Data Science from scratch

# Matrices & Linear Algebra Fundamentals

Linear algebra topics:

* Vectors
* Matrices
* Transpose of a matrix
* Inverse of a matrix
* Determinant of a matrix
* Trace of a matrix
* Covariance Matrix
* Dot product
* Eigenvalues
* Eigenvectors

# Hash Functions, Binary Trees

Hashing is a technique that is used to uniquely identify a specific object from a group of similar objects. Some examples of how hashing is used in our lives include:

* In universities, each student is assigned a unique roll number that can be used to retrieve information about them.
* In libraries, each book is assigned a unique number that can be used to determine information about the book, such as its exact position in the library or the users it has been issued to etc.

In both these examples the students and books were hashed to a unique number.

Diagram

Description automatically generatedWith binary trees you can do a lot of operations like transversals, summations, construction, conversions, etc.

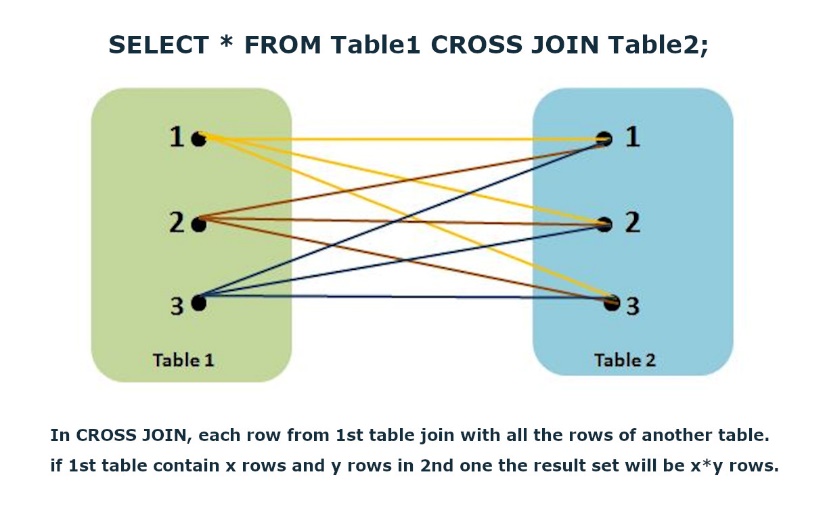
# Relational Algebra and DB basics

 Relational algebra query operations are performed recursively on a relation. The output of these operations is a new relation, which might be formed from one or more input relations.

The relational databases are tabular data that are interrelated.



# Inner, Outer, Cross and Theta Join.



Table

Description automatically generated

# CAP Theorem

* Consistency
* Availability
* Partition Tolerance

There’s a trade off between consistency and availability when the system has partitions. Partitioning is when the system can’t communicate each part with each other. If the system never has partitions, you can make the system consistent and available.

In real world you have degrees of consistency and degrees of availability and make trade offs between those two.

# Sharding

Sharding is when you have an enormous amount of data and you want to access them in the fastest way. This process is by fragmentating the data into small pieces and storing them in different servers to speed up the performance.

One way to access to every piece is by hierarchical sharding. Which means that you cut your piece into another small pieces and so on.

NoSQL uses sharding internally.

# OLAP

Diagram, engineering drawing

Description automatically generatedOLAP enables fast, flexible multidimensional data analysis for business intelligence (BI) and decision support applications.

OLAP (for online analytical processing) is software for performing multidimensional analysis at high speeds on large volumes of data from a data warehouse, data mart, or some other unified, centralized data store.

In theory, a cube can contain an infinite number of layers. (An OLAP cube representing more than three dimensions is sometimes called a hypercube.) And smaller cubes can exist within layers—for example, each store layer could contain cubes arranging sales by salesperson and product. In practice, data analysts will create OLAP cubes containing just the layers they need, for optimal analysis and performance.

**OLAP cubes enable four basic types of multidimensional data analysis:**

**Drill-down**

The drill-down operation converts less-detailed data into more-detailed data through one of two methods—moving down in the concept hierarchy or adding a new dimension to the cube. For example, if you view sales data for an organization’s calendar or fiscal quarter, you can drill-down to see sales for each month, **moving down in the concept hierarchy of the “time” dimension.**

**Roll up**

**Roll up is the opposite of the drill-down function**—it aggregates data on an OLAP cube by moving up in the concept hierarchy or by reducing the number of dimensions. For example, you could move up in the concept hierarchy of the “location” dimension by viewing each country's data, rather than each city.

**Slice and dice**

The slice operation **creates a sub-cube by selecting a single dimension from the main OLAP cube**. For example, you can perform a slice by highlighting all data for the organization's first fiscal or calendar quarter (time dimension).

The dice operation isolates a **sub-cube by selecting several dimensions within the main OLAP cube**. For example, you could perform a dice operation by highlighting all data by an organization’s calendar or fiscal quarters (time dimension) and within the U.S. and Canada (location dimension).

**Pivot**

**The pivot function rotates the current cube view to display a new representation of the data—enabling dynamic multidimensional views of data.** The OLAP pivot function is comparable to the pivot table feature in spreadsheet software, such as Microsoft Excel, but while pivot tables in Excel can be challenging, OLAP pivots are relatively easier to use (less expertise is required) and have a faster response time and query performance.

# Multidimensional Data Model

The multi-Dimensional Data Model is a method which is used for ordering data in the database along with good arrangement and assembling of the contents in the database.

A screenshot of a computer

Description automatically generated with medium confidenceThe Multi-Dimensional Data Model allows customers to interrogate analytical questions associated with market or business trends, unlike relational databases which allow customers to access data in the form of queries. They allow users to rapidly receive answers to the requests which they made by creating and examining the data comparatively fast.

**OLAP (online analytical processing) and data warehousing uses multi-dimensional databases.** It is used to show multiple dimensions of the data to users.

It represents data in the form of data cubes. Data cubes allow to model and view the data from many dimensions and perspectives. It is defined by dimensions and facts and is represented by a fact table. Facts are numerical measures and fact tables contain measures of the related dimensional tables or names of the facts.

The following stages should be followed by every project for building a Multi-Dimensional Data Model:

**Stage 1: Assembling data from the client:** In first stage, a Multi-Dimensional Data Model collects correct data from the client. Mostly, software professionals provide simplicity to the client about the range of data which can be gained with the selected technology and collect the complete data in detail.

**Stage 2: Grouping different segments of the system:**In the second stage, the Multi-Dimensional Data Model recognizes and classifies all the data to the respective section they belong to and builds it problem-free to apply step by step.

**Stage 3: Noticing the different proportions:** In the third stage, it is the basis on which the design of the system is based. In this stage, the main factors are recognized according to the user’s point of view. These factors are also known as “Dimensions”.

**Stage 4: Preparing the actual-time factors and their respective qualities:** In the fourth stage, the factors which are recognized in the previous step are used further for identifying the related qualities. These qualities are also known as **“attributes”** in the database.

**Stage 5: Finding the actuality of factors which are listed previously and their qualities:**In the fifth stage,A Multi-Dimensional Data Model separates and differentiates the actuality from the factors which are collected by it. These play a significant role in the arrangement of a Multi-Dimensional Data Model.

Stage 6: Building the Schema to place the data, with respect to the information collected from the steps above: In the sixth stage, on the basis of the data, which was collected previously, a Schema is built. M

**Advantages of Multi-Dimensional Data Model**

* It is easy to maintain.
* Its performance is better than that of normal databases (e.g. relational databases).
* The representation of data is better than traditional databases. That is because the multi-dimensional databases are multi-viewed and carry different types of factors.
* It is workable on complex systems and applications, contrary to the simple one-dimensional database systems.
* The compatibility in this type of database is an upliftment for projects having lower bandwidth for maintenance staff.

**Disadvantages of Multi-Dimensional Data Model**

* The multi-dimensional Data Model is slightly complicated in nature, and it requires professionals to recognize and examine the data in the database.
* During the work of a Multi-Dimensional Data Model, when the system caches, there is a great effect on the working of the system.
* It is complicated in nature due to which the databases are generally dynamic in design.
* The path to achieving the end product is complicated most of the time.
* As the Multi-Dimensional Data Model has complicated systems, databases have many databases due to which the system is very insecure when there is a security break.

# ETL

ETL is a type of data integration that refers to the three steps (extract, transform, load) used to blend data from multiple sources. **It's often used to build a data warehouse**. During this process, data is taken (extracted) from a source system, converted (transformed) into a format that can be analyzed, and stored (loaded) into a data warehouse or other system. Extract, load, transform (ELT) is an alternate but related approach designed to push processing down to the database for improved performance.

# JSON & XML

JSON:

* Stands for **J**ava**S**cript **O**bject **N**otation
* Is a lightweight format for storing and transporting data
* Is often used when data is sent from a server to a web page
* Is "self-describing" and easy to understand

Graphical user interface, text

Description automatically generatedJSON example:

XML:

* stands for eXtensible Markup Language
* Is a markup language much like HTML
* Was designed to store and transport data
* Graphical user interface, text, application

  Description automatically generatedWas designed to be self-descriptive

Graphical user interface, text

Description automatically generated with medium confidenceXML Example: is displayed as 🡪

# NoSQL

NoSQL databases are purpose built for specific data models and have flexible schemas for building modern applications. NoSQL databases are widely recognized for their ease of development, functionality, and performance at scale. These types of databases are optimized specifically for applications that require large data volume, low latency, and flexible data models, which are achieved by relaxing some of the data consistency restrictions of other databases.

In a NoSQL database, a book record is usually stored as a JSON document. For each book, the item, ISBN, Book Title, Edition Number, Author Name, and AuthorID are stored as attributes in a single document. In this model, data is optimized for intuitive development and horizontal scalability.

NoSQL databases are a great fit for many modern applications such as mobile, web, and gaming that require flexible, scalable, high-performance, and highly functional databases to provide great user experiences.

* Flexibility: NoSQL databases generally provide flexible schemas that enable faster and more iterative development. The flexible data model makes **NoSQL databases ideal for semi-structured and unstructured data.**
* Scalability: NoSQL databases are generally designed to scale out by using distributed clusters of hardware instead of scaling up by adding expensive and robust servers. Some cloud providers handle these operations behind-the-scenes as a fully managed service.
* High-performance: NoSQL database are optimized for specific data models and access patterns that enable higher performance than trying to accomplish similar functionality with relational databases.
* Highly functional: NoSQL databases provide highly functional APIs and data types that are purpose built for each of their respective data models.

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated with medium confidence

# Regex

STATISTIC

There is a dataset where I analyze all the statistics below ()

A fondo:

Eigen values for data science