

Formatting for better analysis

Combine multiple datasets

- ✓

Reading: Import and combine data in spreadsheets and databases

20 min
- 📖

Reading: Step-by-Step: Merge text strings to gain insights

20 min
- ▶

Video: Merge text strings to gain insights

4 min
- 📖

Practice Quiz: Hands-On Activity: Combine multiple pieces of data

1h
- 📖

Reading: Step-by-Step: Strings in spreadsheets

20 min
- ▶

Video: Strings in spreadsheets

3 min
- 📖

Reading: Manipulate strings with SQL

10 min
- 🔗

Ungraded Plugin: SQL query syntax

10 min
- 📖

Practice Quiz: Test your knowledge on combining multiple datasets

8 min

Get support during analysis

Module 2 challenge

Import and combine data in spreadsheets and databases

In earlier lessons, you discovered how to use the **IMPORTRANGE** and **CONCATENATE** functions in spreadsheets. In this reading, you will have the opportunity to extend your knowledge about these concepts to SQL queries.

Import data

As a data analyst, there are many occasions where you will need to import data from one file or location to another. Both spreadsheets and SQL include functionality that enables you to import data.

Import data in spreadsheets

As you learned earlier, in spreadsheets you use the **IMPORTRANGE** function to import a range of cells from another spreadsheet into your current spreadsheet. The syntax is: **=IMPORTRANGE (spreadsheet_url , range_string) .**

In this formula, **spreadsheet_url** is the URL of the spreadsheet from which you want to import data. The specific cells you want to import, such as A2:B6, are specified by **range_string**. If the spreadsheet has multiple tabs, you also need to specify the name of the tab as part of the range.

An example of this is a company that needs to track who made retirement contributions so that it can make sure the company match is correctly distributed. The analysts would use **IMPORTRANGE** to pull all retirement contribution information into a spreadsheet that contains all of the employees year-end salaries and bonuses. This enables them to determine which employees made contributions and are eligible for matching funds.

Import data in SQL

In contrast to spreadsheets, SQL does not include a function for importing data. Instead, a method you can use to import data from one table to another is to use the **INSERT INTO** command together with a **SELECT** statement. The syntax is:

```
3 FROM [source_table_name]
4 WHERE [condition]
```

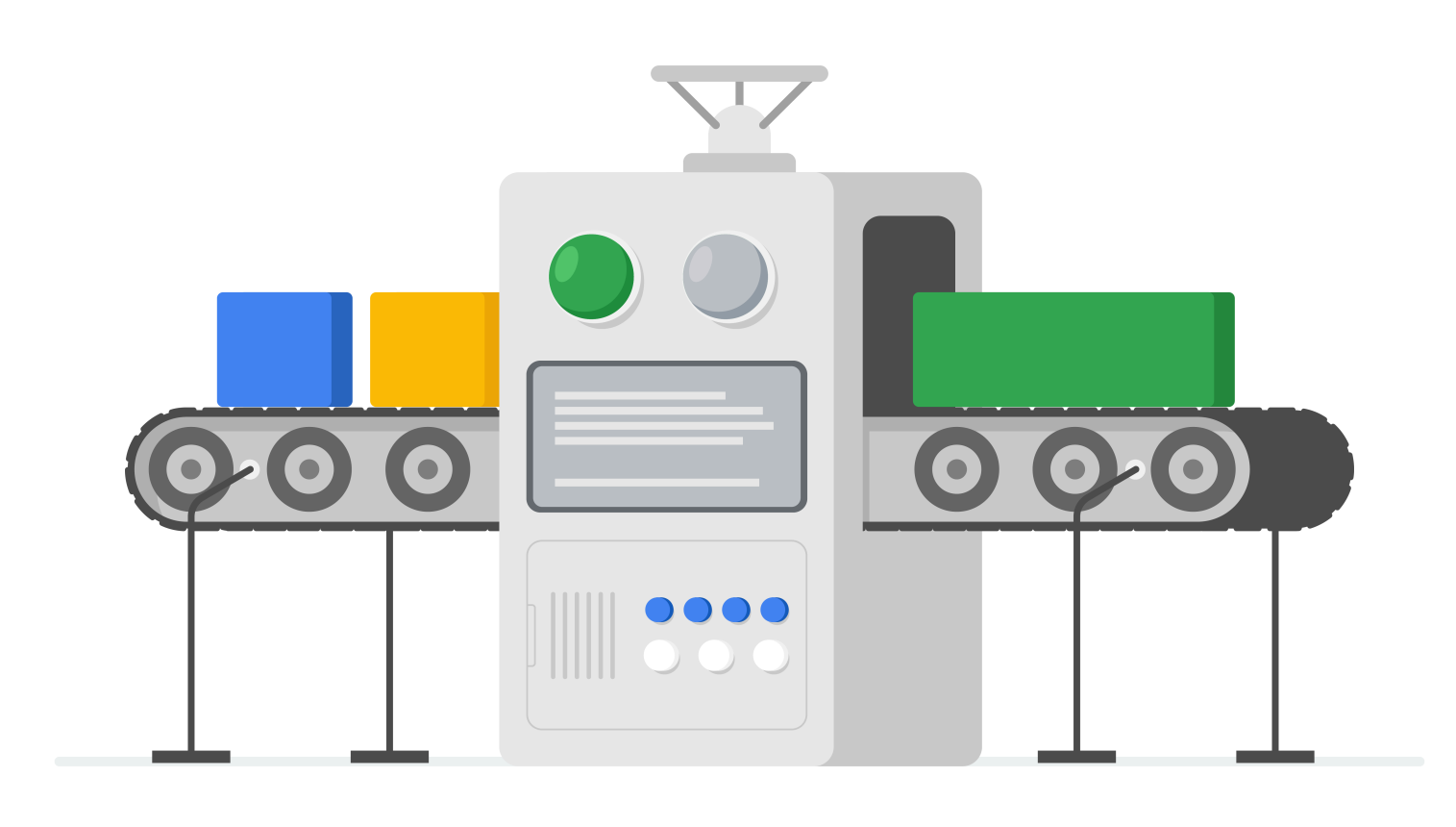
In this syntax, the SQL query inserts rows from a source table into a destination table based on the **WHERE** clause.

For example, imagine you work for a retail company that stores its sales and customer information in a SQL database. The marketing director asks you to provide them with a table containing the names and addresses of customers who have not made a purchase this year and who live in specific postal codes. One way you could gather this information is to use the **INSERT INTO** along with the **SELECT** and **WHERE** commands, as follows:

```
1 INSERT INTO customer_promotion
2 SELECT *
3 FROM customers
4 WHERE total_sales = 0 AND postal_code = '12345'
```

Combine data

Another tool in your data analyst toolkit is your ability to join together two or more text strings that are stored in separate columns or fields. For example, you might want to combine a customer's first and last name to create mailing labels for a marketing promotion. In both spreadsheets and SQL, joining together text strings is referred to as *concatenation*.



Combine data in spreadsheets

In spreadsheets, you use the **CONCATENATE** function to join together two or more text strings, such as combining street addresses and primary contacts in a business' vendor database.

The basic syntax is **=CONCATENATE (item 1, item 2)**. You can add multiple items by separating them with commas. Where appropriate, such as when you're combining a customer's first and last name, you should add a space between the items you're combining by typing quotation marks space quotation marks [" "] between the items. Separate this information by a comma as well. This would change the formula to: **=CONCATENATE (item 1, " ", item 2)**.

Combine data in SQL

In SQL, use the **CONCAT** function to join strings together to create new text strings. You might combine data simply to improve the readability of reports (such as combining a customer's first and last name when generating a customer list). Or, you might combine data to generate a unique identifier for the rows in a table. Here is the basic syntax:

```
2 FROM [table_name]
```

Notice that this syntax includes " " so that there is a space between the combined fields. With this syntax, SQL combines field1 and field2 with a space between them.

By default, SQL includes the field names as headers when you run a query. However, if you use the **CONCAT** function, SQL doesn't know what to use as a header. For this reason, you should include an alias for the combined fields to help with readability. You give the combined fields an alias by using **AS**:

```
1 SELECT CONCAT(field1, " ", field2) AS alias
2 FROM [table_name]
```

For example, if you plan to use **CONCAT** to combine the first and last names of your company's customers into a single expression, you could use this query:

```
1 SELECT CONCAT(first_name, " ", last_name) AS Customer_Name
2 FROM [table_name]
```