

INFOLINK-EXP

**PRICESPIDER**

Introducing client's products

# WHAT'S PRICESPIDER?



PriceSpider is a software platform that provides businesses with real-time competitive pricing and product information for a wide range of products. It is designed to help companies optimize their pricing strategies, increase their sales and profits, and enhance their overall market competitiveness.

PriceSpider's technology crawls the internet and collects data from various sources such as retailer websites, marketplaces, and social media platforms. The platform then analyzes this data using advanced algorithms and machine learning techniques to provide businesses with insights into market trends, consumer behavior, and competitor pricing.

In addition to providing real-time competitive pricing and product information, PriceSpider also offers a range of other features, including dynamic pricing tools, brand protection services, and analytics and reporting capabilities. These features help businesses to make data-driven decisions and optimize their pricing strategies for maximum profitability.

# EXAMPLE

Let's say you run an online retail business that sells electronics, including smartphones. You want to optimize your pricing strategy to stay competitive in the market and maximize your profits.

With PriceSpider, you can track the prices of smartphones from your competitors, as well as the prices of the same models on different marketplaces, such as Amazon or eBay. You can also track the prices of the same smartphone models sold by different retailers, both online and offline.

Based on this data, PriceSpider can give you insights into the most popular smartphone models, the average selling price, and the range of prices across different channels. You can then use this information to adjust your pricing strategy and offer more competitive prices to attract more customers and increase your sales.

PriceSpider also offers dynamic pricing tools that can automatically adjust your prices based on real-time market conditions, such as changes in competitor pricing or demand. This can help you stay ahead of the competition and improve your margins.

In summary, PriceSpider can help you gather and analyze real-time competitive pricing and product information, which can be used to optimize your pricing strategy and increase your sales and profits.

# HOW?

PriceSpider uses a combination of advanced data collection, machine learning, and natural language processing techniques to gather and analyze product data from a variety of sources, such as online marketplaces, retailer websites, and social media platforms.

PriceSpider's algorithm employs techniques such as web crawling, data extraction, and data normalization to collect and clean the product data from these sources. The algorithm then uses machine learning techniques to analyze this data and identify patterns and trends in the market, such as changes in consumer behavior, pricing trends, and product availability.

The algorithm also employs natural language processing (NLP) techniques to analyze product descriptions and user reviews to gain insights into consumer sentiment and preferences. This allows PriceSpider to provide businesses with a more holistic view of the market and helps them to make more informed pricing decisions.

Overall, the algorithm behind PriceSpider is complex and sophisticated, and it relies on advanced technologies to gather, analyze, and interpret product data in real-time.

# UPC VS. SKU

In summary, UPCs are used to identify products universally and are mainly used for scanning at the point of sale, while SKUs are used by retailers to manage their inventory and track sales.

**WEB INTERACTION**



# Web Navigation

It's the action of navigation a network of information resources in the World Wide Web (WWW), which is organized as hypertext. The user interface that is used to do so is called a web browser.

The Google logo, featuring the word "Google" in its characteristic multi-colored font (blue, red, yellow, blue, green, red).The Bing logo, featuring a green chevron icon followed by the word "Bing" in a green sans-serif font.The Yahoo! logo, featuring the word "yahoo!" in a bold, purple, lowercase sans-serif font.

# Web browser

Is an application for accessing websites where the browser retrieves its files from a web server and displays the page from a particular site and displays the page on the user screen.

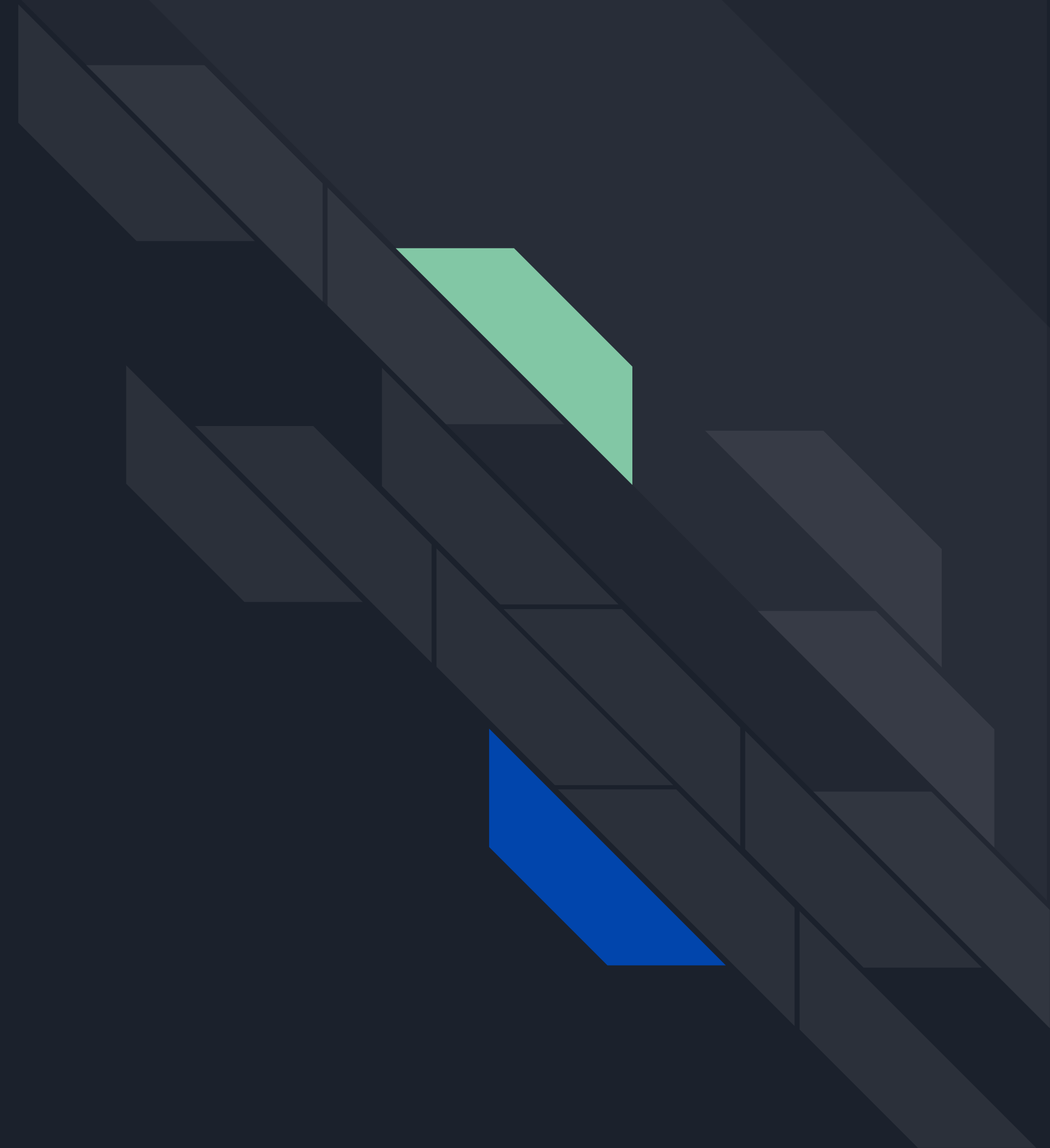
Its purpose is to fetch content from the WWW or from local storage.

This process begins when all URLs are retrieved using a Hypertext Transfer Protocol (HTTP), a set of rules for transfer data.





URL



## What is a URL:

URL stands for *Uniform Resource Locator*. A URL is nothing more than the address of a given unique resource on the Web. A domain name is part of a URL. You can see the visual difference in the following examples

**DOMAIN NAME**

**http://www.verisign.com/domain-names/online/index.shtml**

**URL**

**http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#SomewhereInTheDocument**

Authority

Scheme Domain Name Port Path to the file Parameters Anchor



# URL

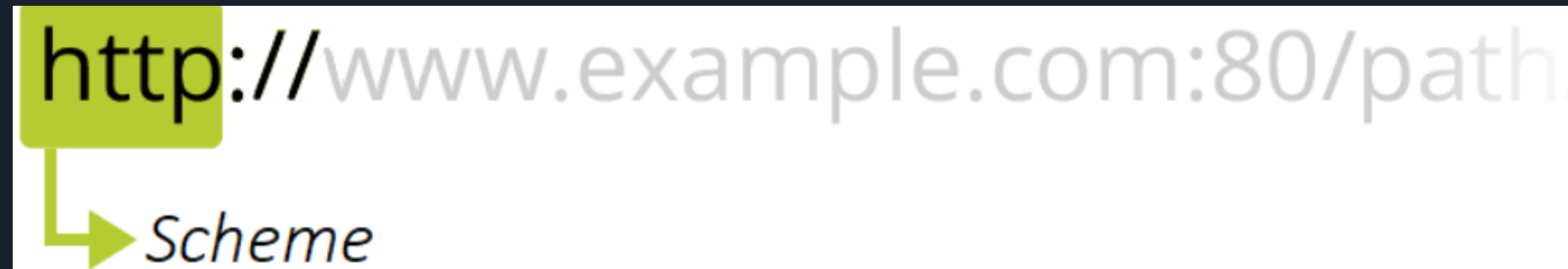
For computer networks and servers to “talk to one another,” on a language made up of numbers and letters called an IP address. Every device that connects to the internet has a unique IP address and looks like this:

**22.231.113.64 or 3ffe:1900:4545:3:200:f8ff:fe21:67cf**

To navigate easily around the web, typing in a long IP address isn’t ideal, or realistic, for an online user. This is the reason why domain names were created. You could consider the domain name as a “nickname” for the IP address.

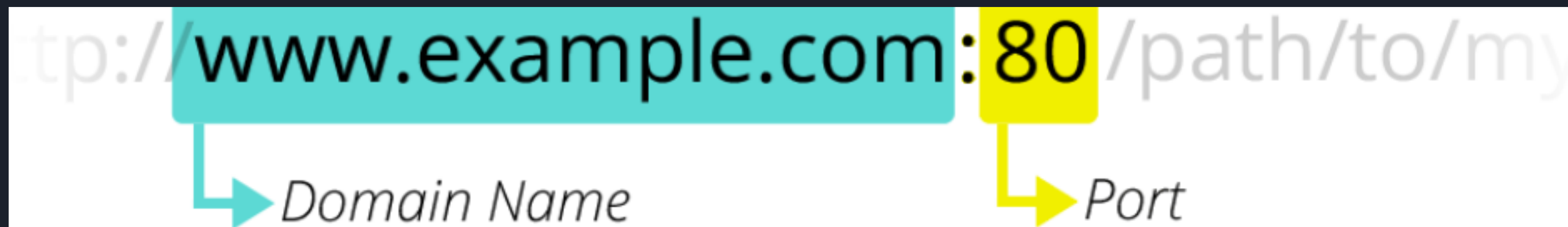
# Anatomy of a URL

The first part of the URL is the *scheme*, which indicates the protocol that the browser must use to request the resource



The diagram shows the URL `http://www.example.com:80/path`. The `http` part is highlighted with a green box. A green arrow points from the bottom of this box to the word *Scheme*.

the *authority*, which is separated from the scheme by the character pattern `://`. If present the authority includes both the *domain* (e.g. `www.example.com`) and the *port* (80), separated by a colon:



The diagram shows the URL `http://www.example.com:80/path/to/my`. The `www.example.com` part is highlighted with a teal box, and the `80` part is highlighted with a yellow box. A teal arrow points from the bottom of the teal box to the words *Domain Name*, and a yellow arrow points from the bottom of the yellow box to the word *Port*.

# Anatomy of a URL

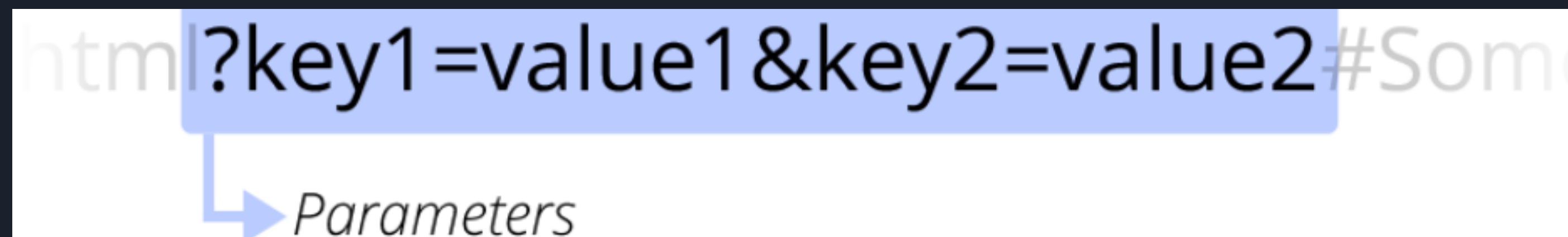
A path like this represented a physical file location on the Web server. Nowadays, it is mostly an abstraction handled by Web servers without any physical reality.



n:80/path/to/myfile.html?key1=value1

Path to resource

Those parameters are a list of key/value pairs separated with the & symbol. The Web server can use those parameters to do extra stuff before returning the resource.



html?key1=value1&key2=value2#Some

Parameters




# Anatomy of a URL

An anchor represents a sort of "bookmark" inside the resource, giving the browser the directions to show the content located at that "bookmarked" spot. On an HTML document, for example, the browser will scroll to the point where the anchor is defined;

lue2#SomewhereInTheDocument



**HTML**




HTML stands for **HyperText Markup Language**. It is a standard markup language for web page creation. It allows the creation and structure of sections, paragraphs, and links using HTML elements (the building blocks of a web page) such as tags and attributes.



HTML has a lot of use cases, namely:

- **Web development.** Developers use HTML code to design how a browser displays web page elements, such as text, hyperlinks, and media files.
- **Internet navigation.** Users can easily navigate and insert links between related pages and websites as HTML is heavily used to embed hyperlinks.
- **Web documentation.** HTML makes it possible to organize and format documents, similarly to Microsoft Word.

It's also worth noting that HTML is not considered a programming language as it can't create dynamic functionality. It is now considered an official web standard. The [World Wide Web Consortium \(W3C\)](#) maintains and develops HTML specifications, along with providing regular updates.



The average website includes several different HTML pages. For instance, a home page, an about page, and a contact page would all have separate HTML files.

HTML documents are files that end with a .html or .htm extension. A web browser reads the HTML file and renders its content so that internet users can view it.

All HTML pages have a series of HTML elements, consisting of a set of tags and attributes.

HTML elements are the building blocks of a web page. A tag tells the web browser where an element begins and ends, whereas an attribute describes the characteristics of an element.






The three main parts of an element are:

- Opening tag – used to state where an element starts to take effect. The tag is wrapped with opening and closing angle brackets. For example, use the start tag `<p>` to create a paragraph.
- Content – this is the output that other users see.
- Closing tag – the same as the opening tag, but with a forward slash before the element name. For example, `</p>` to end a paragraph.

The combination of these three parts will create an HTML element:


```
<p>This is how you add a paragraph in HTML.</p>
```

Another critical part of an HTML element is its attribute, which has two sections – a name and attribute value. The name identifies the additional information that a user wants to add, while the attribute value gives further specifications.



For example, a style element adding the color purple and the font-family verdana will look like this:

```
<p style="color:purple;font-family:verdana">This is how you add a paragraph  
in HTML.</p>
```



Another attribute, the HTML class, is most important for development and programming. The class attribute adds style information that can work on different elements with the same class value.


For example, we will use the same style for a heading `<h1>` and a paragraph `<p>`. The style includes background color, text color, border, margin, and padding, under the class `.important`. To achieve the same style between `<h1>` and `<p>`, add `class="important"` after each start tag:

```
<html>
<head>
<style>
  .important {
    background-color: blue;
    color: white;
    border: 2px solid black;
    margin: 2px;
    padding: 2px;
  }
</style>
</head>
<body>

<h1 class="important">This is a heading</h1>
<p class="important">This is a paragraph.</p>

</body>
</html>
```






Most elements have an opening and a closing tag, but some elements do not need closing tags to work, such as empty elements. These elements do not use an end tag because they do not have content:

```


```

This image tag has two attributes – an src attribute, the image path, and an alt attribute, the descriptive text. However, it does not have content nor an end tag.



Lastly, every HTML document must start with a `<!DOCTYPE>` declaration to inform the web browser about the document type. With HTML5, the doctype HTML public declaration will be:

```
<!DOCTYPE html>
```



Currently, there are 142 HTML tags available that allow for the creation of various elements. Even though modern browsers no longer support some of these tags, learning all the different elements available is still beneficial. This section will discuss the most-used HTML tags and two main elements – block-level elements and inline elements.



## ***Block-Level Elements***

A block-level element takes up the entire width of a page. It always starts a new line in the document. For example, a heading element will be in a separate line from a paragraph element.

Every HTML page uses these three tags:

- `<html>` tag is the root element that defines the whole HTML document.
- `<head>` tag holds meta information such as the page's title and charset.
- `<body>` tag encloses all the content that appears on the page.

```
<html>
  <head>
    <!-- META INFORMATION -->
  </head>
  <body>
    <!-- PAGE CONTENT -->
  </body>
</html>
```

Other popular block-level tags include:

- Heading tags – these range from `<h1>` to `<h6>`, where heading h1 is largest in size, getting smaller as they move up to h6.
- Paragraph tags – are all enclosed by using the `<p>` tag.
- List tags – have different variations. Use the `<ol>` tag for an ordered list, and use `<ul>` for an unordered list. Then, enclose individual list items using the `<li>` tag.





## *Inline Elements*

An inline element formats the inner content of block-level elements, such as adding links and emphasized strings. Inline elements are most commonly used to format text without breaking the flow of the content.

For example, a `<strong>` tag would render an element in bold, whereas the `<em>` tag would show it in italics. Hyperlinks are also inline elements that use an `<a>` tag and an href attribute to indicate the link's destination

```
<a href="https://example.com/">Click me!</a>
```

HTML is the primary markup language found on the internet.

Every HTML page has a series of elements that create the content structure of a web page or application.

HTML is a beginner-friendly language with plenty of support and is mainly used for static website pages. HTML works best together with CSS for the styling and JavaScript for the functionality.

[Reference guide to HTML](#)

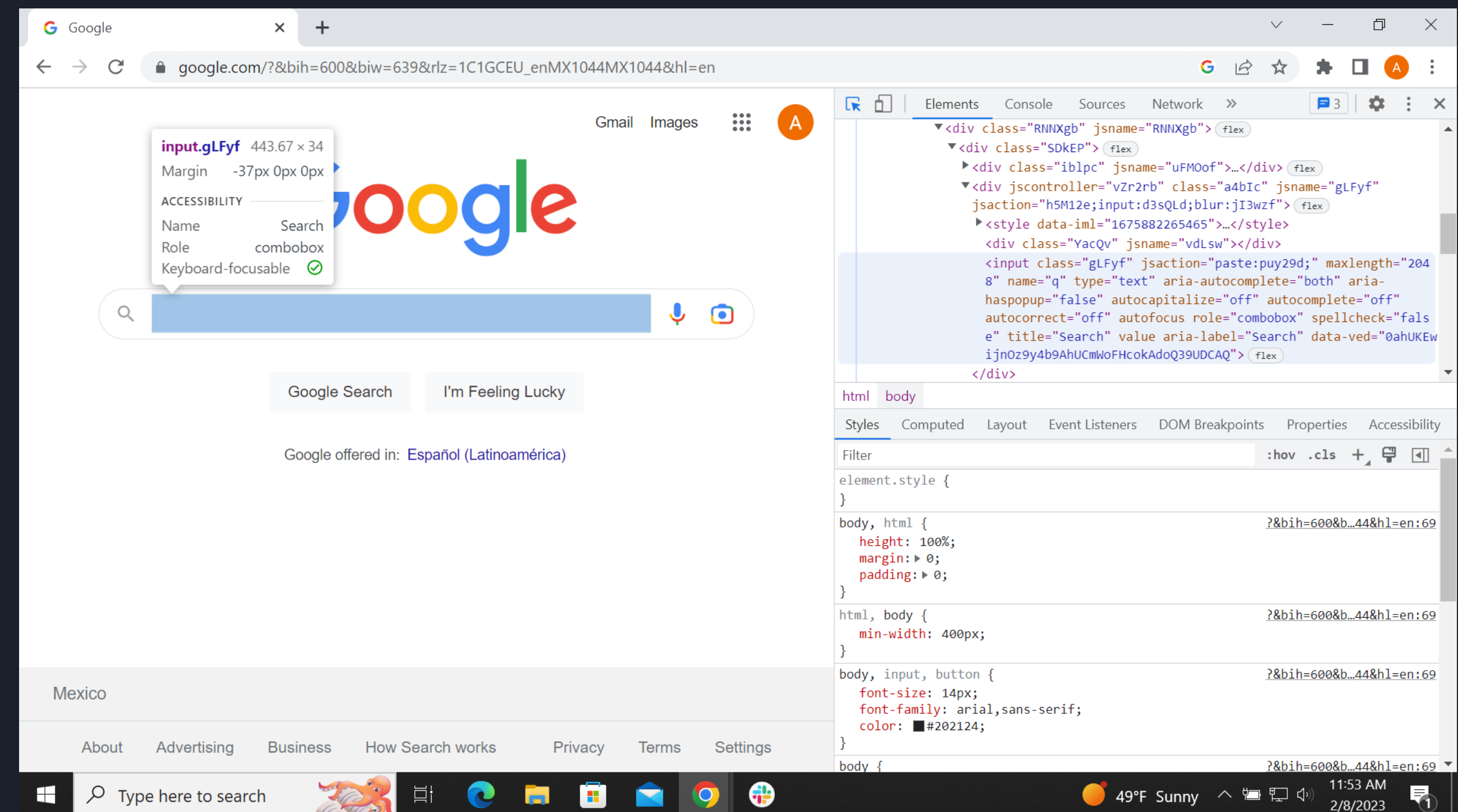
# PAGE INSPECTION

# Page Inspection

Enables you to get details about a page, design, elements, and source behind data displays. It is ideal for learning the data model behind a page.

You can use Ctrl + Shift = I

And/Or F12

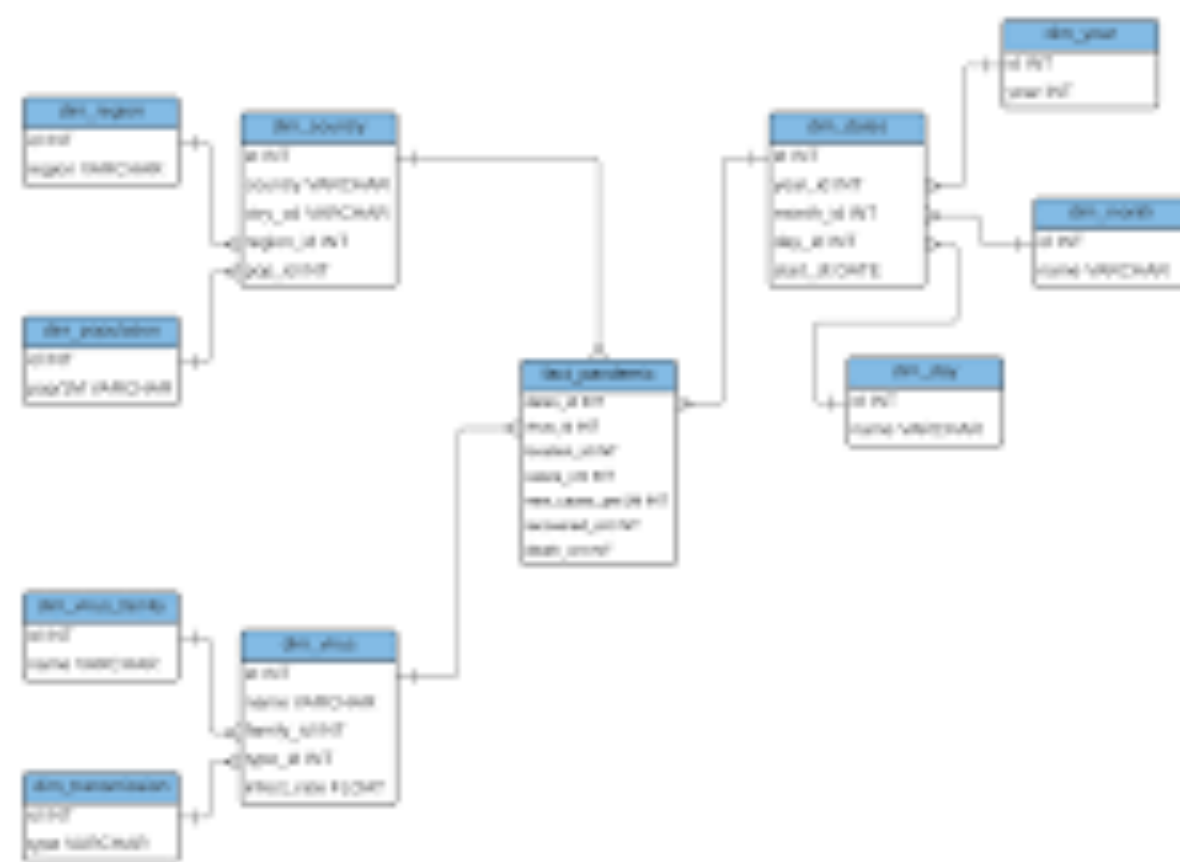


# SQL

SQL is a standard language for storing, manipulating and retrieving data in databases by Data Definition Language (DDL) statements and Data Manipulation Statements (DML).

# DDL modifies the schema of a database table

## DML Manipulate data from the



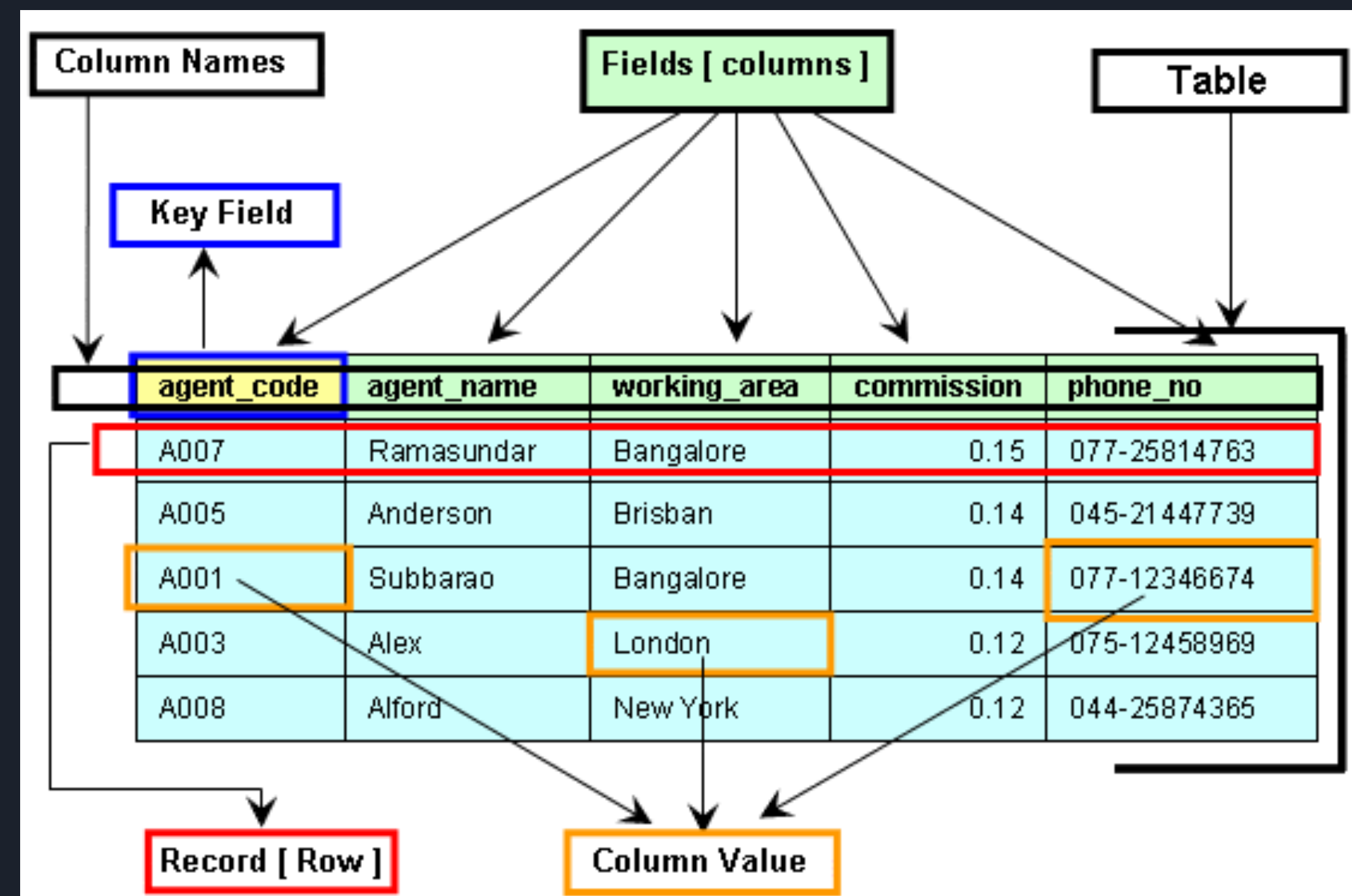
```
SELECT something FROM some table
      WHERE some conditions are satisfied
```

```
UPDATE some table SET something
    WHERE some conditions are satisfied
```

SQL

# SQL Tables

Table is a database object which is comprised of rows and columns. Where data is logically organized in a row-and-column format similar to a spreadsheet. Each row represents a unique record and each column a field in the record.





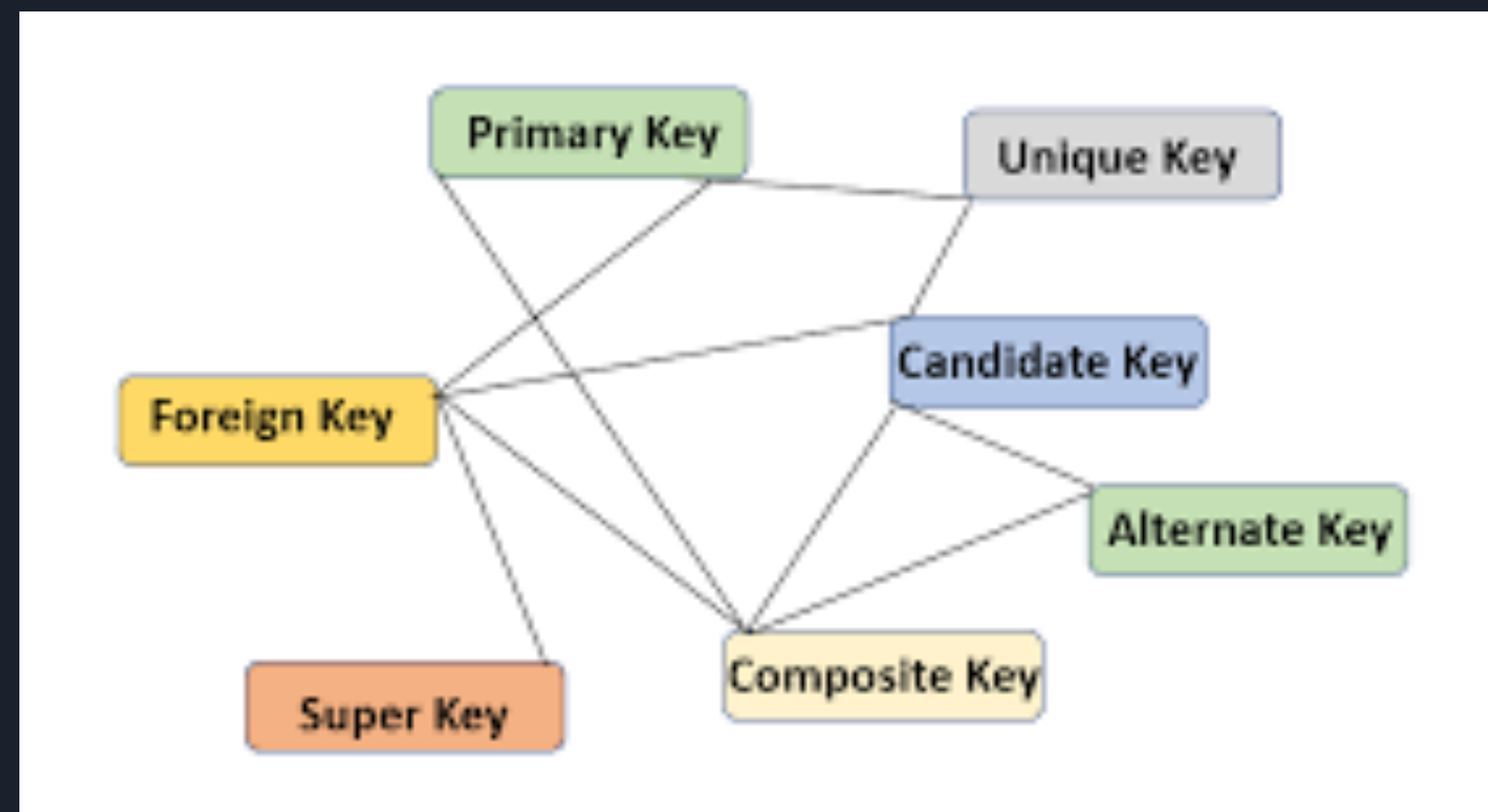


# Elements of a table

- **Fields**
  - The information of a table store in some heads those are fields or columns shown vertically
- **Columns**
  - Each field or column has an individual name. A table cannot contain the same name of two different columns
- **Record (Row)**
  - All the columns in a table make a row> Each row contains all the information of individual topics
- **Column value**
  - The value of each field
- **Key Field**
  - Each table contain a field which can create a link with another one or more table is the key field

# SQL key's

An SQL key is either a single column or group of columns that can uniquely identify rows in a table. They ensure that there are no rows with duplicate information. They also help in establishing a relationship between multiple tables in the database.



# Candidate Key

Candidate key is a single key or a group of multiple keys that uniquely identify **rows** in a table.

Id	Name	Gender	City	Email	Dep_Id
1	Ajay	M	Delhi	ajay@gmail.com	1
2	Vijay	M	Mumbai	vijay@gmail.com	2
3	Radhika	F	Bhopal	radhika@gmail.com	1
4	Shikha	F	Jaipur	shikha@gmail.com	2
5	Hritik	M	Jaipur	hritik@gmail.com	2

5 rows in set (0.00 sec)

# Primary key

Primary key is the Candidate key selected by the database administrator to uniquely identify tuples in a table. There can only be one and non-null attribute(s).

## Alternate or Secondary Key

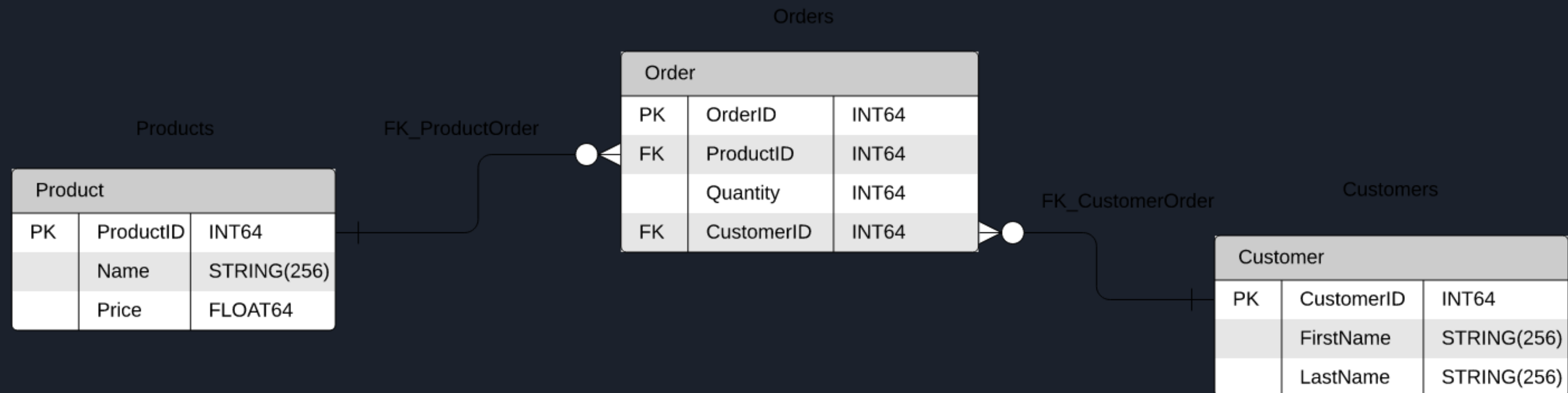
Alternate keys are those candidate keys which are not the Primary key.

Primary Key				Alternate Key	
Id	Name	Gender	City	Email	Dep_Id
1	Ajay	M	Delhi	ajay@gmail.com	1
2	Vijay	M	Mumbai	vijay@gmail.com	2
3	Radhika	F	Bhopal	radhika@gmail.com	1
4	Shikha	F	Jaipur	shikha@gmail.com	2
5	Hritik	M	Jaipur	hritik@gmail.com	2

5 rows in set (0.00 sec)

# Foreign key

Foreign key is an attribute which is a Primary key in a parent table, but is included as an attribute in another host table.





# Super key

Super key is a single key or group of multiple keys that can uniquely identify **tuples** in a table.

ID	Name	Register_no
2026	Andrew	012917323
2025	John	123471904
2024	William	233490183

# Composite Key

A Composite key is a candidate key or Primary key that consist of more than one attribute .  
Cases were no single attribute will have the property to uniquely identify tuples in a table.

```
mysql> select * from product;
```

Transaction_Id	Product_Id	Customer_Id	Product	Quantity
A1001	P1005	C9001	Smartphone	1
A1001	P2010	C9001	Screen guard	1
A1002	P2013	C9003	Smartwatch	1
A1003	P2010	C9010	Screen guard	2

```
4 rows in set (0.00 sec)
```



# ANALOGY

Imagine that a company has a large number of employees, and the company needs to keep track of all their personal and employment information. The company could create a database to store this information, with each employee represented as a record in the database.

Just like how each person has a unique identifier such as a name or social security number, each record in the database would have a unique identifier, such as an employee ID. This ID would be used to uniquely identify each employee record in the database.

Now, just as people have different attributes such as age, gender, and job title, the employee records in the database would have different fields to store different pieces of information, such as name, address, phone number, job title, department, and salary.

SQL would be used to interact with the employee database. For example, an SQL query could be used to select all employees who work in a specific department, or to update the salary of all employees who meet certain criteria.

In this analogy, the employee database is like a huge filing cabinet, with each record representing a person, and SQL is like the language that the HR department uses to manage the information stored in the filing cabinet.





SQL (Structured Query Language) is a programming language that is used to manage and manipulate relational databases. It is used to create, modify, and delete databases, tables, and other database objects, as well as to insert, update, and retrieve data from the database.

SQL is used in a wide variety of applications, from small businesses managing customer data to large corporations managing complex financial systems. It is used by software developers, data analysts, and data scientists to work with databases.

SQL is a standard language, meaning that its syntax and commands are standardized across database platforms. However, there may be some variations in the way different database systems implement certain features or functions.

There are many different types of SQL commands, including SELECT, INSERT, UPDATE, and DELETE, as well as commands for creating and modifying tables, indexes, and views. SQL also includes functions for performing calculations, formatting data, and working with dates and times.

# WHAT ABOUT THE RELATIONAL PART?

The "relational" part of relational databases refers to the way that different tables of data are related to each other. In a relational database, data is organized into one or more tables, with each table containing multiple rows (also known as records) and columns.

For example, if we go back to the employee database analogy, the company might have a separate table to store information about each department, with each row representing a different department and each column containing information about the department, such as its name, location, and manager.

To relate the two tables together, the employee table might have a column called "department ID" that is used to link each employee to their department. This column would contain the unique identifier for each department, which could be used to look up additional information about the department in the department table.

SQL includes commands for working with related data across multiple tables. For example, an SQL JOIN command can be used to combine data from two or more tables based on a common field, such as the department ID in the employee and department tables.

By using tables and establishing relationships between them, relational databases can efficiently store and manage large amounts of data while still allowing for complex queries and data analysis.

1. **CREATE TABLE:** This command is used to create a new table in the database. It specifies the table's name, the names and data types of its columns, and any constraints or rules for the data.

Example:

```
CREATE TABLE employees (  
    id INT PRIMARY KEY,  
    name VARCHAR(50),  
    department_id INT,  
    hire_date DATE,  
    salary DECIMAL(10,2)  
);
```

This command creates a new table called "employees" with columns for the employee ID, name, department ID, hire date, and salary.

2.INSERT INTO: This command is used to insert a new row of data into a table.

Example:

```
INSERT INTO employees (id, name, department_id, hire_date, salary)
VALUES (1, 'John Smith', 3, '2022-01-01', 50000.00);
```



This command inserts a new row of data into the "employees" table, with values for the employee ID, name, department ID, hire date, and salary.

3. SELECT: This command is used to retrieve data from one or more tables. It can be used to filter and sort the data, as well as to perform calculations and groupings.

Example:

```
SELECT name, department_id, salary
FROM employees
WHERE department_id = 3
ORDER BY salary DESC;
```

This command retrieves the name, department ID, and salary for all employees who work in department 3, and sorts them in descending order by salary.

4.UPDATE: This command is used to modify existing data in a table.

Example:

```
UPDATE employees  
SET salary = 55000.00  
WHERE id = 1;
```



This command updates the salary of the employee with ID 1 to 55000.00.

5.DELETE: This command is used to delete rows of data from a table.

Example:

```
DELETE FROM employees  
WHERE department_id = 3;
```

This command deletes all rows from the "employees" table where the department ID is 3.

# SQL VS. EXCEL

Excel and SQL are two different tools used for different purposes.

Excel is a spreadsheet program used for organizing, analyzing, and manipulating data in a tabular format. It provides a graphical user interface that allows users to create spreadsheets with cells, rows, and columns. It can perform mathematical operations, create charts, graphs, and pivot tables. Excel is a standalone application that is installed on a user's computer.

SQL (Structured Query Language), on the other hand, is a programming language designed to manage and manipulate data in relational databases. SQL is used to create, modify, and retrieve data from databases. It is used to create, modify, and delete tables, as well as to insert, update, and delete data. SQL is often used in conjunction with database management systems such as MySQL, Oracle, and Microsoft SQL Server.

One key difference between Excel and SQL is that Excel is better suited for analyzing smaller data sets, while SQL is designed for working with larger data sets stored in a database. Excel is also more suitable for ad-hoc data analysis and quick calculations, while SQL is better suited for complex queries and data manipulation. Additionally, SQL offers more advanced functionalities such as data normalization, indexing, and data security that Excel does not provide.

# ANALOGY

One possible analogy to illustrate the difference between Excel and SQL is that Excel is like a calculator or a small tool for performing quick calculations, while SQL is like a toolbox filled with a variety of tools that are used for complex construction and engineering tasks.

Just as a calculator can perform simple mathematical operations and provide quick answers, Excel can quickly organize, analyze, and manipulate small sets of data. However, just as a calculator has limited capabilities and cannot be used for more complex tasks, Excel also has its limitations and may not be the best tool for handling large or complex data sets.

On the other hand, SQL is like a toolbox filled with many tools designed for different tasks, such as hammers, saws, drills, and screwdrivers. SQL provides a set of tools for managing and manipulating large databases, just as a toolbox provides a set of tools for managing and manipulating different materials and construction projects. While SQL may require more knowledge and training to use effectively, it offers a much wider range of capabilities and can be used for complex data analysis and management.

# WEB CRAWLERS

- EXAMPLE IN PYTHON USING BS