**Konsep inti VLOOKUP**

Fungsi dapat digunakan untuk menemukan informasi dengan cepat dan melakukan penghitungan menggunakan nilai tertentu. Dalam bacaan ini, Anda akan mempelajari pentingnya salah satu fungsi tersebut, **VLOOKUP** , atau Pencarian Vertikal, yang mencari nilai tertentu di kolom spreadsheet dan mengembalikan potongan informasi terkait dari baris tempat nilai yang dicari ditemukan.



**Kapan Anda perlu menggunakan VLOOKUP?**

Dua alasan umum untuk menggunakan VLOOKUP adalah:

* Mengisi data dalam spreadsheet
* Menggabungkan data dari satu spreadsheet dengan data di spreadsheet lainnya

**Sintaks VLOOKUP**

Fungsi VLOOKUP tersedia di Microsoft Excel dan Google Sheets. Anda akan diperkenalkan dengan sintaks umum di Google Sheets. (Anda dapat merujuk ke sumber daya di akhir bacaan ini untuk informasi lebih lanjut tentang VLOOKUP di Microsoft Excel.)



Berikut sintaksnya.



**kunci\_penelusuran**

* Nilai yang dicari.
* Misalnya, 42, "Kucing", atau I24.

**jangkauan**

* Kisaran yang perlu dipertimbangkan untuk pencarian.
* Kolom pertama dalam rentang dicari untuk menemukan data yang cocok dengan nilai yang ditentukan oleh kunci\_penelusuran.

**indeks**

* Indeks kolom dari nilai yang akan dikembalikan, dengan kolom pertama dalam rentang diberi nomor 1.
* Jika indeks tidak berada di antara 1 dan jumlah kolom dalam rentang, #VALUE! dikembalikan.

**adalah\_diurutkan**

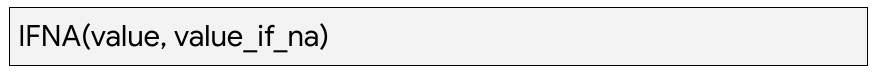
* Menunjukkan apakah kolom yang akan dicari (kolom pertama dari rentang yang ditentukan) sudah diurutkan. BENAR secara default.
* Disarankan untuk menyetel is\_sorted ke FALSE. Jika disetel ke FALSE, hasil yang sama persis akan dikembalikan. Jika ada beberapa nilai yang cocok, konten sel yang sesuai dengan nilai pertama yang ditemukan akan dikembalikan, dan #N/A dikembalikan jika tidak ada nilai tersebut yang ditemukan.
* Jika is\_sorted adalah TRUE atau dihilangkan, kecocokan terdekat (kurang dari atau sama dengan kunci pencarian) akan dikembalikan. Jika semua nilai di kolom pencarian lebih besar dari kunci pencarian, #N/A dikembalikan.

**Bagaimana jika Anda mendapatkan #N/A?**

Seperti yang baru saja Anda baca, #N/A menunjukkan bahwa nilai yang cocok tidak dapat dikembalikan sebagai hasil VLOOKUP. Kesalahan ini tidak berarti ada yang salah dengan data, namun orang mungkin bertanya-tanya jika mereka melihat kesalahan dalam laporan. Anda dapat menggunakan fungsi **IFNA** untuk mengganti kesalahan #N/A dengan sesuatu yang lebih deskriptif, seperti “Tidak ada.”



Berikut sintaksnya.



**nilai**

* Ini adalah nilai yang diperlukan.
* Fungsi ini memeriksa apakah nilai sel cocok dengan nilainya; seperti #T/A.

**nilai\_jika\_na**

* Ini adalah nilai yang diperlukan.
* Fungsi mengembalikan nilai ini jika nilai sel cocok dengan nilai dalam argumen pertama; ia mengembalikan nilai ini ketika nilai sel #N/A.

**Pengingat VLOOKUP yang bermanfaat**

* TRUE berarti pencocokan perkiraan, FALSE berarti pencocokan tepat pada tombol pencarian. Jika data yang digunakan untuk kunci pencarian sudah diurutkan, TRUE dapat digunakan.
* Anda ingin kolom yang cocok dengan kunci pencarian dalam rumus VLOOKUP berada di sisi kiri data. VLOOKUP hanya melihat data di sebelah kanan setelah ditemukan kecocokan. Dengan kata lain, indeks untuk VLOOKUP menunjukkan kolom di sebelah kanan saja. Ini mungkin mengharuskan Anda memindahkan kolom sebelum menggunakan VLOOKUP.
* Setelah Anda mengisi data dengan rumus VLOOKUP, Anda dapat menyalin dan menempelkan data sebagai nilai hanya untuk menghapus rumus sehingga Anda dapat memanipulasi data lagi.

**Sumber daya VLOOKUP untuk Microsoft Excel**

VLOOKUP mungkin sedikit berbeda di Microsoft Excel, namun konsep keseluruhannya masih dapat diterapkan secara umum. Lihat sumber daya berikut jika Anda bekerja dengan Excel.

* [Cara menggunakan VLOOKUP di Excel](https://support.microsoft.com/en-us/office/vlookup-function-0bbc8083-26fe-4963-8ab8-93a18ad188a1): Tutorial ini menyertakan video untuk membantu Anda mendapatkan pemahaman umum tentang cara kerja fungsi VLOOKUP di Excel, serta contoh praktis untuk dilihat.
* [Tutorial VLOOKUP di Excel](https://www.youtube.com/watch?v=d3BYVQ6xIE4): Ikuti pelajaran video ini dan pelajari cara menulis rumus VLOOKUP di Excel dan kuasai tip dan trik berguna yang menghemat waktu.
* [23 hal yang perlu Anda ketahui tentang VLOOKUP di Excel](https://exceljet.net/things-you-should-know-about-vlookup): Jelajahi daftar 23 fakta VLOOKUP serta tantangan yang mungkin Anda hadapi, dan mulailah belajar cara menguasainya.
* [Cara menggunakan fungsi VLOOKUP Excel](https://edu.gcfglobal.org/en/excel-tips/how-to-use-excels-vlookup-function/1/): Artikel ini membagikan contoh spesifik tentang cara menerapkan VLOOKUP dalam penelusuran Anda.
* [VLOOKUP di Excel vs Google Spreadsheet](https://infoinspired.com/sheets-vs-excel-formula/vlookup-formula-in-excel-and-google-sheets/): Panduan ini menawarkan perbandingan VLOOKUP Excel dan Google Sheets.

**Identitas rahasia: Pentingnya alias**

Dalam bacaan ini, Anda akan belajar tentang penggunaan aliasing untuk menyederhanakan query SQL Anda. **Alias** ​​​​digunakan dalam kueri SQL untuk membuat nama sementara untuk kolom atau tabel. Alias ​​membuat referensi tabel dan kolom dalam kueri SQL Anda menjadi lebih sederhana bila Anda memiliki nama tabel atau kolom yang terlalu panjang atau rumit untuk digunakan dalam kueri. Bayangkan nama tabel seperti special\_projects\_customer\_negotiation\_mileages. Akan sulit untuk mengetik ulang setiap kali Anda menggunakan tabel itu. Dengan alias, Anda dapat membuat nama panggilan bermakna yang dapat digunakan untuk analisis Anda. Dalam hal ini “special\_projects\_customer\_negotiation\_mileages” dapat juga disebut dengan “mileage”. Daripada harus menuliskan nama tabel yang panjang, Anda bisa menggunakan nama panggilan yang bermakna sesuai keinginan Anda.

**Sintaks dasar untuk aliasing**

**Aliasing** adalah proses penggunaan alias. Dalam kueri SQL, alias diimplementasikan dengan menggunakan perintah AS. Sintaks dasar perintah AS dapat dilihat pada query berikut untuk membuat aliasing suatu tabel:



Perhatikan bahwa AS diawali dengan nama tabel dan diikuti dengan nama panggilan baru. Ini adalah pendekatan yang mirip dengan membuat alias pada kolom:



Dalam kedua kasus tersebut, Anda sekarang memiliki nama baru yang bisa Anda gunakan untuk merujuk ke kolom atau tabel yang diberi alias.

**Sintaks alternatif untuk alias**

Jika penggunaan AS menghasilkan kesalahan saat menjalankan kueri karena database SQL yang Anda gunakan tidak mendukungnya, Anda bisa mengabaikannya. Pada contoh sebelumnya, sintaks alternatif untuk membuat alias pada tabel atau kolom adalah:

* DARI nama\_tabel alias\_nama
* PILIH nama\_kolom alias\_nama

Kesimpulan utamanya adalah kueri dapat dijalankan dengan atau tanpa penggunaan AS untuk aliasing, namun penggunaan AS mempunyai keuntungan membuat kueri lebih mudah dibaca. Ini membantu membuat alias terlihat lebih jelas.

**Aliasing sedang beraksi**

Mari kita lihat contoh query SQL yang menggunakan aliasing. Misalkan Anda bekerja dengan dua tabel: salah satunya berisi data karyawan dan tabel lainnya berisi data departemen. Pernyataan FROM untuk alias tabel-tabel tersebut dapat berupa:

DARI hari\_kerja.karyawan SEBAGAI karyawan

Alias ​​ini tetap memberi tahu Anda secara pasti apa yang ada di tabel ini, namun kini Anda tidak perlu memasukkan nama tabel panjang tersebut secara manual. Alias ​​bisa sangat membantu untuk pertanyaan yang panjang dan rumit. Lebih mudah untuk membaca dan menulis pertanyaan Anda ketika Anda memiliki alias yang memberi tahu Anda apa yang termasuk dalam tabel Anda.

**Untuk informasi lebih lanjut**

Jika Anda tertarik mempelajari lebih lanjut tentang aliasing, berikut beberapa sumber untuk membantu Anda memulai:

* [**Alias ​​​​SQL**](https://www.w3schools.com/sql/sql_alias.asp)**:** Tutorial tentang aliasing ini adalah sumber yang sangat berguna ketika Anda mulai berlatih menulis kueri dan membuat tabel aliasing sendiri. Ini juga menunjukkan cara kerja aliasing dengan tabel nyata.
* [**Alias ​​SQL**](https://www.sqltutorial.org/sql-alias/)**:** Pengenalan mendetail tentang aliasing ini mencakup banyak contoh. Ini adalah sumber referensi bagus lainnya jika Anda memerlukan lebih banyak contoh.
* [**Menggunakan Aliasing Kolom**](https://documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.5&docsetId=sqlproc&docsetTarget=p0aymxwsvbt5wcn1lncugwjtf758.htm&locale=en)**:** Ini adalah panduan yang berfokus pada aliasing kolom secara khusus. Secara umum, Anda akan membuat alias pada seluruh tabel, namun jika Anda perlu membuat alias hanya pada satu kolom, ini adalah sumber yang bagus untuk di-bookmark.

**Menggunakan GABUNG secara efektif**

Dalam bacaan ini, Anda akan meninjau bagaimana GABUNG digunakan dan diperkenalkan dengan beberapa sumber yang dapat Anda gunakan untuk mempelajari lebih lanjut tentang GABUNG. JOIN menggabungkan tabel dengan menggunakan kunci utama atau kunci asing untuk menyelaraskan informasi yang berasal dari kedua tabel dalam proses kombinasi. GABUNG menggunakan kunci ini untuk mengidentifikasi hubungan dan nilai terkait di seluruh tabel.

Jika Anda memerlukan penyegaran tentang kunci utama dan asing, lihat[Glosarium](https://www.coursera.org/learn/analyze-data/supplement/0p8b6/glossary-terms-and-definitions)untuk kursus ini, atau kembali ke[Basis data dalam analisis data](https://www.coursera.org/learn/data-preparation/supplement/uXqEX/databases-in-data-analytics).

**Sintaks GABUNG umum**

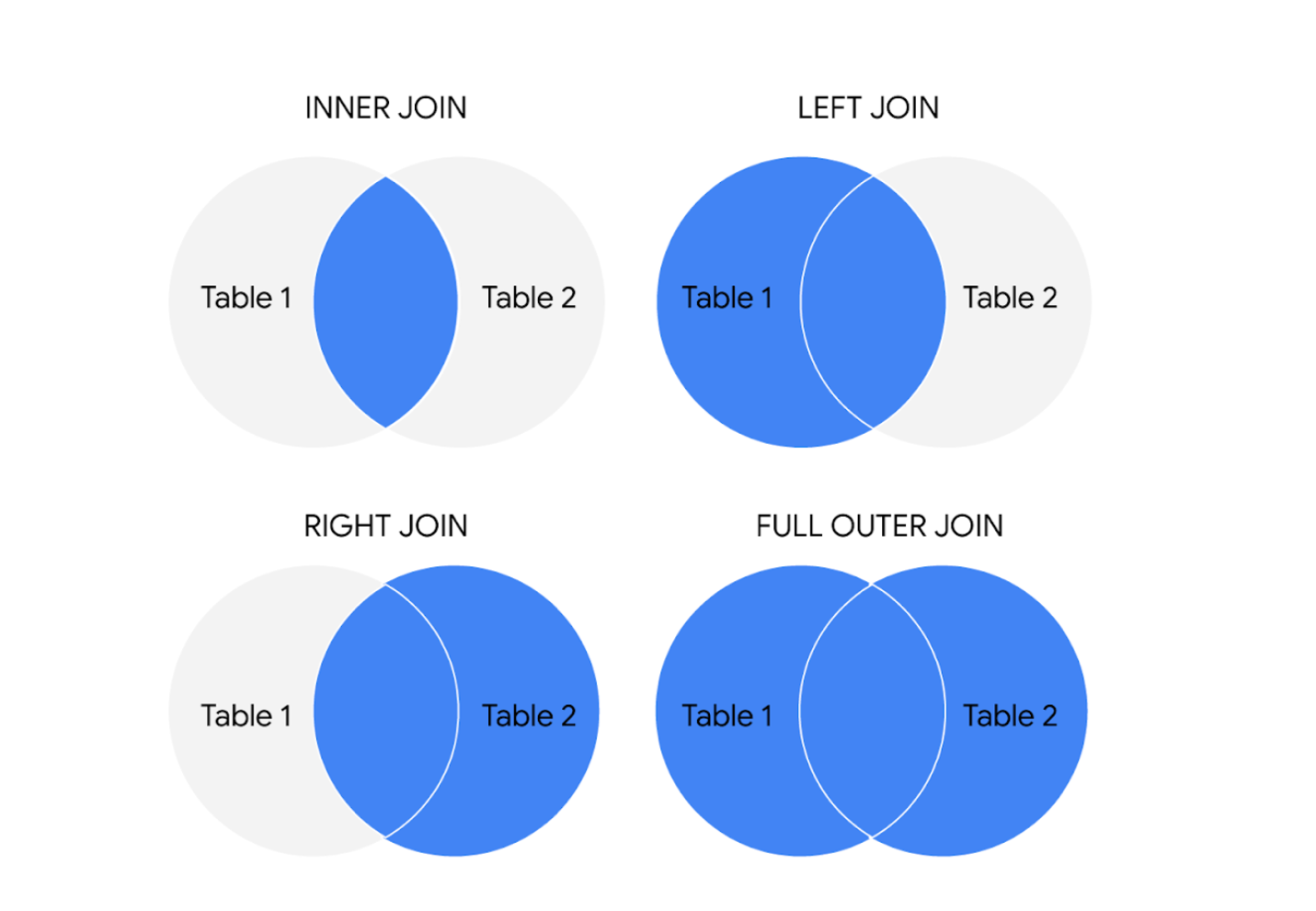
A screen shot of a computer

Description automatically generated

Seperti yang Anda lihat dari sintaksnya, pernyataan JOIN adalah bagian dari klausa FROM kueri. **GABUNG** di SQL menunjukkan bahwa Anda akan menggabungkan data dari dua tabel. **ON** di SQL mengidentifikasi bagaimana tabel harus dicocokkan agar informasi yang benar dapat digabungkan dari keduanya.

**Jenis GABUNG**

Ada empat cara umum untuk melakukan JOIN dalam query SQL: INNER, LEFT, RIGHT, dan FULL OUTER.

Lingkaran mewakili tabel kiri dan kanan, dan tempat penggabungannya disorot dengan warna biru

Inilah yang dilakukan oleh kueri GABUNG yang berbeda ini.

**GABUNG DALAM**

INNER bersifat *opsional* dalam kueri SQL ini karena ini adalah operasi default dan juga operasi GABUNG yang paling umum digunakan. Anda mungkin melihat ini sebagai GABUNG saja. INNER JOIN mengembalikan catatan jika data berada di kedua tabel. Misalnya, jika Anda menggunakan INNER JOIN untuk tabel 'pelanggan' dan 'pesanan' dan mencocokkan data menggunakan kunci id\_pelanggan, Anda akan menggabungkan data untuk setiap id\_pelanggan yang ada di kedua tabel. Jika id\_pelanggan ada di tabel pelanggan namun tidak ada di tabel pesanan, data untuk id\_pelanggan tersebut tidak digabungkan atau dikembalikan oleh kueri.

A grey screen with white text

Description automatically generated

Hasil dari kueri mungkin terlihat seperti berikut, dengan nama\_pelanggan berasal dari tabel pelanggan dan id\_produk serta tanggal\_pengiriman berasal dari tabel pesanan:

| **Nama Pelanggan** | **ID Produk** | **tanggal pengiriman** |
| --- | --- | --- |
| Es Krim Martin | 043998 | 23-02-2021 |
| Camilan Tepi Pantai | 872012 | 25-02-2021 |
| Rasa Alami Mona | 724956 | 28-02-2021 |
| ... dll. | ... dll. | ... dll. |

Data dari kedua tabel digabungkan dengan mencocokkan id\_pelanggan yang umum pada kedua tabel. Perhatikan bahwa customer\_id tidak muncul di hasil kueri. Ini hanya digunakan untuk membangun hubungan antara data dalam dua tabel sehingga data dapat digabungkan dan dikembalikan.

**KIRI GABUNG**

Anda mungkin melihat ini sebagai LEFT OUTER JOIN, namun sebagian besar pengguna lebih memilih LEFT JOIN. Keduanya adalah sintaks yang benar. LEFT JOIN mengembalikan semua catatan dari tabel kiri dan hanya catatan yang cocok dari tabel kanan. Gunakan LEFT JOIN kapan pun Anda membutuhkan data dari seluruh tabel pertama dan nilai dari tabel kedua, jika ada. Misalnya, dalam kueri di bawah, LEFT JOIN akan mengembalikan nama\_pelanggan dengan perwakilan\_penjualan yang sesuai, jika tersedia. Jika ada pelanggan yang tidak berinteraksi dengan perwakilan penjualan, pelanggan tersebut akan tetap muncul di hasil kueri namun dengan nilai NULL untuk sales\_rep.

A screen shot of a computer

Description automatically generated

Hasil dari kueri mungkin terlihat seperti berikut ini, dimana nama\_pelanggan berasal dari tabel pelanggan dan perwakilan\_penjualan berasal dari tabel penjualan. Sekali lagi, data dari kedua tabel digabungkan dengan mencocokkan id\_pelanggan yang umum untuk kedua tabel meskipun id\_pelanggan tidak dikembalikan dalam hasil kueri.

| **Nama Pelanggan** | **perwakilan\_penjualan** |
| --- | --- |
| Es Krim Martin | Luis Reyes |
| Camilan Tepi Pantai | BATAL |
| Rasa Alami Mona | Aula Geri |
| ...dll. | ...dll. |

**BENAR BERGABUNG**

Anda mungkin melihatnya sebagai RIGHT OUTER JOIN atau RIGHT JOIN. GABUNG KANAN mengembalikan semua catatan dari tabel kanan dan catatan terkait dari tabel kiri. Secara praktis, RIGHT JOIN jarang digunakan. Kebanyakan orang hanya berpindah tabel dan tetap menggunakan LEFT JOIN. Namun jika menggunakan contoh sebelumnya untuk LEFT JOIN, query yang menggunakan RIGHT JOIN akan terlihat seperti berikut:

A screen shot of a computer

Description automatically generated

Hasil querynya sama dengan contoh LEFT JOIN sebelumnya.

| **Nama Pelanggan** | **perwakilan\_penjualan** |
| --- | --- |
| Es Krim Martin | Luis Reyes |
| Camilan Tepi Pantai | BATAL |
| Rasa Alami Mona | Aula Geri |
| ...dll. | ...dll. |

**GABUNG LUAR LENGKAP**

Terkadang Anda mungkin melihat ini sebagai FULL JOIN. FULL OUTER JOIN mengembalikan semua catatan dari tabel yang ditentukan. Anda dapat menggabungkan tabel dengan cara ini, namun ingat bahwa hal ini berpotensi menghasilkan penarikan data yang besar. FULL OUTER JOIN mengembalikan semua catatan dari *kedua* tabel meskipun data tidak diisi di salah satu tabel. Misalnya, dalam kueri di bawah ini, Anda akan mendapatkan semua pelanggan dan tanggal pengiriman produk mereka. Karena Anda menggunakan FULL OUTER JOIN, Anda mungkin mendapatkan pelanggan dikembalikan tanpa tanggal pengiriman yang sesuai atau tanggal pengiriman tanpa pelanggan yang sesuai. Nilai NULL dikembalikan jika data terkait tidak ada di tabel mana pun.

A computer screen shot of a computer

Description automatically generated

Hasil dari kueri mungkin terlihat seperti berikut.

| **Nama Pelanggan** | **tanggal pengiriman** |
| --- | --- |
| Es Krim Martin | 23-02-2021 |
| Camilan Tepi Pantai | 25-02-2021 |
| BATAL | 25-02-2021 |
| Berita Harian | BATAL |
| Es Krim Gunung | BATAL |
| Rasa Alami Mona | 28-02-2021 |
| ...dll. | ...dll. |

**Untuk informasi lebih lanjut**

GABUNG akan berguna untuk bekerja dengan database relasional dan SQL—dan Anda akan memiliki banyak kesempatan untuk mempraktikkannya sendiri. Berikut adalah beberapa sumber lain yang dapat memberi Anda informasi lebih lanjut tentang GABUNG dan cara menggunakannya:

* [**SQL BERGABUNG**](https://www.w3schools.com/sql/sql_join.asp)**:** Ini adalah penjelasan dasar yang bagus tentang GABUNG beserta contohnya. Jika Anda memerlukan pengingat singkat tentang apa yang dilakukan GABUNG yang berbeda, ini adalah sumber yang bagus untuk ditandai dan dilihat lagi nanti.
* [**Database JOINs - Pengenalan Jenis dan Konsep JOIN**](https://www.essentialsql.com/introduction-database-joins/)**:** Ini adalah pengenalan menyeluruh tentang GABUNG. Artikel ini tidak hanya menjelaskan apa itu GABUNG dan cara menggunakannya, tetapi juga menjelaskan berbagai skenario secara lebih rinci tentang kapan dan mengapa Anda menggunakan GABUNG yang berbeda. Ini adalah sumber yang bagus jika Anda tertarik untuk mempelajari lebih lanjut tentang logika di balik BERGABUNG.
* [**Tipe SQL JOIN Dijelaskan dalam Visual**](https://dataschool.com/how-to-teach-people-sql/sql-join-types-explained-visually/)**:** Sumber daya ini memiliki representasi visual dari GABUNG yang berbeda. Ini adalah cara yang sangat berguna untuk memikirkan GABUNG jika Anda seorang pembelajar visual, dan ini bisa menjadi cara yang sangat berguna untuk mengingat GABUNG yang berbeda.
* [**SQL JOIN: Menyatukan Data Satu Gabungan Sekaligus**](https://towardsdatascience.com/sql-join-8212e3eb9fde)**:** Sumber daya ini tidak hanya memiliki penjelasan mendetail tentang GABUNG beserta contohnya, tetapi juga menyediakan contoh data yang dapat Anda gunakan untuk mengikuti panduan langkah demi langkahnya. Ini adalah cara yang berguna untuk melatih GABUNG dengan beberapa data nyata.
* [**SQL BERGABUNG:**](https://www.dofactory.com/sql/join)Ini adalah sumber lain yang memberikan penjelasan jelas tentang GABUNG dan menggunakan contoh untuk menunjukkan cara kerjanya. Contohnya juga menggabungkan GABUNG dengan aliasing. Ini adalah kesempatan bagus untuk melihat bagaimana GABUNG dapat dikombinasikan dengan konsep SQL lain yang telah Anda pelajari dalam kursus ini.

**Optional: Upload the warehouse dataset to BigQuery**

The next video demonstrates how to use COUNT and COUNT DISTINCT in SQL to count and return the number of certain values in a dataset.

If you would like to follow along with the instructor, you will need to log in to your BigQuery account and upload the warehouse data provided as two CSV files. If you have hopped around courses, [Using BigQuery](https://www.coursera.org/learn/data-preparation/supplement/DYOQK/using-bigquery) in the **Prepare Data for Exploration** course covers how to set up a BigQuery account.

**Prepare for the next video**

* First, download the two CSV files from the attachments below:

[Warehouse Orders - Warehouse](https://d3c33hcgiwev3.cloudfront.net/imJai0YnRlyiWotGJ4Zc4g_681d60c67a5c443f90940a3e3b548cf1_Warehouse-Orders---Warehouse.csv?Expires=1694563200&Signature=FvO9seL29z9XaiZrGeJb5~s4LWCNHZ-T8QzjUXLET4LPuRSCdHwToL0I0vmbPOM8iQodxNDud69TRhiIfD0soHHpGalf5uoXnJKRwC52qQQH3Lwrkn-4dACtcs6eGDKpS6-kJfOCuKSYGq1QPh7MLJ7p~OYh4GV6V3pFOG2NyEE_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A" \t "_blank)

[CSV File](https://d3c33hcgiwev3.cloudfront.net/imJai0YnRlyiWotGJ4Zc4g_681d60c67a5c443f90940a3e3b548cf1_Warehouse-Orders---Warehouse.csv?Expires=1694563200&Signature=FvO9seL29z9XaiZrGeJb5~s4LWCNHZ-T8QzjUXLET4LPuRSCdHwToL0I0vmbPOM8iQodxNDud69TRhiIfD0soHHpGalf5uoXnJKRwC52qQQH3Lwrkn-4dACtcs6eGDKpS6-kJfOCuKSYGq1QPh7MLJ7p~OYh4GV6V3pFOG2NyEE_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A" \t "_blank)

[Warehouse Orders - Orders](https://d3c33hcgiwev3.cloudfront.net/7wHK8ZKvRoGByvGSr1aBNw_5433b9c9b98d41b09152b3331141def1_Warehouse-Orders---Orders.csv?Expires=1694563200&Signature=FktkzMApuQOG-HY7CfmWPlHU0V7bYXSahlpuTxQgcRB6btCtpopO9MEn~OSgDRo~ako4hkq7-JySZlSg70VjGIVj8duD8Zzzy-DUdZKlfN2wpbd34vEYm325Ex79beZVSmFY7S3xdZRtFPcL9c3UA~47g9MJaG6qVNeCVmIHH~E_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A" \t "_blank)

[CSV File](https://d3c33hcgiwev3.cloudfront.net/7wHK8ZKvRoGByvGSr1aBNw_5433b9c9b98d41b09152b3331141def1_Warehouse-Orders---Orders.csv?Expires=1694563200&Signature=FktkzMApuQOG-HY7CfmWPlHU0V7bYXSahlpuTxQgcRB6btCtpopO9MEn~OSgDRo~ako4hkq7-JySZlSg70VjGIVj8duD8Zzzy-DUdZKlfN2wpbd34vEYm325Ex79beZVSmFY7S3xdZRtFPcL9c3UA~47g9MJaG6qVNeCVmIHH~E_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A" \t "_blank)

* Next, complete the following steps in your BigQuery console to upload the Warehouse Orders dataset with the two Warehouse and Orders tables.

**Step 1:** Open your BigQuery console and click on the project you want to upload the data to.

**Step 2:** In the Explorer on the left, click the Actions icon (three vertical dots) next to your project name and select **Create dataset**.

A screenshot of a search engine

Description automatically generated

**Step 3:** In the upcoming video, the name "warehouse\_orders" will be used for the dataset. If you plan to follow along with the video, enter **warehouse\_orders** for the Dataset ID in the Create dataset window. Make sure the *Location type* is set to **Multi-region (US)** and all the default *Advanced options* remain set to the **Google-managed encryption key** option.

A screenshot of a computer

Description automatically generated

**Step 4:** Click **CREATE DATASET** (blue button) to add the dataset to your project.

**Step 5:** In the Explorer on the left, click the **warehouse\_orders** dataset you just created. You will then see the dataset info window appear in the main editor window.

A screenshot of a computer

Description automatically generated

**Step 6:** Navigate to the far right side of the screen and click the blue **+ CREATE TABLE** to open the *Create table* window. Use the visual settings in the next image to complete the steps below.

A screenshot of a computer

Description automatically generated

**Step 7:** Under Source, for the *Create table from* selection, choose where the data will be coming from.

* Select **Upload**.
* Click **Browse** to select the Warehouse Orders - Orders CSV file you downloaded.
* Choose **CSV** from the file format drop-down. (file name may automatically be detected)

**Step 8:** For Table name, enter **orders** if you plan to follow along with the video.

**Step 9:** For Schema, click the Auto detect check box.

**Step 10:** At the bottom of the page click the **CREATE TABLE** blue button. You will now see the **orders** table under your **warehouse\_orders** dataset in your project.

A screenshot of a search engine

Description automatically generated

**Step 11:** Click the **warehouse\_orders** dataset again.

**Step 12:** Navigate to the far right side of the screen and click the **+ CREATE TABLE** button to open the *Create table* window again. Use the visual settings in the next image to complete the steps below.

A screenshot of a computer

Description automatically generated

**Step 13:** Under Source, for the Create table from selection, choose where the data will be coming from.

* Select **Upload**.
* Click **Browse** to select the Warehouse Orders - Warehouse CSV file you downloaded.
* Choose **CSV** from the file format drop-down.

**Step 14:** For Table name, enter **warehouse** if you plan to follow along with the video.

**Step 15:** For Schema, click the Auto detect check box.

**Step 16:** At the bottom of the page click the **CREATE TABLE** blue button. You will now see the **warehouse** table under your **warehouse\_orders** dataset in your project.

A screenshot of a search engine

Description automatically generated

**Step 17:** Click the **orders** table and click the **Preview** tab to verify that you have the first 50 rows of data like the image down below. You may have to scroll down the page to view all 50 rows (there are 9999 pages total).

A screenshot of a table

Description automatically generated

**Step 18:** Click the **warehouse** table and click the **Preview** tab to verify that you have the 10 rows of data like the image shown below.

If your data previews match, you are ready to follow along with the next video.

A screenshot of a computer

Description automatically generated

If both your data previews match, you are ready to follow along with the next video.

**Fungsi dan subkueri SQL: Persahabatan fungsional**

Dalam bacaan ini, Anda akan mempelajari tentang fungsi SQL dan bagaimana fungsi tersebut terkadang digunakan dengan subkueri. **Fungsi SQL** adalah alat yang dibangun ke dalam SQL untuk memungkinkan melakukan penghitungan. Subkueri (juga disebut kueri **dalam** atau kueri bersarang) adalah kueri di dalam kueri lain.

**Bagaimana fungsi SQL, berfungsi?**

Fungsi SQL membantu memungkinkan agregasi data. (Sebagai pengingat, agregasi data adalah proses pengumpulan data dari berbagai sumber untuk digabungkan menjadi satu kumpulan ringkasan.) Jadi, bagaimana cara kerja fungsi SQL? Kembali ke W3Schools, mari kita tinjau beberapa fungsi berikut untuk mendapatkan pemahaman yang lebih baik tentang cara menjalankan kueri ini:

* [**SQL MEMILIKI**](http://www-db.deis.unibo.it/courses/TW/DOCS/w3schools/sql/sql_having.asp.html)**:** Ini adalah ikhtisar klausa HAVING, termasuk apa itu klausa HAVING dan tutorial tentang cara dan waktu kerjanya.
* [**KASUS SQL**](https://www.w3schools.com/sql/sql_case.asp)**:** Jelajahi penggunaan pernyataan CASE dan contoh cara kerjanya.
* [**SQL JIKA**](https://www.w3schools.com/sql/func_mysql_if.asp)**:** Ini adalah tutorial fungsi IF dan memberikan contoh yang dapat Anda praktikkan.
* [**SQL JUMLAH**](http://www-db.deis.unibo.it/courses/TW/DOCS/w3schools/sql/sql_func_count.asp.html)**:** Fungsi COUNT sama pentingnya dengan fungsi lainnya, dan tutorial ini menawarkan banyak contoh untuk ditinjau.

**Subkueri - ceri di atasnya**

Anggaplah sebuah pertanyaan seperti kue. Sebuah kue dapat memiliki banyak lapisan di dalamnya dan bahkan lapisan di dalam lapisan tersebut. Masing-masing lapisan ini adalah subkueri kami, dan ketika Anda menggabungkan semua lapisan, Anda mendapatkan kue (kueri). Biasanya, Anda akan menemukan subkueri yang bersarang di klausa SELECT, FROM, dan/atau WHERE. Tidak ada sintaks umum untuk subquery, namun sintaks untuk subquery dasar adalah sebagai berikut:

PILIH tabel\_akun.\* DARI ( PILIH \* DARI transaksi.sf\_model\_feature\_2014\_01 WHERE hari\_minggu\_ = 'Jumat' ) tabel\_akun WHERE akun\_tabel.availabilitas = 'YA'

Anda akan menemukan bahwa, di dalam klausa SELECT pertama terdapat klausa SELECT lainnya. Klausa SELECT kedua menandai dimulainya subkueri dalam pernyataan ini. Ada banyak cara berbeda untuk menggunakan subkueri, dan sumber daya yang direferensikan akan memberikan panduan tambahan saat Anda belajar. Namun pertama-tama, mari kita rekap aturan subkueri.

Ada beberapa aturan yang harus dipatuhi oleh subkueri:

* Subkueri harus diapit dalam tanda kurung
* Subkueri hanya dapat memiliki satu kolom yang ditentukan dalam klausa SELECT. Namun jika Anda ingin subkueri membandingkan beberapa kolom, kolom tersebut harus dipilih di kueri utama.
* Subkueri yang mengembalikan lebih dari satu baris hanya dapat digunakan dengan beberapa operator nilai, seperti operator IN yang memungkinkan Anda menentukan beberapa nilai dalam klausa WHERE.
* Subkueri tidak dapat disarangkan dalam perintah SET. Perintah SET digunakan dengan UPDATE untuk menentukan kolom (dan nilai) mana yang akan diperbarui dalam sebuah tabel.

**Sumber daya tambahan**

Sumber daya berikut menawarkan panduan lebih lanjut tentang subkueri dan penggunaannya:

* [**Subkueri SQL:**](https://www.w3resource.com/sql/subqueries/understanding-sql-subqueries.php)Pengenalan mendetail ini mencakup definisi subquery, tujuannya dalam SQL, kapan dan bagaimana menggunakannya, serta apa hasilnya.
* [**Menulis subkueri dalam SQL**](https://mode.com/sql-tutorial/sql-sub-queries/)**:** Jelajahi dasar-dasar subkueri dalam tutorial interaktif ini, termasuk contoh dan latihan soal yang dapat Anda kerjakan

Saat Anda terus mempelajari lebih lanjut tentang penggunaan SQL, fungsi, dan subkueri, Anda akan menyadari betapa banyak waktu yang benar-benar dapat Anda hemat ketika menghafal tip dan trik ini.

**Fungsi dan subkueri SQL: Persahabatan fungsional**

Dalam bacaan ini, Anda akan mempelajari tentang fungsi SQL dan bagaimana fungsi tersebut terkadang digunakan dengan subkueri. **Fungsi SQL** adalah alat yang dibangun ke dalam SQL untuk memungkinkan melakukan penghitungan. Subkueri (juga disebut kueri **dalam** atau kueri bersarang) adalah kueri di dalam kueri lain.

**Bagaimana fungsi SQL, berfungsi?**

Fungsi SQL membantu memungkinkan agregasi data. (Sebagai pengingat, agregasi data adalah proses pengumpulan data dari berbagai sumber untuk digabungkan menjadi satu kumpulan ringkasan.) Jadi, bagaimana cara kerja fungsi SQL? Kembali ke W3Schools, mari kita tinjau beberapa fungsi berikut untuk mendapatkan pemahaman yang lebih baik tentang cara menjalankan kueri ini:

* [**SQL MEMILIKI**](http://www-db.deis.unibo.it/courses/TW/DOCS/w3schools/sql/sql_having.asp.html)**:** Ini adalah ikhtisar klausa HAVING, termasuk apa itu klausa HAVING dan tutorial tentang cara dan waktu kerjanya.
* [**KASUS SQL**](https://www.w3schools.com/sql/sql_case.asp)**:** Jelajahi penggunaan pernyataan CASE dan contoh cara kerjanya.
* [**SQL JIKA**](https://www.w3schools.com/sql/func_mysql_if.asp)**:** Ini adalah tutorial fungsi IF dan memberikan contoh yang dapat Anda praktikkan.
* [**SQL JUMLAH**](http://www-db.deis.unibo.it/courses/TW/DOCS/w3schools/sql/sql_func_count.asp.html)**:** Fungsi COUNT sama pentingnya dengan fungsi lainnya, dan tutorial ini menawarkan banyak contoh untuk ditinjau.

**Subkueri - ceri di atasnya**

Anggaplah sebuah pertanyaan seperti kue. Sebuah kue dapat memiliki banyak lapisan di dalamnya dan bahkan lapisan di dalam lapisan tersebut. Masing-masing lapisan ini adalah subkueri kami, dan ketika Anda menggabungkan semua lapisan, Anda mendapatkan kue (kueri). Biasanya, Anda akan menemukan subkueri yang bersarang di klausa SELECT, FROM, dan/atau WHERE. Tidak ada sintaks umum untuk subquery, namun sintaks untuk subquery dasar adalah sebagai berikut:

PILIH tabel\_akun.\* DARI ( PILIH \* DARI transaksi.sf\_model\_feature\_2014\_01 WHERE hari\_minggu\_ = 'Jumat' ) tabel\_akun WHERE akun\_tabel.availabilitas = 'YA'

Anda akan menemukan bahwa, di dalam klausa SELECT pertama terdapat klausa SELECT lainnya. Klausa SELECT kedua menandai dimulainya subkueri dalam pernyataan ini. Ada banyak cara berbeda untuk menggunakan subkueri, dan sumber daya yang direferensikan akan memberikan panduan tambahan saat Anda belajar. Namun pertama-tama, mari kita rekap aturan subkueri.

Ada beberapa aturan yang harus dipatuhi oleh subkueri:

* Subkueri harus diapit dalam tanda kurung
* Subkueri hanya dapat memiliki satu kolom yang ditentukan dalam klausa SELECT. Namun jika Anda ingin subkueri membandingkan beberapa kolom, kolom tersebut harus dipilih di kueri utama.
* Subkueri yang mengembalikan lebih dari satu baris hanya dapat digunakan dengan beberapa operator nilai, seperti operator IN yang memungkinkan Anda menentukan beberapa nilai dalam klausa WHERE.
* Subkueri tidak dapat disarangkan dalam perintah SET. Perintah SET digunakan dengan UPDATE untuk menentukan kolom (dan nilai) mana yang akan diperbarui dalam sebuah tabel.

**Sumber daya tambahan**

Sumber daya berikut menawarkan panduan lebih lanjut tentang subkueri dan penggunaannya:

* [**Subkueri SQL:**](https://www.w3resource.com/sql/subqueries/understanding-sql-subqueries.php)Pengenalan mendetail ini mencakup definisi subquery, tujuannya dalam SQL, kapan dan bagaimana menggunakannya, serta apa hasilnya.
* [**Menulis subkueri dalam SQL**](https://mode.com/sql-tutorial/sql-sub-queries/)**:** Jelajahi dasar-dasar subkueri dalam tutorial interaktif ini, termasuk contoh dan latihan soal yang dapat Anda kerjakan

Saat Anda terus mempelajari lebih lanjut tentang penggunaan SQL, fungsi, dan subkueri, Anda akan menyadari betapa banyak waktu yang benar-benar dapat Anda hemat ketika menghafal tip dan trik ini.



**Lengkap**