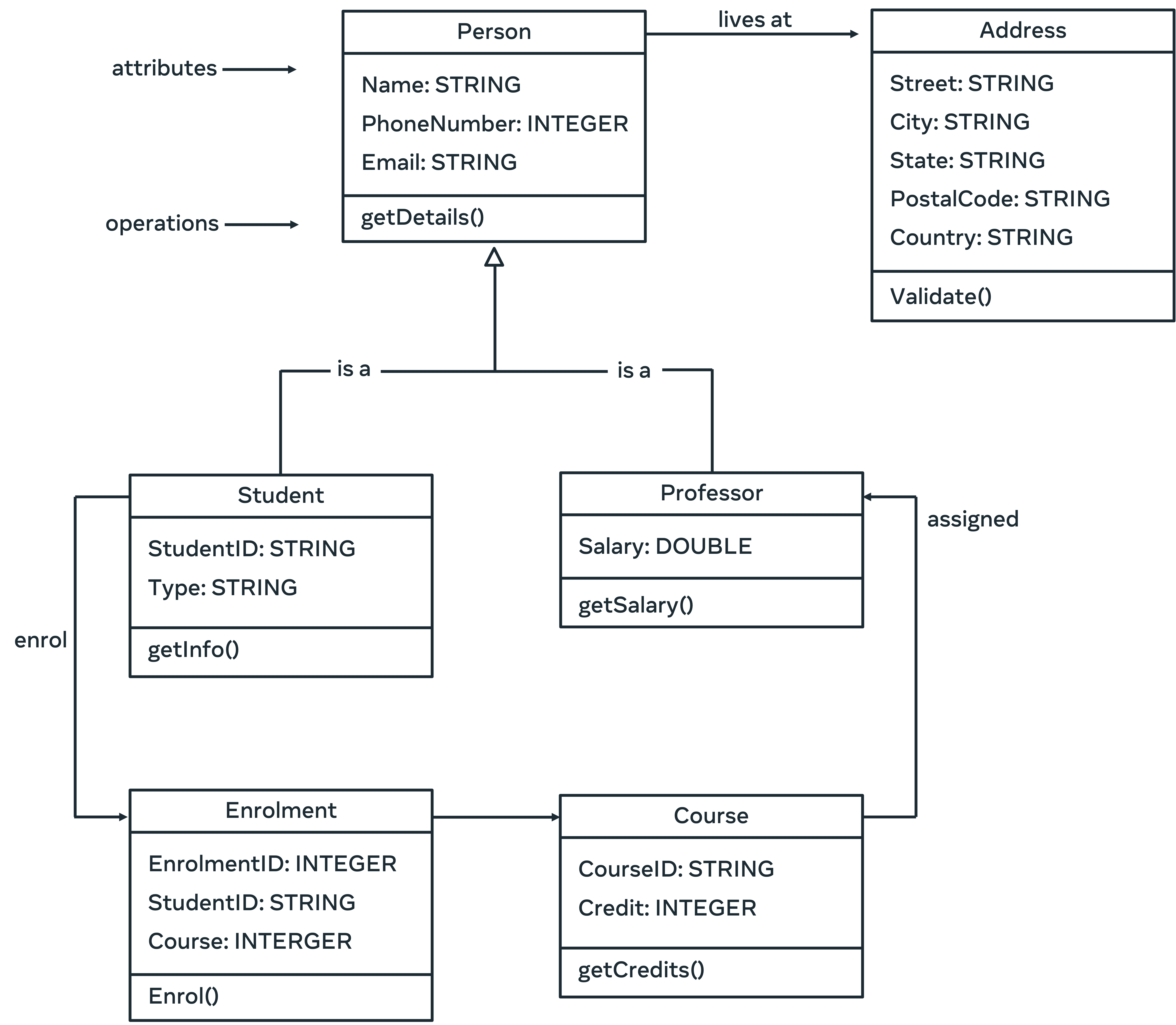
The Object-Oriented data model is based on the Object-Oriented concept, where each object is translated into a class that defines the respective object’s characteristics and behavior.

In this model you can use different types of associations between objects such as aggregations compositions and inheritance. These features make object-oriented databases suitable for complicated projects. However, making use of this model requires a good understanding of object-oriented programming.

In the following course enrollment diagram, there are six classes:

* Person,
* Student,
* Professor,
* Course,
* Address
* and Enrolment.

Each class consists of a set of attributes and methods.



In this example, each class represents similar objects in the database. This means that similar objects have the same characteristics and behavior. For example, each student registered in the database system has same set of attributes (or instance variables) such as id and type. Also, each object instance of a class utilizes the same operations stated in the class.

In addition, there are different types of associations defined in the diagram as follows:

* Each student and professor are a person (inheritance relationship) and each person is associated with an address.
* Each enrolment involves a student and a course. The enrolment class in this case is called association class.
* Each professor is assigned a course.

## **Conclusion**

This reading provided you with an overview of how to model databases using different types of data models through the use of relevant real-life examples.

**Normalization** is an important process used in database systems. It involves structuring tables in order to reduce data duplication, avoid data modification implications, and simplify data queries from the database.

**Insertion anomaly →** New data requires the insertion of additional data

**Update anomaly →** An update of a record results in further updates.

**Deletion anomaly →** A deletion of a record deletes more than one data set.

**Data atomicity**

Atomicity means that multiple operations can be grouped into a single logical entity, that is, other threads of control accessing the database will either see all of the changes or none of the changes.

An example of an atomic transaction is a monetary transfer from bank account A to account B.

**First normal form**

Enforces data atomicity and eliminates unnecessary repeating data groups.

**Second normal form**

A table cannot contain any relationships built on functional or partial dependency.

The table must be defined with a composite primary key.

**Third normal form**

Resolve issues of transitive dependency, when non-key attribute depend on one another.

# Additional resources

The following resources offer additional reading materials on the topic of database modeling.

* [IBM guide to data modeling](https://www.ibm.com/cloud/learn/data-modeling)
* [Microsoft guide to the basics of database normalization](https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description)
* [SAP guide to data modeling](https://www.sap.com/insights/what-is-data-modeling.html)

**Database modeling in MySQL Workbench**

A unified visual tool for database modeling and management

MySQL Workbench's forward engineer feature can be used to create a data model in MySQL Workbench, and then transform it into a SQL schema that can be implemented automatically in MySQL.

MySQL Workbench for data modeling and data management

This reading outlines the role of MySQL Workbench in database modeling and data management. It provides an overview of the main database tasks that can be performed in MySQL Workbench.

## **Overview**

MySQL Workbench (WB) is a unified visual tool for database modeling and database management. The WB was developed by Oracle, the company that manage MySQL. MySQL Workbench is a free, open-source and cross-platform tool that runs on Windows, Mac and Linux.

MySQL Workbench enables you to speed up the database design process by using the visual SQL editor. It also allows you to migrate data from different kinds of relational database systems.

## **Visual Database Design**

MySQL Workbench simplifies database design and maintenance. It enables database engineers to document the requirements in the form of visualized ER diagrams. This lets you communicate your design ideas with stakeholders in a professional way.

MySQL WB designer tool facilitates model-driven database development, which automates the development process without the need to code each part of the database system.

In addition, by using MySQL WB database engineers must design data models that meet the expected standards, otherwise they can’t implement these data models directly in MySQL.

In other words, MySQL WB allows you to develop a complete entity-relationship diagram with auto Layout and arrangement of data models.

## **Forward and Reverse Engineer Methods**

MySQL Workbench provides capabilities for creating data models in the visual designer. You can then use the Forward Engineer method to create a relevant data model in your MySQL server. The SQL schema is automatically generated for you. You can amend the code if required, or you can execute it with just a few mouse clicks.

MySQL Workbench also offers the Reverse Engineer method. You can use this method to create or import a MySQL database file and then generate a relevant data model from the SQL script. You can modify the model if needed, and then utilize the forward engineer method to create a new database system in MySQL.

## **Database Management**

Database management typically involves maintaining different versions of your database schemas. To help you with database management, MySQL WB can be used to perform Schema Synchronization and Comparison functionality. These features provide you with the ability to compare either two databases or a database and a data model, so you can visually detect the differences. MySQL WB also allows you to synchronize the data model with a live database and vice versa.

## **Database Documentation**

As a database engineer, you need to document the design of your database. This can be very time consuming. MySQL Workbench provides a documentation tool called DBDoc to document all details of your database design. You can use this tool to generate the documentation of your models in either HTML or plain text format.

## **The Visual SQL Editor**

MySQL WB visual SQL editor is used to write, run and debug SQL statements. It also facilitates SQL code completion and provides a context-sensitive list of SQL keywords and objects.

In addition, it includes a SQL code formatter that automatically formats SQL code to make it easier to read and edit. Other useful features include:

* SQL syntax highlighting,
* SQL code generation,
* The ability to reuse  your favorite SQL snippets,
* A running history of all executed SQL queries,
* And the ability to quickly create tables, indexes, views, stored procedures, triggers and functions.

## **Visual Database Administration**

With MySQL WB you have complete administrative control over all aspects of your MySQL server, including:

* User accounts,
* Roles,
* Privileges,
* Server start/stop,
* Logs,
* And server configuration.

## **Data Management**

MySQL WB provides useful features for managing data in MySQL database including the ability import and export mysqldump files, and export query results sets as CSV, XML, HTML files.

The WB Visual Data Editor also allows you to view and edit result sets in a grid table.

You can also view multiple result sets in the same visual data window. And you can perform data search and pattern matching searches across all tables and schema.

## **Database Migration**

In MySQL WB, you can use the Database Migration Wizard to perform data migration between different versions of MySQL servers.

In addition, you can also perform data migration with different relational database management systems such as Microsoft SQL Server, Sybase ASE, Sybase SQL Anywhere, PostgreSQL, and SQLite.

## **Conclusion**

MySQL Workbench has many useful tools that allow database developers to virtually create database models that can be easily transformed into physical MySQL databases.

Also, the WB allows MySQL administrators to configure and secure their database systems and assign users different types of roles and privileges.

In addition, it enables designers to use the visual editor to create database schemas, tables, views, stored procedures, functions and triggers.

# Additional resources

The following resources offer additional reading materials on the topic of data modeling and data management in MySQL Workbench:

* [Overview of process steps for installing MySQL Workbench](https://dev.mysql.com/doc/workbench/en/wb-installing.html)
* [Oracle guide to downloading and installing MySQL Workbench on MacOS](https://dev.mysql.com/doc/refman/8.0/en/macos-installation.html)
* [Oracle guide to getting started with MySQL Workbench](https://docs.oracle.com/cd/E19078-01/mysql/mysql-workbench/wb-getting-started-tutorial.html)
* [MySQL.com guide to visual database design in MySQL Workbench](https://www.mysql.com/products/workbench/design/)
* [Oracle guide to creating a data model](https://docs.oracle.com/cd/E19078-01/mysql/mysql-workbench/wb-getting-started-tutorial.html)

**------------------------->>>>>>>>>>WEEK2 <<<<<<<<←-----------------------------**

**Data warehouse**

Centralized data repository that aggregates, stores and processes large amount of data from multiple sources. It separates the data analysis workload from the standard transaction workload of a regular database management system. Users can then query this data to perform data analysis. This type of database is called online analytical processing or OLAP. A regular database focuses on collecting, storing, and processing data in real-time. It's also known as online transactional processing or OLTP.

**There are four key characteristics of a data warehouse.**

* **Subject-Oriented** → a data warehouse which explores one or more subject or topic areas.

When building a data warehouse, you need to choose one or more subject areas to explore. For example, global superstore can build a data warehouse that focuses on sales. They can then use the warehouse to find all relevant information on their sales processes.

* **Integrated** → a data warehouse which integrates data from a range of different sources.

This data must be integrated in a consistent format. Integrated data must also resolve issues such as naming conflicts and data types. Global superstores data warehouse integrates data from online purchases, website interactions, and social media.

* **Non-volatile** → a data warehouse must maintain data as it was loaded.

Non-volatile means data should not be deleted once it's loaded to the data warehouse. The purpose of a data warehouse is to analyze data as it exists. The more data you have, the better the results of your analysis.

* **Time variant** → Data warehouse must aggregate data over a long period of time for accuracy.

A data warehouse aggregates data over a long period so that it can measure changes in data over time. This helps users to discover trends, patterns, and relationships between data elements. For example, global superstore can use data from the last few years of sales to find out why their profits have declined.

* **Structured data** → data that’s structured within a well-defined data model.

They are structured data, semi-structured data, and unstructured data. Let's start with a look at structured data. This is data that's presented in a structured format within a well-defined data model. The relational database model is commonly used for structured data. The organized tables help users to access manage and search for data using sequel. A data warehouse typically uses structured data. This data type is organized for a specific purpose so it's easier to gain insights from and uncover answers to specific questions.

* **Semi-structured data →** partially structured data which requires more effort when performing data analysis. An example of semi structured data is an email message. It can contain structured data like a sender and subject, but the body is unstructured and can contain several different kinds of data like text, images, and videos.
* **Unstructured data →** data that doesn’t adhere to any specific pre-defined data model.

It can include any kind of data like text, video, or audio. This data can be collected and stored without applying any form of data model. But analyzing unstructured data requires advanced data analytics mechanisms like machine learning and data mining.

Semi-structured and unstructured data are more suited to a data lake. This is like a data warehouse but it can handle unstructured data. Data lake is used more widely by data scientists. Businesses prefer working with structured data and data warehouses because of its accuracy.

**Global super store’s data warehouse architecture**

Global super store have built a data warehouse architecture that you can explore.

**Data warehouse architecture**

The design of the data warehouse various components.

Interactive reports

Analytical reports

Static reports

Data analysis

Data mining

Static reporting

Data source

External data

Internal data

Operational data

Flat files

Data staging

Extract

Transform

Load

Data warehouse

Presentation

Interactive reports

Analytical reports

Static reports

External data

Internal data

Operational data

Flat files

Extract

Transform

Load

Interactive reports

Analytical reports

Static reports

Let's explore these components and find out more about how they contribute to the data analysis process. The first component of a data warehouses architecture, is the sources of data that it relies on for its insights. These include external sources like Global Super Stores, online surveys or social media data. Internal sources like information collected within the company database on customers and products. Operational data produced by day to day business activities like customer orders, and data sources can also include flat files. These are files without an internal structure like customers, online behavior or data log entries. Make sure that the data sources are accurate, so that you can avoid irrelevant or poor data analytics.

The next component is data staging, the data staging area includes a set of processes known as the ETL or extract, transform and load pipeline.

Data is stored in the data storage component. This is a central database repository that serves as the foundation of the data warehouse, it organizes data in relational databases. It also includes a metadata repository that holds different kinds of information about the data. Like, where it was sourced from the features of the data and the tables the data is stored in along with their attributes. What does metadata mean in the context of a data warehouse. Metadata is essentially a table of contents for the data in the data warehouse. It helps database engineers manage and keep track of the changes within their source systems, methods and processes. For example, Global Super Stores metadata contains information like where the data was sourced from. It also shows, when each file was created, who created it and other important information

The next component in the data warehouse is data marts, these are subject oriented databases that meet the demands of a specific group of users. Each mark contains a subset of data that focuses on particular parts of the business or organization. For example, Global Super Stores data marts relate to specific departments and business functions. They can use these mass to perform focused analytical processes on specific parts of the business.

Finally, once the data is ready, you can perform data analytics. Data analytics is performed using different analytics techniques like data mining. Once you've analyzed the data, you can then present it. The data can be presented in the form of reports like interactive reports, analytics, reports or static reports. Global Super Stores data analysts, can analyze the data within their repository using different techniques. They can then produce reports that provide information on sales, profits and other important aspects of the business.

**Data warehouse architecture best practices**

* Separate analytical and transactional operations
* Use scalable solutions to process large amount of data
* Build a flexible architecture for future upgrades
* Implement data security features
* Develop a simple, flexible architecture
* Create an easy-to-use warehouse
* Document the development of the data warehouse

# The data warehouse Extract, Transform and Load (ETL) process

This reading provides more details about the data warehouse ETL processes. It provides an overview of the process, an in-depth exploration of each process step, and a list of useful tools that you can use when working with a data warehouse.

## **Overview**

ETL is an acronym for “Extract, Transform, and Load”. It is an essential part of the data staging phase in the data warehouse architecture. ETL describes the set of processes that extract data from different source systems, transform the data into a suitable format, and load it into a data warehouse repository for data analytics.

Typically, data warehousing includes complex ETL processes and requires active involvement from stakeholders including data engineers, data analysts, business analysts, testers and managers.

As a database engineer, you should carefully evaluate any changes that impact the business in your design of ETL processes in a data warehouse. You must ensure a complete separation between the analytical operations and the day-to-day transactional operations. You also need a scalable solution that can process  huge amounts of data.

## **How does ETL work?**

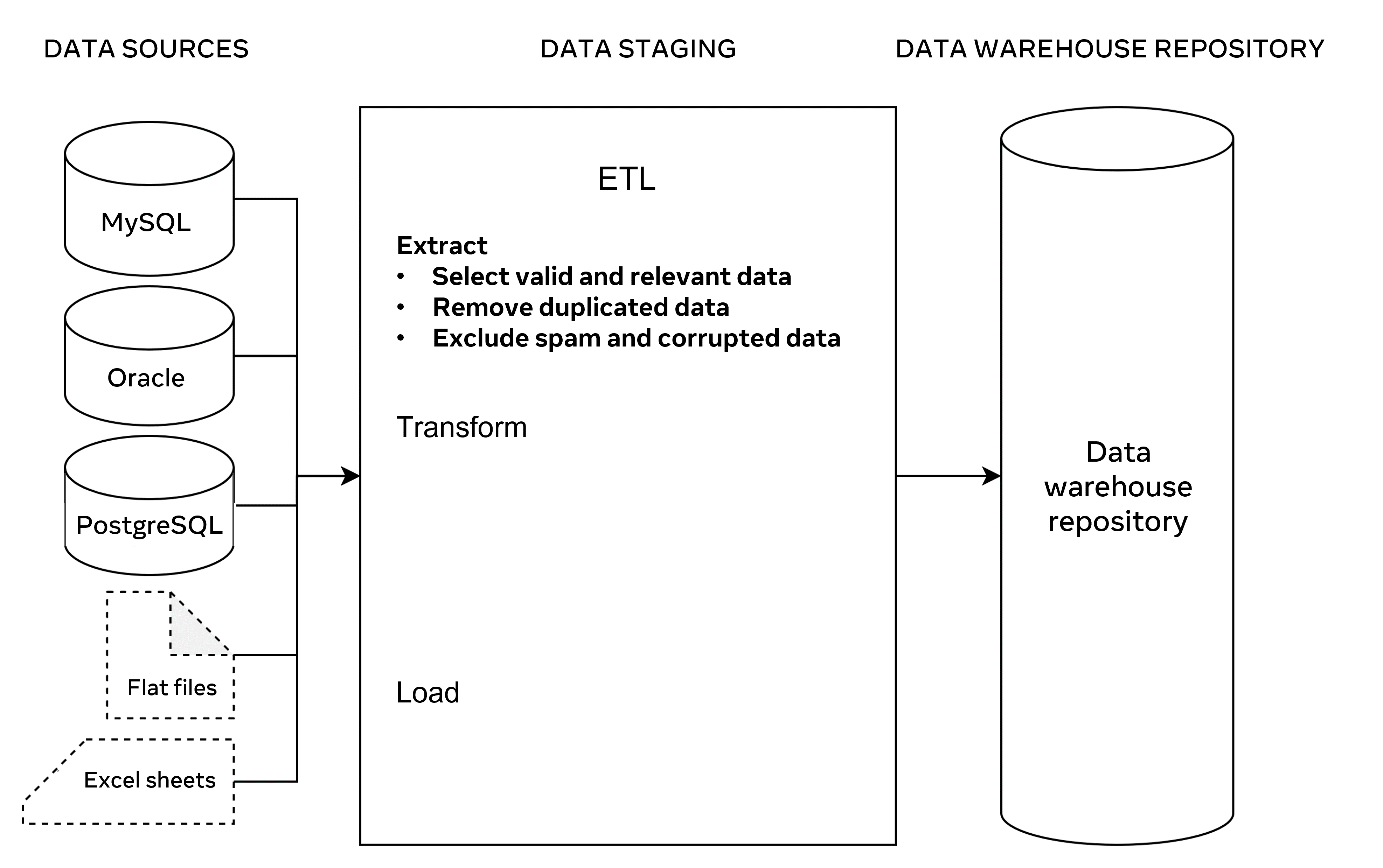
The ETL three step process represents the pipeline that moves data from its various sources and integrates it in a data warehouse. There are three steps in this process:

* Data extraction.
* Transformation.
* And Loading.

Let’s take a few minutes to explore each of these process steps in more detail.

## **Step 1: Data Extraction**

This step in the process involves extracting data from multiple sources into the staging area. For example,  Global Super Store may use different database management systems and files to maintain their data in the logistics, operations, sales and marketing departments. This could include MySQL, Oracle, PostgreSQL, Excel sheets and flat files as illustrated in the following diagram.



Global Super Store extract data from each of these sources in the staging area. When you design this process, it is important that you:

* Extract relevant data only to avoid data overloading.
* Include valid data and remove all duplications to obtain more accurate results.
* Exclude spam and corrupted data, as this will save a lot of effort during data cleaning and preparation.
* Validate data against the initial data in the sources to ensure it can be used in the same way in the data warehouse

## **Step 2: Transformation**

This process requires converting data from one format into another, or from one structure into another according to the predefined data model established in the data warehouse. This is a crucial step to integrate data in the data warehouse repository.

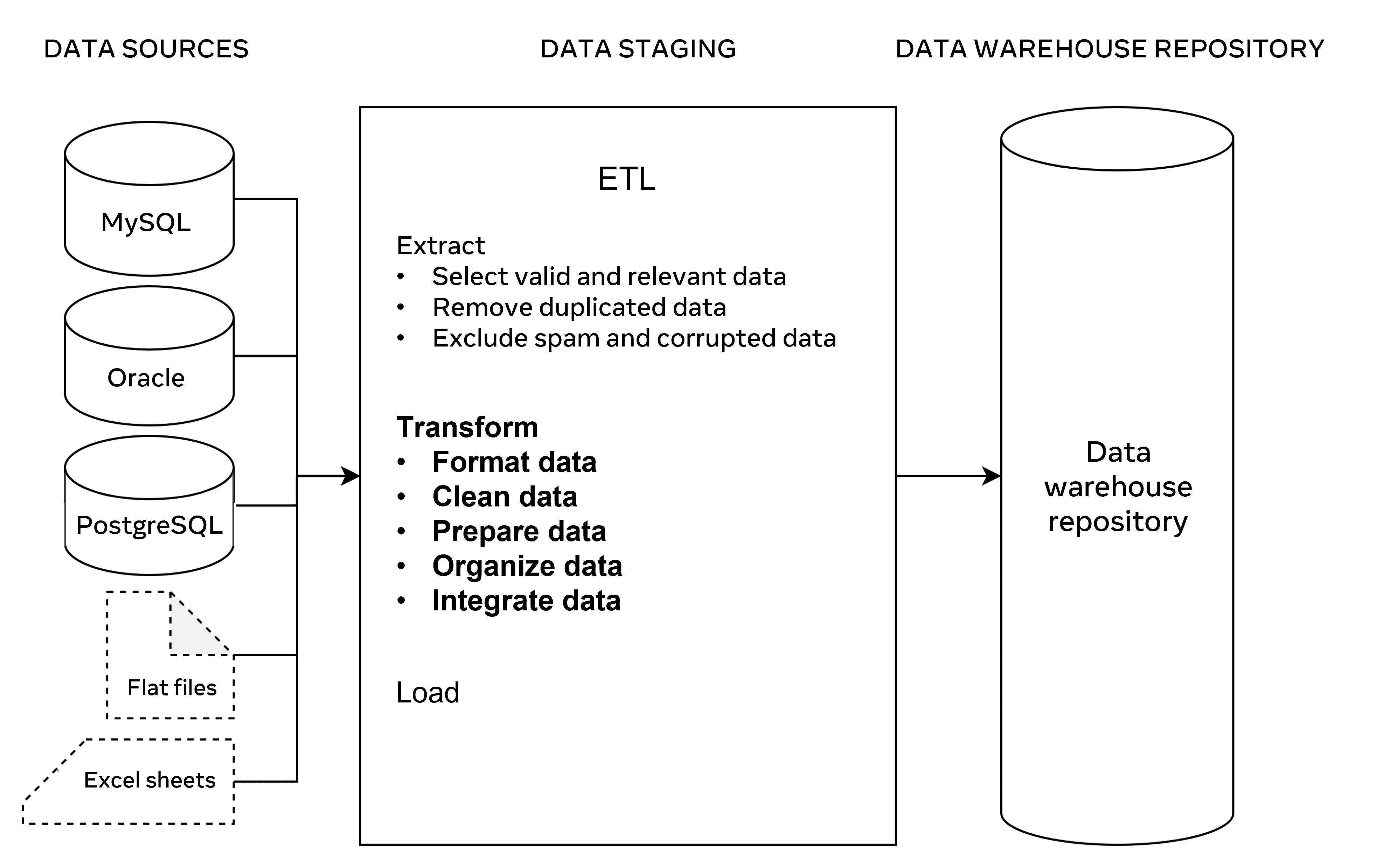
Data already built-in to the relational database may not necessarily require a transformation. It may be imported directly as is to the data warehouse repository. However, you may need to clean and prepare the data before you use it for data analytics.

For example, if a customer’s first name and last name values are stored separately in two different columns then they can be concatenated in a ‘full name’ field before they are loaded into the data warehouse repository. Or if data includes buying and selling costs, then you can create new calculated field for data profit.

Data transformation becomes more crucial when you work with raw data (like the atomic type of data that a computer cannot make useful information of it on its own). Good examples of raw data include computer logs, user navigation cookies and search histories. These types of data can be found in flat files, or in comma separated values (CSV) files.

In a data warehouse you should include structured data for data analytics. To do this, you should organize transformed data in tables, define data with specific datatypes, and set constraints in tables and columns. This creates records of related data that can be accessed, queried and analyzed to present useful information.

The main tasks in the transformation process include formatting, cleaning, preparing, organizing and integrating data, as illustrated in the following diagram.



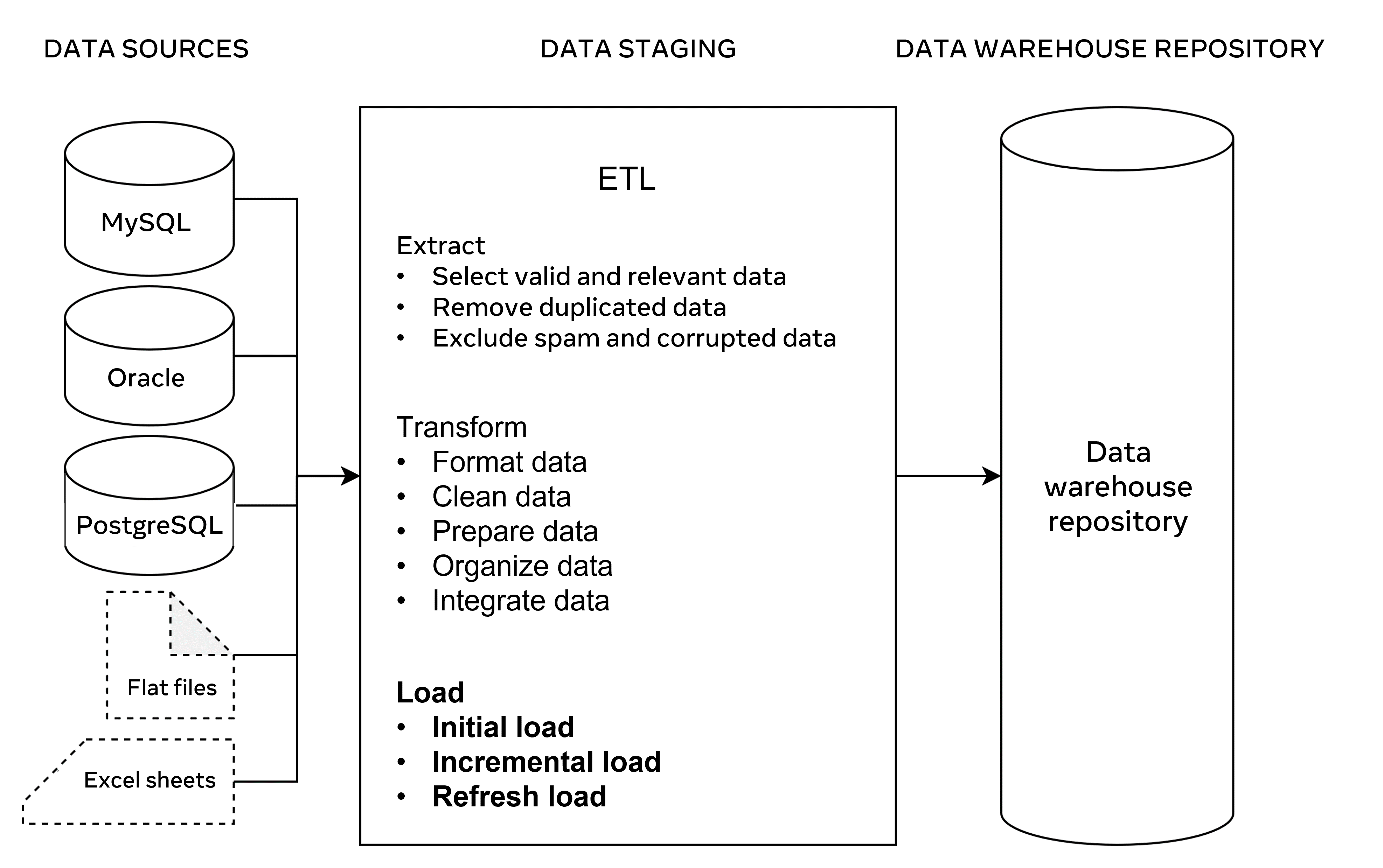
It is also important to ensure data integrity and consistency in the Transform phase. For example, a customer’s name may be spelled in different ways in different data sources such as John or Jon. Or a phone number could be recorded in different ways (For example, with and without country code). Dates and times could also be collected in different formats.

## **Step 3: Loading**

This process is the last step in ETL. In this step, large amounts of data are loaded into the data warehouse repository.  So, it’s important to deploy some recovery mechanisms in case of data loading failure. This means that you can resume the loading process from the point of failure without data loss and while also maintaining data integrity.

In this context, you can perform three types of data loading:

* Initial load: Populates extracted data to the data warehouse repository.
* Incremental load: Loads new data from the data sources to the current repository.
* Full refresh load: Erases the contents of the database in the data warehouse repository either partially or totally and re-populates it with fresh new data.



## **ETL Tools**

There are several tools available that you can use to carry out these ETL processes. These tools automate the extract, transform and load processes into an advanced level. Some examples of these tools include:

* Oracle: The industry-leading database that offers a wide range of choices of different data warehouse solutions.
* Amazon RedShift: A simple and cost-effective data warehousing tool that can be used to analyze all types of data.
* MarkLogic: A data warehousing solution that makes data integration easier and faster than other tools. It can query different types of data like search engine data, flat files and RDF.

## **Conclusion**

ETL provides a systematic approach for moving data from various data sources into a data warehouse. The data warehouse provides a central data repository that supports different types of data analytics and reporting. Therefore, a well-designed and documented ETL process is essential for developing an optimized and useful data warehouse.

# ETL testing

This reading provides an overview of ETL testing and best practices that should be followed during the testing process.

## **Overview**

A data warehouse provides a central data repository that supports advanced data analytics and reporting. ETL refers to the Extract, Transform, and Load processes required to move data from various data sources into a data warehouse.

To complete the ETL processes you need to:

* Extract relevant and valid data.
* Transform the extracted data into a suitable format and structure.
* And load the transformed data from various data sources into the data warehouse repository.

But how can you ensure that the loaded data is correct, valid, reliable, and consistent?

## **ETL testing**

The main purpose of ETL testing is to ensure accuracy, reliability, validity and consistency of loaded data. ETL testing investigates whether there are problems in ETL processes that prevent the creation of good quality data. This can be applied by carrying out the following actions:

* Mapping data in an appropriate way.
* Validating data.
* Checking duplicates of data.
* Validating dates.
* And verifying the completeness and correctness of data.

## **ETL testing in practice**

Global Super Store has many different departments, each of which manages customer data in a different way on different servers using different unique identifiers. For example, the sales department uses customer phone numbers as unique IDs, while the marketing department uses automatically generated customer IDs.

Also, certain data is represented in different ways. For example, customer full names are stated as one field in the sales department. But they’re stated in two fields in the marketing department (as first name and last name). The date in sales department is formatted as yyyy-m-d (for example 2020-9-5). Yet it's formatted as yy\m\d (or 20\9\5) in the marketing department.

This raises several problems for Global Super Store. A key issue that it can’t investigate which customers placed orders after their recent marketing campaign, because customer IDs are stored in different formats across the different departments. The solution to this problem is to create a new database model in the data warehouse. They can then extract relevant data, transform it and load it from the sales and marketing databases.

The new model could include:

* The Orders table with order ID, and order date.
* The Marketing table with marketing ID and marketing date.
* And the customer table with each customer’s full name, initial unique identifiers from both departments and a new unique key that enables the identification of customers from both departments.

This new model makes it possible to link the purchase history of customers with the marketing campaign, and then perform relevant data analysis.

## **Performing ETL testing**

To ensure that the data has been loaded correctly, ETL testing should be performed using the following methods.

## **Data mapping document**

Create a data mapping document to transform data from both department’s data sources to the destination repository in the data warehouse. This document should be reviewed, verified and validated by database engineers and data analysts to ensure correct data mapping.

## **Data validation**

Perform data validation by comparing data structure, datatypes, data formats and constraints in the related source and target tables against the corresponding mapping document. You should also perform data validation to verifying date values when creating new records of data, as this allows the data warehouse system to extract new data from the data sources and update the relevant repository databases in case of changes in data.

## **Data completeness**

Ensure data completeness by verifying that all expected data is loaded into the target tables including records counts and rejected records. You also need to check for any constraints that have been applied and ensured that all unique values have been populated as required.

## **Data correctness**

Perform data correctness to resolve any data naming conflicts, check that all misspelled data has been corrected, and make sure that any unexpected data values (in other words, values that are out of range) have been fixed or removed.

## **Data quality**

Ensure good data quality by checking names and numbers of formats, precision, and null values where NOT NULL constraint is specified.

## **Unique data**

Run a duplicate check to ensure there are unique records of data available in all tables and that all duplicate records of data have been removed.

## **Conclusion**

ETL testing aims to verify whether data is loaded from a data source to the target database in the data warehouse as expected. This ensures proper data analytics by removing duplicated data and excluding spam and corrupted data. It also validates data against the initial data in the data sources, changes the existing data format into the right format and loads correct and consistent data into the data warehouse.

# Additional resources

The following resources offer additional reading materials on the topic of data warehousing:

* [Oracle overview of key data warehousing concepts](https://docs.oracle.com/cd/B10501_01/server.920/a96520/concept.htm)
* [Astera guide to data types](https://www.astera.com/type/blog/structured-semi-structured-and-unstructured-data/)
* [Oracle guide to ETL](https://www.oracle.com/integration/what-is-etl/)
* [Oracle overview of data lakes](https://www.oracle.com/big-data/what-is-data-lake/)