INFOLINK-EXP

WHAT IS PRICESPIDER?

Introducing client’s products

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.

SQL

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.

ANALOGY

magine 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.

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