

Praktikum Teknologi Perekayasaan Data

Tanggal Praktikum : Friday, June 16, 2023

A Tour of Google Cloud Hands-on Labs

- Task 1. Accessing the Cloud Console

The screenshot shows the Google Cloud Platform dashboard for the project 'qwiklabs-gcp-02-55924724ad1b'. The left sidebar includes sections for Cloud overview, Products & solutions, PINNED services (APIs & Services, Billing, IAM & Admin, Marketplace, Compute Engine, Kubernetes Engine, Cloud Storage, BigQuery, VPC network, Cloud Run, SQL, Security), and ADD PEOPLE TO THIS PROJECT. The main area displays Project info (Project name: qwiklabs-gcp-02-55924724ad1b, Project number: 74238621903, Project ID: qwiklabs-gcp-02-55924724ad1b), API APIs (Requests (requests/sec) chart showing no data available for the selected time frame), Google Cloud Platform status (All services normal), Monitoring (Create my dashboard, Set up alerting policies, Create uptime checks, View all dashboards, Go to Monitoring), and API Error Reporting (No sign of any errors. Have you set up Error Reporting? Learn how to set up Error Reporting).

The screenshot shows a hands-on lab titled 'A Tour of Google Cloud Hands-on Labs' with a timer at 00:43:38. It features a sidebar with 'GSP282' and a list of tasks: Overview, Lab fundamentals, Task 1. Accessing the Cloud Console (selected), Task 2. Projects in the Cloud console, Task 3. Roles and permissions, Task 4. APIs and services, Task 5. Ending your lab, and Congratulations!. The main content area contains a question: 'What field is NOT found in the left pane?' with options: Project ID (unchecked), System admin (checked with a green checkmark), Password (unchecked), and Open Google console (unchecked). Below it is another question: 'The username in the left panel, which resembles googlexxxxx_student@qwiklabs.net, is a Cloud IAM identity.' with options: False (unchecked), True (checked with a green checkmark), and Submit (button). A note at the top says 'ANSWER THE FOLLOWING MULTIPLE CHOICE QUESTIONS TO REINFORCE YOUR UNDERSTANDING OF THE CONCEPTS COVERED SO FAR.'

- Task 2. Projects in the Cloud console

The image consists of two side-by-side screenshots of a Windows desktop environment.

Left Screenshot: A screenshot of the Google Cloud Console. The URL in the address bar is `https://console.cloud.google.com/welcome?project=qwiklabs-gcp-04-1dfef78f1dd7`. A modal window titled "Select a project" is open, showing a list of projects under "No organization". The "ALL" tab is selected. The list includes "Qwiklabs Resources" (ID: qwiklabs-resources) and "qwiklabs-gcp-04-1dfef78f1dd7" (ID: qwiklabs-gcp-04-1dfef78f1dd7). At the bottom right of the modal are "CANCEL" and "OPEN" buttons. The background shows the Google Cloud homepage with options like "Create a VM" and "Quick access".

Right Screenshot: A screenshot of a Qwiklabs assessment titled "A Tour of Google Cloud Hands-on Labs". The URL is `https://www.cloudskillsboost.google/focuses/2794parent=catalog`. The page displays a timer at 00:43:01. On the left, there's a sidebar with user information (student-94-5906eee8f62c) and a "Open Google Console" button. The main content area has a heading "Test your understanding" and a question: "Answer the following multiple choice questions to reinforce your understanding of the concepts covered so far." Below this is a list of four options with radio buttons: "Google Cloud Project" (selected), "Password", "Username", and "Cloud Storage bucket". A "Submit" button is at the bottom of this section. To the right of the main content is a sidebar titled "GSP282" with sections like "Overview", "Lab fundamentals", and "Task 2. Projects in the Cloud console" (which is currently active). At the bottom right of the sidebar is a "Congratulations!" message. The status bar at the bottom of the screen shows the date as 6/13/2023 and the time as 16:08.

- Task 3. Roles and permissions

Screenshot 1: Google Cloud IAM & Admin - Permissions for project

Screenshot 2: A Tour of Google Cloud Hands-on Labs - Task 3. Roles and permissions

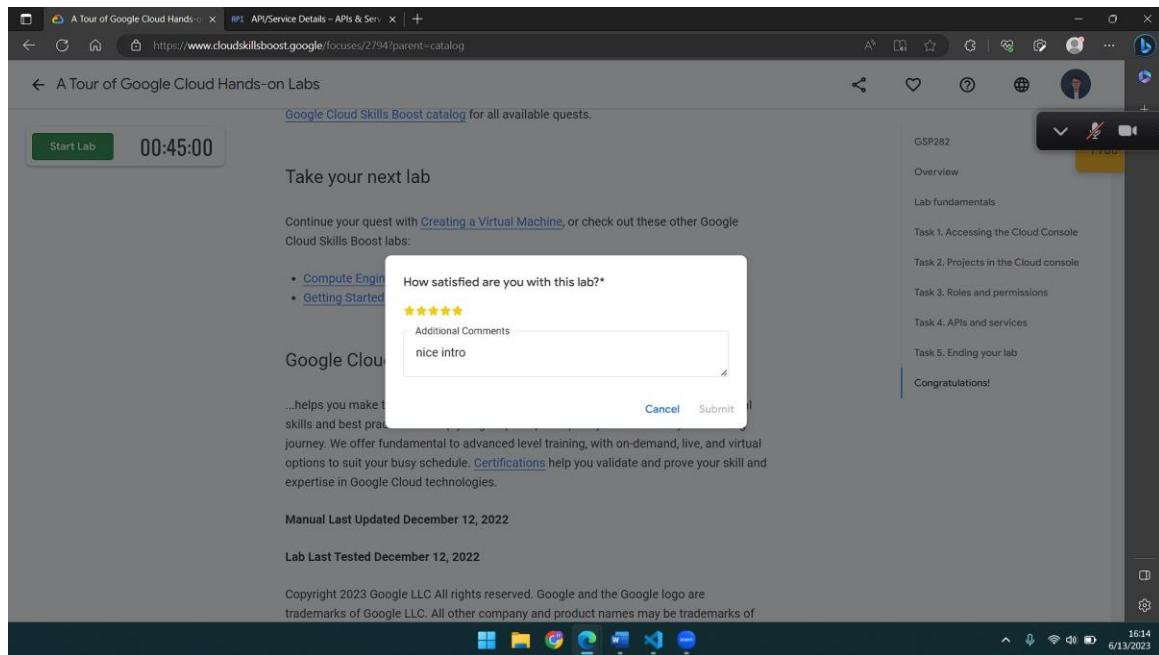
Screenshot 3: A Tour of Google Cloud Hands-on Labs - Task 3. Roles and permissions

- Task 4. APIs and services

The screenshot shows the Google Cloud Console interface. The URL in the address bar is <https://console.cloud.google.com/apis/library/dialogflow.googleapis.com?authuser=5&project=qwiklabs-gcp-04-1dfe78f1dd7>. The page displays the Dialogflow API product details, including its logo (an orange speech bubble icon), name (Dialogflow API), and description (Builds conversational interfaces). There are 'ENABLE' and 'TRY THIS API' buttons. Below the main title, there are tabs for OVERVIEW, PRICING, DOCUMENTATION, SUPPORT, and RELATED PRODUCTS. The OVERVIEW tab is selected. The Overview section includes a brief description: 'Builds conversational interfaces (for example, chatbots, and voice-powered apps and devices.)'. The Additional details section provides technical information: Type: SaaS & APIs, Last product update: 4/30/22, Category: Machine learning, Google Enterprise APIs, and Service name: dialogflow.googleapis.com.

The screenshot shows a browser window titled 'A Tour of Google Cloud Hands-on Labs'. The URL is <https://www.cloudskillsboost.google/focuses/2794?parent=catalog>. The page displays a task for enabling the Dialogflow API. It includes a timer at 00:41:49, a note about activity tracking, and a form to enter a username and project ID. A sidebar on the right lists tasks: GSP282, Overview, Lab fundamentals, Task 1. Accessing the Cloud Console, Task 2. Projects in the Cloud console, Task 3. Roles and permissions, Task 4. APIs and services (which is currently selected), Task 5. Ending your lab, and Congratulations!. The status bar at the bottom shows the date as 6/13/2023 and the time as 16:07.

- Task 5. Ending your lab



Penugasan

Build and Optimize Data Warehouses with BigQuery

1. Creating a Data Warehouse Through Joins and Unions

The screenshot shows a web-based lab interface for Google Cloud Skills Boost. The title is "Creating a Data Warehouse Through Joins and Unions". The time is 00:54:18. A sidebar on the right lists tasks: Task 1: The BigQuery console, Task 2: Create a new dataset to store your tables, Task 3: BigQuery project, Task 4: Explore the product sentiment dataset, Task 5: Examine the data, Task 6: Join datasets to find insights, Task 7: Append additional records, and Congratulations!. The main area shows a query editor with the following SQL:

```
SELECT
  SKU,
  name,
  sentimentScore,
  sentimentMagnitude
FROM
  `data-to-insights.ecommerce_products`
ORDER BY
  sentimentScore DESC
LIMIT 5
```

Below the query editor is a question: "What product has the highest sentiment?". The options are:

- G Noise-reducing Bluetooth Headphones
- G Noise-reducing Bluetooth Headphones
- USB wired soundbar - in store only
- Stylus Pen w/ LED Light

A green checkmark icon is next to the correct answer.

The screenshot shows the same web-based lab interface. The title is "Creating a Data Warehouse Through Joins and Unions". The time is 00:54:18. The sidebar shows the same tasks as the previous screenshot. The main area now shows a preview tab for a dataset with the question: "How many Aluminum Handy Emergency Flashlight have been ordered?". The options are:

- 90
- 85
- 0
- 66

A green checkmark icon is next to the correct answer. Below this is another question: "2. Click the Schema tab." followed by a question: "What data type are the sentimentScore and sentimentMagnitude fields?". The options are:

- FLOAT
- RECORD
- INTEGER

A green checkmark icon is next to the correct answer.

Creating a Data Warehouse Through Joins and Unions

00:51:15

3. Modify the previous query to add a filter to limit the results to just 08/02/2017.

Possible solution:

```
SELECT * FROM `ecommerce_sales_by_sku_2017`
WHERE _TABLE_SUFFIX = '0802'
```

Note: Another option to consider is to create a Partitioned Table which automatically can ingest daily sales data into the correct partition.

A UNION ALL join does not include duplicate records.

True
 False

Click Check my progress to verify the objective.

Append additional records Check my progress Assessment Completed!

Check complete. Points earned: 25. Message: Assessment Completed!

Checkpoints

- Create a new dataset to store the tables
- Explore the product advertising dataset
- Join datasets to find insights
- Append additional records

Task 4. Join datasets to find insights.

Screenshot copied to clipboard and saved Select here to mark up and share the image

2. Creating Date-Partitioned Tables in BigQuery

Creating Date-Partitioned Tables in BigQuery

00:54:39

Cooler. When you run a job, it's common to test queries from the job instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Console

Dataset student-64-cb421cc082b

Project w7LX3Zeu1z

Job ID query1bq-grp-03-a5664b

```
CREATE TABLE `ecommerce_stations`  
PARTITION BY EXPIRATION_DAYS=60,  
DESCRIPTION='weather stations with precipitation, partitioned  
by day'  
AS  
SELECT  
DATE(CAST(year AS INT64), CAST(month AS INT64), CAST(day AS  
INT64)) AS date,  
station, -- station name  
CAST(value AS INT64) AS precipitation,  
data_new_2017_stations AS stations  
WHERE stations.usaf = str)  
AS station_name; -- Stations may  
have multiple names  
group  
by station, data_new_prcd_good AS weather  
WHERE prcp < 49.9 -- Filter unknown values  
AND prcp > 0 -- Filter  
AND _TABLE_SUFFIX == '2018'
```

Click Check my progress to verify the objective.

Your task: Create a Partitioned Table Assessment Completed!

Confirm data partition expiration is working

To confirm you are only storing data from 60 days in the past up until today, run the DATE_DIFF query to get the age of your partitions, which are set to expire after 60 days.

Below is a query which tracks the average rainfall for the NOAA weather station in Wakayama, Japan which has significant precipitation.

Checkpoints

- Create a dataset named 'ecommerce'
- Create a new partitioned table based on date
- Your task: Create a Partitioned Table

partitioned table
Task 4. Your turn: create a partitioned table
Task 5. Confirm the oldest partition, age is at or below 60 days
Congratulations!

Creating Date-Partitioned Tables in BigQuery

00:57:27

Cooler. When you run a job, it's common to test queries from the job instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Console

Dataset student-64-cb421cc082b

Project w7LX3Zeu1z

Job ID query1bq-grp-03-a5664b

```
#standardSQL  
SELECT *  
FROM `data-to-insights.ecommerce_partition_by_day`  
WHERE data_formatted = '2016-08-01'
```

This time 25 KB or 0.025MB is processed, which is a fraction of what you queried.

2. Now run the below query, and note the total bytes to be processed.

```
#standardSQL  
SELECT  
SUM(data_to_insights.ecommerce_partition_by_day...  
WHERE data_formatted = '2016-07-01'
```

You should see This query will process 0 B when run.

Why was there 0 bytes processed?

The query is running from a saved View
 The query is running from query cache
 The query engine knows which date partitions exist before the query is run (and there is no 2016 partitions)

Submit

Checkpoints

- Create a dataset named 'ecommerce'
- Create a new partitioned table based on date
- Your task: Create a Partitioned Table

Task 4. Creating an auto-expiring partitioned table
Task 5. Your turn: create a partitioned table
Task 6. Confirm the oldest partition, age is at or below 60 days
Congratulations!

MV Musim yang Selanjutnya Creating Date-Partitioned Tables

Query results - BigQuery - qwi... +

Creating Date-Partitioned Tables in BigQuery

End Lab 00:57:52

Note: Partitions within partitioned tables on your lab account will auto-expire after 60 days from the value in your date column. Your personal Google Cloud account with billing enabled will let you have partitioned tables that don't expire.

Caveats: When you are in the console, do not delete from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Console

Username: student-64-eb421cc002b: []

Password: w67tK312qcIz []

GCP Project ID: qwiklabs-gcp-03-a5664b []

For the purposes of this lab, the remaining queries will be run against partitioned tables that have already been created.

Click Check my progress to verify the objective.

Create a new partitioned table based on date ✓ Check my progress Assessment Completed!

Checkpoints

Create a dataset named 'ecommerce' Check my progress 20 / 30

Create a new partitioned table based on date Check my progress 35 / 35

Your turn: Create a Partitioned Table Check my progress 0 / 35

Task 4: Creating an auto-expiring partitioned table

Task 5: Your turn: create a partitioned table

Task 6: Confirm the oldest partition age is at or below 60 days

Congratulations!

Task 3. View data processed with a partitioned table

1. Run the below query, and note the total bytes to be processed:

```
#standardSQL  
SELECT *  
Check complete. Ready to run? Assessment Completed!
```

MV Musim yang Selanjutnya Creating Date-Partitioned Tables

Query results - BigQuery - qwi... +

Creating Date-Partitioned Tables in BigQuery

End Lab 00:58:19

The Query Validator tells you how much data this query will process.

2. Click Run

Notice that the query still processes 1.74 GB even though it returns 0 results. Why? The query engine needs to scan all records in the dataset to see if they satisfy the matching condition in the WHERE clause. It must look at each record to compare the date against the condition of 20180708.

Additionally, the LIMIT 5 does not reduce the total amount of data processed, which is a common misconception.

Why did the previous query return 0 records but still scan through 1.74GB of data?

- The query engine has the metadata for each partition stored, but still needs to scan all records even if the table is partitioned.
- Before the query runs, the query engine does not know whether 2018 data exists to satisfy the WHERE clause condition and it needs to scan through all records in a non-partitioned table.
- The query was written incorrectly

Submit

Checkpoints

Create a dataset named 'ecommerce' Check my progress 50 / 30

Create a new partitioned table based on date Check my progress 0 / 35

Your turn: Create a Partitioned Table Check my progress 0 / 35

Task 4: Creating an auto-expiring partitioned table

Task 5: Your turn: create a partitioned table

Task 6: Confirm the oldest partition age is at or below 60 days

Congratulations!

Common use-cases for date-partitioned tables

Scanning through the entire dataset everytime to compare rows against a WHERE

MV Musim yang Selanjutnya Creating Date Partitioned Tables

BigQuery - qwiklabs-gcp-03-a5664b +

Creating Date-Partitioned Tables in BigQuery

End Lab 00:58:51

Caveats: When you are in the console, do not delete from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Console

Username: student-64-eb421cc002b: []

Password: w67tK312qcIz []

GCP Project ID: qwiklabs-gcp-03-a5664b []

3. Name your dataset ecommerce.

Leave the other options at their default values (Data Location, Default table Expiration).

4. Click Create dataset.

Click Check my progress to verify the objective.

Create a dataset named ecommerce ✓ Check my progress Assessment Completed!

Checkpoints

Create a dataset named 'ecommerce' Check my progress 30 / 30

Create a new partitioned table based on date Check my progress 0 / 35

Your turn: Create a Partitioned Table Check my progress 0 / 35

Task 4: Creating an auto-expiring partitioned table

Task 5: Your turn: create a partitioned table

Task 6: Confirm the oldest partition age is at or below 60 days

Congratulations!

Task 2. Creating tables with date partitions

A partitioned table is a table that is divided into segments based on a date column. This makes it easier to manage and query your data. By dividing a large table into smaller partitions, you

Google Cloud BigQuery interface showing a partitioned table named "partition_by_day". The table has columns "date_formatted" and "fullviewid". A preview pane shows 14 rows of data from June 2017.

Row	date_formatted	fullviewid
1	2017/06/11	149707110301300881
2	2017/06/15	201843200100000034
3	2017/06/19	377075823273447945
4	2017/06/15	6267058583064310776
5	2017/06/15	240151583046001039
6	2017/06/15	31991051348941791
7	2017/06/15	1261036150509902075
8	2017/06/15	77046191321830076
9	2017/06/15	14456048054744004
10	2017/06/15	664309754546186213
11	2017/06/15	77630283930151118
12	2017/06/15	6793099388750000655
13	2017/06/15	84283011440040673
14	2017/06/15	05752674749000719

3. Troubleshooting and Solving Data Join Pitfalls

Troubleshooting and Solving Data Join Pitfalls

Question: When you run the console, do you notice any errors or warnings? Does this mean your code is incorrect?

Open Google Cloud Console

Dataset: student-64-8c3eb3d8465

Project ID: 174035114096

gskilabs-gcp-02-74468f

Student Resources

- Get meaningful insights with logical step-by-step instructions
- Step-by-step start - DevSkills Preview

How many products are on clearance?

93
52
✓ 92
0

Submit

Note: For a CROSS JOIN you will notice there is no join condition (e.g. ON or USING). The field is simply multiplied against the first dataset or 05 discount across all items.

See the impact of unintentionally adding more than one record in the discount table.

Clear the previous query and run the below query to insert two more records into the promotion table:

```
INSERT INTO ecommerce.sale.write_promotion (discount)
VALUES (.04),
(.09);
```

Checkpoints

- Create a new dataset: Check my progress / 25
- Identify a key field in your ecommerce dataset: Check my progress / 25
- Pitfall: unnecessary key: Check my progress / 25
- Join pitfall solution: Check my progress / 0 / 25

Task 4: Join pitfall solution: use distinct SKU before joining

Troubleshooting and Solving Data Join Pitfalls

Question: How many products are missing?

Answer: 819 products are missing (SKU IS NULL) from your product inventory dataset.

Clear the previous query and run the below query to confirm using one of the specific SKUs from the website dataset:

```
#standardSQL
# You can also click here and centre
# the table: https://data-faith-public.firebaseio.com/ecommerce-products
WHERE SKU = '200001488577'
# every returns zero results
```

Why might the product inventory dataset be missing SKUs?

- Some SKUs could be digital products that you do not store in warehouse inventory
- Old products could be past website orders and no longer offering in-current inventory
- There is legitimate missing data from inventory and should be tracked
- All-of-the-above

Submit

Now, what about the reverse situation? Are there any products in the product inventory dataset but missing from the website?

1. Write a query using a different join type to investigate.

Possible solution

Checkpoints

- Create a new dataset: Check my progress / 25
- Identify a key field in your ecommerce dataset: Check my progress / 25
- Pitfall: unnecessary key: Check my progress / 25
- Join pitfall solution: Check my progress / 0 / 25

Task 4: Join pitfall solution: use distinct SKU before joining

00:52:47

Task 5. Join pitfall solution: use distinct SKUs before joining

Checkpoints

- Create a new dataset: **Check my progress** / 25
- Identify a key field in your ecommerce dataset: **Check my progress** / 25
- Pitfall: non-unique key: **Check my progress** / 25
- Join pitfall solution: **Check my progress** / 0 / 25

Cheat: When you are in the console, do not delete from the left navigation. Doing so may cause your account to be blocked.

Open Google Cloud Console

Username: student-84-8cdeb3e8465

Password: 174065t1m0Fe

KDF Project ID: quicklabs-gcp-02-7446bf

Student Resources

- Get meaningful insights with Google BigQuery
- Bigquery_Quick_Start - Okwikkos Preview

True or False: Many inventory SKU values are NULL.

True
 False

Submit

How many SKUs are missing from your product inventory set?

1. Write a query to filter on NULL values from the inventory table.

Possible solution:

```
standardSQL
# Find product SKUs in website table but not in product
# inventory table
SELECT DISTINCT
    website.productSKU AS website_SKU,
    inventory.SKU AS inventory_SKU
```

00:53:17

Now you're ready to join against your product inventory dataset again.

1. Clear the previous query and run the below query:

Checkpoints

- Create a new dataset: **Check my progress** / 25
- Identify a key field in your ecommerce dataset: **Check my progress** / 25
- Pitfall: non-unique key: **Check my progress** / 25
- Join pitfall solution: **Check my progress** / 0 / 25

Cheat: When you are in the console, do not delete from the left navigation. Doing so may cause your account to be blocked.

Open Google Cloud Console

Username: student-84-8cdeb3e8465

Password: 174065t1m0Fe

KDF Project ID: quicklabs-gcp-02-7446bf

Student Resources

- Get meaningful insights with Google BigQuery
- Bigquery_Quick_Start - Okwikkos Preview

How many records were returned? All 1,009 distinct SKUs?

No, just 1,090 records
 Yes, all 1,090 records

Submit

It seems 819 SKUs were lost after joining the datasets. Investigate by adding more specificity in your fields (one SKU column from each dataset).

2. Clear the previous query and run the below query:

standardSQL

```
# pull ID fields from both tables
SELECT DISTINCT
    website.productSKU AS website_SKU,
    inventory.SKU AS inventory_SKU
FROM data-to-insights.ecommerce.all_sessions_raw AS website
JOIN data-to-insights.ecommerce.products AS inventory
ON website.productSKU = inventory.SKU
```

00:54:19

Cheat: When you are in the console, do not delete from the left navigation. Doing so may cause your account to be blocked.

Open Google Cloud Console

Username: student-84-8cdeb3e8465

Password: 174065t1m0Fe

KDF Project ID: quicklabs-gcp-02-7446bf

Student Resources

- Get meaningful insights with Google BigQuery
- Bigquery_Quick_Start - Okwikkos Preview

Is the dog Filibee properly showing a stock level of 154?

Yes, it is at 154
 No, it is now at 462 showing three times (one for each record)

Submit

Oh no! It is 154 x 3 = 462 or triple counting the inventory! This is called an unintentional cross join (a topic that will be revisited later).

Click Check my progress to verify the objective.

Check my progress

Success: Pitfall: non-unique key

Task 6. Join pitfall solution: use distinct SKUs before joining

Check complete. Points earned: 25. Message: Success: Pitfall: non-unique key

00:53:17

Now you're ready to join against your product inventory dataset again.

1. Clear the previous query and run the below query:

Checkpoints

- Create a new dataset: **Check my progress** / 25
- Identify a key field in your ecommerce dataset: **Check my progress** / 25
- Pitfall: non-unique key: **Check my progress** / 25
- Join pitfall solution: **Check my progress** / 0 / 25

Cheat: When you are in the console, do not delete from the left navigation. Doing so may cause your account to be blocked.

Open Google Cloud Console

Username: student-84-8cdeb3e8465

Password: 174065t1m0Fe

KDF Project ID: quicklabs-gcp-02-7446bf

Student Resources

- Get meaningful insights with Google BigQuery
- Bigquery_Quick_Start - Okwikkos Preview

How many records were returned? All 1,009 distinct SKUs?

No, just 1,090 records
 Yes, all 1,090 records

Submit

It seems 819 SKUs were lost after joining the datasets. Investigate by adding more specificity in your fields (one SKU column from each dataset).

2. Clear the previous query and run the below query:

standardSQL

```
# pull ID fields from both tables
SELECT DISTINCT
    website.productSKU AS website_SKU,
    inventory.SKU AS inventory_SKU
FROM data-to-insights.ecommerce.all_sessions_raw AS website
JOIN data-to-insights.ecommerce.products AS inventory
ON website.productSKU = inventory.SKU
```

00:54:19

Cheat: When you are in the console, do not delete from the left navigation. Doing so may cause your account to be blocked.

Open Google Cloud Console

Username: student-84-8cdeb3e8465

Password: 174065t1m0Fe

KDF Project ID: quicklabs-gcp-02-7446bf

Student Resources

- Get meaningful insights with Google BigQuery
- Bigquery_Quick_Start - Okwikkos Preview

Is the dog Filibee properly showing a stock level of 154?

Yes, it is at 154
 No, it is now at 462 showing three times (one for each record)

Submit

Oh no! It is 154 x 3 = 462 or triple counting the inventory! This is called an unintentional cross join (a topic that will be revisited later).

Click Check my progress to verify the objective.

Check my progress

Success: Pitfall: non-unique key

00:54:19

Cheat: When you are in the console, do not delete from the left navigation. Doing so may cause your account to be blocked.

Open Google Cloud Console

Username: student-84-8cdeb3e8465

Password: 174065t1m0Fe

KDF Project ID: quicklabs-gcp-02-7446bf

Student Resources

- Get meaningful insights with Google BigQuery
- Bigquery_Quick_Start - Okwikkos Preview

Is the dog Filibee properly showing a stock level of 154?

Yes, it is at 154
 No, it is now at 462 showing three times (one for each record)

Submit

Oh no! It is 154 x 3 = 462 or triple counting the inventory! This is called an unintentional cross join (a topic that will be revisited later).

Click Check my progress to verify the objective.

Check my progress

Success: Pitfall: non-unique key

Task 6. Join pitfall solution: use distinct SKUs before joining

Check complete. Points earned: 25. Message: Success: Pitfall: non-unique key

00:54:49

FROM data-to-insights.ecommerce.all_sessions_raw AS website
JOIN data-to-insights.ecommerce_products AS inventory
ON website.productSKU = inventory.SKU
WHERE productSKU = 'G00EPUJ019899'

What happens when you join the website table and the product inventory table on SKU? Do you now have inventory stock levels for the products?

No, there is no inventory data, the join did not work.
 Yes, there are inventory levels but the stockLevel is showing three times (one for each record).
 Yes, there is inventory data and everything looks fine.

Submit

Checkpoints

- Create a new dataset: Check my progress / 25
- Identify a key field in your ecommerce dataset: Check my progress / 25
- Pitfall: non-unique key: Check my progress / 0 / 25
- Join pitfall solution: Check my progress / 0 / 25

Task 5: Pitfall: non-unique key
Task 6: Join pitfall solution: use distinct SKUs before joining
Congratulations!

00:54:49

SELECT
SKU,
name,
stockLevel
FROM data-to-insights.ecommerce.products
WHERE SKU = 'G00EPUJ019899'

Is the SKU unique in the product inventory dataset?

Yes, just one record is returned.
 No, there are duplicate SKUs in the inventory dataset.

Submit

Checkpoints

- Create a new dataset: Check my progress / 25
- Identify a key field in your ecommerce dataset: Check my progress / 25
- Pitfall: non-unique key: Check my progress / 0 / 25
- Join pitfall solution: Check my progress / 0 / 25

Task 5: Pitfall: non-unique key
Task 6: Join pitfall solution: use distinct SKUs before joining
Congratulations!

Join pitfall: Unintentional many-to-one SKU relationship

00:55:21

FROM products,
productsSKU
INNER JOIN data-to-insights.ecommerce.all_sessions_raw
WHERE productSKU = 'G00EPUJ019899'

2. Click Run

v2ProductName	productSKU
TinyCut Dog Frisbee	G00EGRU019099
2" Dog Frisbee	G00EGRU019099
Google / inch Dog Flying Disc Blue	G00EGRU019099

What do you notice about the product names?

They are mostly the same except for a few characters.
 They are exactly the same.

Submit

Checkpoints

- Create a new dataset: Check my progress / 25
- Identify a key field in your ecommerce dataset: Check my progress / 25
- Pitfall: non-unique key: Check my progress / 0 / 25
- Join pitfall solution: Check my progress / 0 / 25

Task 5: Pitfall: non-unique key
Task 6: Join pitfall solution: use distinct SKUs before joining
Congratulations!

Joining website data against your product inventory list

Now see the impact of joining on a dataset with multiple products for a single SKU. First explore the product inventory dataset (the products table) to see if this SKU is unique there.

End Lab 00:55:24

Note: Try replacing STRING_AGG() with ARRAY_AGG() instead. Pretty cool, right? BigQuery natively supports nested array values. You can learn more from the [Work with arrays guide](#).

Basename: student-84-8c3eb3d465
Password: 17400510tDFe
BigQuery Project ID: quicklabs-grp-02-7446bf1

Student Resources:

- [Get meaningful insights with Google BigQuery](#)
- [BigQuery: Quick Start - Overview Preview](#)

Checkpoints

- Create a new dataset: [Check my progress](#) 25 / 25
- Identify a key field in your ecommerce dataset: [Check my progress](#) 25 / 25
- Pitfall: non-unique key: [Check my progress](#) 0 / 25
- Join pitfall solution: [Check my progress](#) 0 / 25

Congratulations!

You will see why this many-to-many data relationship will be an issue in the next section.

Click Check my progress to verify the objective:

Identify a key field in your ecommerce dataset

Success: Identify a key field in your ecommerce dataset

Check complete. Points earned: 25. Message: Success! Identify a key field in your ecommerce dataset

End Lab 00:56:53

Note: When you are in the console, do not clear the query editor or you may lose history. Doing so may cause your account to be blocked. [Learn more](#)

Basename: student-84-8c3eb3d465
Password: 17400510tDFe
BigQuery Project ID: quicklabs-grp-02-7446bf1

Student Resources:

- [Get meaningful insights with Google BigQuery](#)
- [BigQuery: Quick Start - Overview Preview](#)

Checkpoints

- Create a new dataset: [Check my progress](#) 25 / 25
- Identify a key field in your ecommerce dataset: [Check my progress](#) 0 / 25
- Pitfall: non-unique key: [Check my progress](#) 0 / 25
- Join pitfall solution: [Check my progress](#) 0 / 25

Congratulations!

Do some product names have more than one SKU? Look at the query results to confirm.

Yes **No**

Which product has the most SKUs associated?

Google Sunglasses
 Wazz Women's Typography Short Sleeve Tee
 Android Women's Short Sleeve Badge Tee Dark Heather

Submit

The ecommerce website catalog shows that each product name may have multiple options (size, color) – which are sold as separate SKUs.

End Lab 00:57:24

Note: When you are in the console, do not clear the query editor or you may lose history. Doing so may cause your account to be blocked. [Learn more](#)

Basename: student-84-8c3eb3d465
Password: 17400510tDFe
BigQuery Project ID: quicklabs-grp-02-7446bf1

Student Resources:

- [Get meaningful insights with Google BigQuery](#)
- [BigQuery: Quick Start - Overview Preview](#)

Checkpoints

- Create a new dataset: [Check my progress](#) 25 / 25
- Identify a key field in your ecommerce dataset: [Check my progress](#) 0 / 25
- Pitfall: non-unique key: [Check my progress](#) 0 / 25
- Join pitfall solution: [Check my progress](#) 0 / 25

Congratulations!

Examine the relationship between SKU & Name

Now determine which products have more than one SKU and which SKUs have more than one Product Name.

1. Clear the previous query and run the below query to determine if some product names have more than one SKU. The use of the STRING_AGG function to aggregate all the product SKUs that are associated with one product name into comma separated values.

```
SELECT
    productName,
    COUNT(DISTINCT productSKU) AS SKU_Count,
    STRING_AGG(DISTINCT productSKU LIMIT 5) AS SKU
FROM
    `data-to-insights.ecommerce.all_sessions_raw`
GROUP BY productName
HAVING SKU_Count > 1
ORDER BY SKU_Count DESC
```

00:57:24

End Lab

Checkpoints

- Create a new dataset Check my progress 25 / 25
- Identify a key field in your ecommerce dataset Check my progress 0 / 25
- Pitfall: non-unique key Check my progress 0 / 25
- Join pitfall solution Check my progress 0 / 25

Task 5. Pitfall: non-unique key
Task 6. Join pitfall solution: use distinct fields before joining
Congratulations!

Student Resources

- Get More Helpful Insights with Google BigQuery
- BigQuery Quick Start - Qwiklabs Preview

Code Editor

```
standardSQL
# Find the count of unique SKUs
SELECT
    COUNT(DISTINCT
        productSKU
    ) AS numDistinctSKUs
FROM `data-to-insights.ecommerce.all_sessions_raw`
```

How many DISTINCT SKUs are returned?

- 119 distinct SKUs
- 1,909 distinct SKUs ✓
- 2,723 distinct SKUs

Submit

BigQuery

Explorer

Search: bigque

Viewing workspace resources: SHOW STARRED ONLY

SCHEMA

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
fullVisitorId	STRING	NULLABLE					
channelGrouping	STRING	NULLABLE					
time	INTEGER	NULLABLE					
country	STRING	NULLABLE					
city	STRING	NULLABLE					
totalTransactionRevenue	INTEGER	NULLABLE					
transactions	INTEGER	NULLABLE					
timeOnSite	INTEGER	NULLABLE					
pageviews	INTEGER	NULLABLE					
sessionQualityDim	INTEGER	NULLABLE					
date	STRING	NULLABLE					
visitId	INTEGER	NULLABLE					

Data-to-insights' was started.

PERSONAL HISTORY PROJECT HISTORY **REFRESH**

00:58:58

End Lab

Checkpoints

- Create a new dataset Check my progress 25 / 25
- Identify a key field in your ecommerce dataset Check my progress 0 / 25
- Pitfall: non-unique key Check my progress 0 / 25
- Join pitfall solution Check my progress 0 / 25

Task 5. Pitfall: non-unique key
Task 6. Join pitfall solution: use distinct fields before joining
Congratulations!

Student Resources

- Get More Helpful Insights with Google BigQuery
- BigQuery Quick Start - Qwiklabs Preview

Code Editor

```
CREATE OR REPLACE TABLE ecommerce.all_sessions_raw AS
SELECT
    fullVisitorId,
    channelGrouping,
    time,
    country,
    city,
    totalTransactionRevenue,
    transactions,
    timeOnSite,
    pageviews,
    sessionQualityDim,
    date,
    visitId
FROM `data-to-insights.ecommerce.sessions`
```

Success: Create a new dataset

Task 2. Pin the lab project in BigQuery

Scenario: Your team provides you with a new dataset on the inventory stock levels for each of your products for sale on your ecommerce website. You want to become familiar with the products on the website and the fields you could use to potentially join on to other datasets.

The project with the new dataset is **data-to-insights**.

1. Click Navigation menu → BigQuery.

The Welcome to BigQuery in the Cloud Console message box opens.

Check complete: Points earned: 25. Message: Success: Create a new dataset

Notes: The Welcome to BigQuery in the Cloud Console message box provides a link

00:50:14

What was the Impact of the CROSS JOIN?

Answer: Since there are 3 discount codes to cross join on, you are multiplying the original dataset by 3.

Note: This behavior isn't limited to cross joins, with a normal join you can unintentionally cross join when the data relationships are many-to-many this can easily result in returning millions or even billions of records unintentionally.

The solution is to know your data relationships before you join and don't assume keys are unique.

Click Check my progress to verify the objective.

Join pitfall solution

Check my progress

Success: Join pitfall solution

Congratulations!

You've completed this lab and worked through some serious SQL join pitfalls by identifying duplicate records and knowing when to use each type of JOIN. Nice work!

4. Working with JSON, Arrays, and Structs in BigQuery

01:03:30

Possible solution:

```
#standardSQL
SELECT
  p.name,
  split.time
FROM racing.race_results AS r
, UNNEST(r.participants) AS p
, UNNEST(p.splits) AS split
WHERE split.time = 23.2;
```

Row	name	split_time
1	Kiplister	23.2

Click Check my progress to verify the objective.

Execute the query to see which runner ran fastest lap time

Check my progress

Assessment Completed!

Check my progress

Check my progress

Check my progress

Check my progress

Congratulations!

SANDBOX Set up billing to upgrade to the full BigQuery experience. Learn more

Explorer

- SQL workspace
- Data transfers
- Scheduled queries
- Analytics Hub
- Dataform
- Partner Center
- Migration
- Assessment
- SQL translation
- Administration
- Monitoring
- Capacity management
- Release Notes

Untitled **RUN** **SAVE** **SHARE** **SCHEDULE** **MORE** **DISMISS** **UPGRADE**

```
1 #standardSQL
2 SELECT
3   p.name,
4   split.time
5 FROM racing.race_results AS r
6 , UNNEST(r.participants) AS p
7 , UNNEST(p.splits) AS split
8 WHERE split.time = 23.2;
```

Query results

Row	name	split_time
1	Kiplister	23.2

Press Alt+F1 for Accessibility Options.

PERSONAL HISTORY **PROJECT HISTORY** **REFRESH**

Working with JSON, Arrays, and Structs in BigQuery

End Lab 01:04:14

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked.

Open Google Console

Username: student-04-e79d9a963fb1

Password: 98t8d10Qn198

GCP Project ID: quicklabs-gcp-03-be3625

```
FROM racing.race_results AS r
, UNNEST(r.participants) AS p
, UNNEST(p.splits) AS split_times
WHERE p.name LIKE 'R%'
GROUP BY p.name
ORDER BY total_race_time ASC;
```

Row	name	total_race_time
1	Rudisha	102.19999999999999
2	Rotich	103.6

Click **Check my progress** to verify the objective.

Execute the query that will list the total race time for racers whose names begin with R

Assessment Completed!

on array field

- Create a dataset and a table to ingest JSON data **Check my progress** 15 / 15
- Execute the query to COUNT how many racers were there in total **Check my progress** 20 / 20
- Execute the query that will list the total race time for racers whose names begin with R **Check my progress** 10 / 10
- Execute the query to see which runner ran fastest lap time **Check my progress** 0 / 10

Task 4. Querying datasets that already have arrays

Task 5. Introduction to STRUCTs

Task 6. Practice with STRUCTs and arrays

Task 7. Lab question: STRUCT()

Task 8. Lab question: Unpacking arrays with UNNEST()

Task 9. Filtering within array values

Congratulations!

Task 9. Filtering within array values

Working with JSON, Arrays, and Structs in BigQuery

Query results - BigQuery - qwiklab

Google Cloud

BigQuery

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#)

DISMISS UPGRADE

Analysis

SQL workspace

Data transfers

Scheduled queries

Analytics Hub

Dataform

Partner Center

Migration

Assessment

SQL translation

Administration

Monitoring

Capacity management

Release Notes

Explorer

Untitled

RUN SAVE SHARE SCHEDULE MORE

```
1 #standardSQL
2 SELECT
3   p.name
4   SUM(split_times) as total_race_time
5   FROM `racing.race_results` AS r
6   , UNNEST(r.participants) AS p
7   , UNNEST(p.splits) AS split_times
8   WHERE p.name LIKE 'R%'
9   GROUP BY p.name
10 ORDER BY total_race_time ASC;
```

Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	name	total_race_time
1	Rudisha	102.19999999999999
2	Rotich	103.6

PERSONAL HISTORY PROJECT HISTORY

REFRESH

Working with JSON, Arrays, and Structs in BigQuery

End Lab 01:04:14

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked.

Open Google Console

Username: student-04-e79d9a963fb1

Password: 98t8d10Qn198

GCP Project ID: quicklabs-gcp-03-be3625

```
#standardSQL
SELECT COUNT(p.name) AS racer_count
FROM `racing.race_results` AS r, UNNEST(r.participants) AS p
```

Row	racer_count
1	8

Answer: There were 8 racers who ran the race.

Click **Check my progress** to verify the objective.

Execute the query to COUNT how many racers were there in total

Assessment Completed!

on array field

- Create a dataset and a table to ingest JSON data **Check my progress** 15 / 15
- Execute the query to COUNT how many racers were there in total **Check my progress** 20 / 20
- Execute the query that will list the total race time for racers whose names begin with R **Check my progress** 10 / 10
- Execute the query to see which runner ran fastest lap time **Check my progress** 0 / 10

Task 4. Querying datasets that already have arrays

Task 5. Introduction to STRUCTs

Task 6. Practice with STRUCTs and arrays

Task 7. Lab question: STRUCT()

Task 8. Lab question: Unpacking arrays with UNNEST()

Task 9. Filtering within array values

Congratulations!

Check complete. Points earned: 10. Message: Assessment Completed!

Working with JSON, Arrays, and Structs in BigQuery

Query results - BigQuery - qwiklabs-gcp-03-ba3625968f93

Untitled

```
1 #standardSQL
2 SELECT * FROM racing.race_results
```

Query results

Row	race	participants.name	participants.splits
1	800M	Rudisha	28.3 26.4 26.1
		Makhloufi	24.5 25.4

Working with JSON, Arrays, and Structs in BigQuery

End Lab

01:05:15

Caution: When you are in the console, do not deviate from the lab instructions, doing so may cause your account to be blocked.

Open Google Console

Username: student-04-e7d9e963f81

Password: 9818d10nI9B

GCP Project ID: qwiklabs-gcp-03-ba3625968f93

participants

participants.name	participants.splits
String	Float

REPEATED

Which field is the STRUCT? How do you know?

The participants field is the STRUCT because it is of type RECORD.

Which field is the ARRAY?

The participants.splits field is an array of floats inside of the parent participants struct. It has a REPEATED Mode which indicates an array. Values of that array are called nested values since they are multiple values inside of a single field.

Click Check my progress to verify the objective.

Create a dataset and a table to ingest JSON data

Check my progress

Assessment Completed!

Execute the query to see how many unique products were viewed

Check my progress 15 / 15

Execute the query to use the UNNEST() on array field

Check my progress 15 / 15

Create a dataset and a table to ingest JSON data

Check my progress 0 / 20

Execute the query to COUNT how many racers were there in total

Check my progress 0 / 10

Execute the query that will list the total

Task 4: Querying datasets that already have arrays

Task 5: Introduction to STRUCTs

Task 6: Practice with STRUCTs and arrays

Task 7: Lab question: STRUCTs

Task 8: Lab question: Unpacking arrays with UNNEST()

Task 9: Filtering within array values

Congratulations!

Practice querying nested and repeated fields

Working with JSON, Arrays, and Structs in BigQuery

Untitled

race_results

SCHEMA

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
racer	STRING	NULLABLE					
participants	RECORD	REPEATED					
name	STRING	NULLABLE					
splits	FLOAT	REPEATED					

EDIT SCHEMA

VIEW ROW ACCESS POLICIES

"race_results" created. GO TO TABLE

MV Gingham Check - JKT4 Working with JSON, Arrays, and Structs in BigQuery Query results - BigQuery - qwiklabs

Working with JSON, Arrays, and Structs in BigQuery

End Lab 01:06:48

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Console

Username: student-04-e79d9a963f8f

Password: 90t8d10n198

GCP Project ID: qwiklabs-gcp-03-ba3629

What do you notice about the field aliases? Since there are fields nested within the struct (name and split are a subset of runner) you end up with a dot notation.

What if the runner has multiple split times for a single race (like time per lap)?

How could you have multiple split times within a single record? Hint: the splits all have the same numeric datatype.

Store each split time in a separate table called race_splits

Use a SQL UNION to join the race and split details

Store each split time in a separate STRING field with STRING_AGG

Store each split time as an element in an ARRAY of splits

Submit

With an array of course!

2. Run the below query to confirm:

```
#standardSQL
SELECT STRUCT("Kudisha" as name, [23.4, 26.3, 26.4, 26.1] as splits) AS runner
```

Execute the query to see how many unique products were viewed

Check my progress 15 / 15

Execute the query to use the UNNEST() on array field

Check my progress 15 / 15

Create a dataset and a table to ingest JSON data

Check my progress 0 / 20

Execute the query to COUNT how many racers were there in total

Check my progress 0 / 10

Execute the query that will list the total

Task 4. Querying datasets that already have arrays

Task 5. Introduction to STRUCTs

Task 6. Practice with STRUCTs and arrays

Task 7. Lab question: STRUCT()

Task 8. Lab question: Unpacking arrays with UNNEST()

Task 9. Filtering within array values

Congratulations!

MV Gingham Check - JKT4 Working with JSON, Arrays, and Structs in BigQuery - BigQuery - qwiklabs-gcp-03-be

Working with JSON, Arrays, and Structs in BigQuery

End Lab 01:07:19

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Console

Username: student-04-e79d9a963f8f

Password: 90t8d10n198

GCP Project ID: qwiklabs-gcp-03-ba3629

They are not STRUCTs

Because they are all ARRAYS

This is an invalid data type

Submit

In a BigQuery schema, an ARRAY field is noted as a REPEATED Mode. Search for REPEATED in the Google Analytics schema. How many ARRAYS are present in this dataset?

1

5

11

32

Submit

As you can imagine, there is an incredible amount of website session data stored for a modern ecommerce website.

Execute the query to see how many unique products were viewed

Check my progress 15 / 15

Execute the query to use the UNNEST() on array field

Check my progress 15 / 15

Create a dataset and a table to ingest JSON data

Check my progress 0 / 20

Execute the query to COUNT how many racers were there in total

Check my progress 0 / 10

Execute the query that will list the total

Task 4. Querying datasets that already have arrays

Task 5. Introduction to STRUCTs

Task 6. Practice with STRUCTs and arrays

Task 7. Lab question: STRUCT()

Task 8. Lab question: Unpacking arrays with UNNEST()

Task 9. Filtering within array values

Congratulations!

MV Gingham Check - JKT4 Working with JSON, Arrays, and Structs in BigQuery - BigQuery - qwiklabs-gcp-03-be

Working with JSON, Arrays, and Structs in BigQuery

End Lab 01:07:19

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Console

Username: student-04-e79d9a963f8f

Password: 90t8d10n198

GCP Project ID: qwiklabs-gcp-03-ba3629

What are the names of some of the STRUCT (RECORD Type) fields?

Totals

trafficSource

trafficSource.adwordsClickInfo

device

All of the above

Submit

Execute the query to see how many unique products were viewed

Check my progress 15 / 15

Execute the query to use the UNNEST() on array field

Check my progress 15 / 15

Create a dataset and a table to ingest JSON data

Check my progress 0 / 20

Execute the query to COUNT how many racers were there in total

Check my progress 0 / 10

Execute the query that will list the total

Task 4. Querying datasets that already have arrays

Task 5. Introduction to STRUCTs

Task 6. Practice with STRUCTs and arrays

Task 7. Lab question: STRUCT()

Task 8. Lab question: Unpacking arrays with UNNEST()

Task 9. Filtering within array values

Congratulations!

Working with JSON, Arrays, and Structs in BigQuery

End Lab 01:08:51

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked.

Open Google Console

Username: student-04-e79d9e963f81

Password: 98t8diQ0T98

GCP Project ID: quicklabs-gcp-83-ba3625f

```
data.google.analytics.sample.ga_sessions_20170801;
UNNEST(hits) AS h
WHERE visitId = 15015780398
LIMIT 10
```

We'll cover UNNEST() more in detail later but for now just know that:

- You need to UNNEST() arrays to bring the array elements back into rows
- UNNEST() always follows the table name in your FROM clause (think of it conceptually like a pre-jointed table)

Click Check my progress to verify the objective.

Execute the query to use the UNNEST() on array field

Check my progress

Assessment Completed!

Execute the query to see how many unique products were viewed

Check my progress

15 / 15

Execute the query to use the UNNEST() on array field

Check my progress

15 / 15

Create a dataset and a table to ingest JSON data

Check my progress

0 / 20

Execute the query to COUNT how many racers were there in total

Check my progress

0 / 10

Execute the query that will list the total

Check my progress

Task 4: Querying datasets that already have arrays

Task 5: Introduction to STRUCTs

Task 6: Practice with STRUCTs and arrays

Task 7: Lab question: STRUCT()

Task 8: Lab question: Unpacking arrays with UNNEST()

Task 9: Filtering within array values

Congratulations!

Check complete. Points earned: 15. Message: Assessment Completed!

You may have wondered why the field alias hit.page.pageTitle looks like three fields in one separated by commas - but an ADDAV column now uses the flexibility of an array instead.

10:44 6/15/2023

Working with JSON, Arrays, and Structs in BigQuery

End Lab 01:09:23

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked.

Open Google Console

Username: student-04-e79d9e963f81

Password: 98t8diQ0T98

GCP Project ID: quicklabs-gcp-83-ba3625f

```
SELECT *
FROM `bigquery-public-data`.
data.google.analytics.sample.ga_sessions_20170801
WHERE visitId = 15015780398
```

Run the query.

3. Scroll right in the results until you see the hits.product.v2ProductName field (multiple field aliases are discussed shortly).

You'll note a lot of seemingly blank values in the results as you scroll. Why do you think that is?

BigQuery is still reading the data for those fields

The fields that appear to be missing data are actually at a higher level of granularity than other fields

The dataset is missing data values for those fields

Submit

The amount of fields available in the Google Analytics schema can be overwhelming for analysis.

on array field

Check my progress

0 / 10

Create a dataset and a table to ingest JSON data

Check my progress

0 / 20

Execute the query to COUNT how many racers were there in total

Check my progress

0 / 10

Execute the query that will list the total race time for racers whose names begin with R

Check my progress

0 / 10

Execute the query to see which runner ran fastest lap time

Check my progress

0 / 10

Task 4: Querying datasets that already have arrays

Task 5: Introduction to STRUCTs

Task 6: Practice with STRUCTs and arrays

Task 7: Lab question: STRUCT()

Task 8: Lab question: Unpacking arrays with UNNEST()

Task 9: Filtering within array values

Congratulations!

10:44 6/15/2023

Working with JSON, Arrays, and Structs in BigQuery

Query results - BigQuery - quicklabs-gcp-03-ba3625968f93

BigQuery

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#)

DISMISS UPGRADE

Analysis SQL workspace

Explorer Type to search

Viewing workspace resources. SHOW STARRED ONLY

quicklabs-gcp-03-ba3625968f93 External connections fruit_store fruit_details SHOW MORE

Untitled RUN SAVE SHARE SCHEDULE MORE

```
1. SELECT
2. fullVisitorId,
3. date,
4. ARRAY_AGG(DISTINCT v2ProductName) AS products_viewed,
5. ARRAY_LENGTH(ARRAY_AGG(DISTINCT v2ProductName)) AS distinct_products_viewed,
6. ARRAY_AGG(DISTINCT pageTitle) AS pages_viewed,
7. ARRAY_LENGTH(ARRAY_AGG(DISTINCT pageTitle)) AS distinct_pages_viewed,
8. COUNT(DISTINCT ts) AS sessions,
9. WHERE visitId = 15015780398
10. GROUP BY fullVisitorId, date
11. ORDER BY date
```

Query results

JOE INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	fullVisitorId	date	products_viewed	distinct_products_viewed	pages_viewed
1	570379250208906569	20170731	Google Womens Lightweight ...	2	Google RFID Journal
2	570379250208906569	20170801	Google Snapback Hat Black	1	Google Snapback Hat Black
			Waze Dress Socks	61	Apparel Google Merchandise...
			Plastic Slicing Flashlight		Show by Brand Google Merchandise...
			1 oz Hand Sanitizer		Fun Accessories Google Merchandise...
			Portable Shower Head w/ LED Light		Portable Shower Head w/ LED Light

PERSONAL HISTORY PROJECT HISTORY

REFRESH

10:43 6/15/2023

Working with JSON, Arrays, and Structs in BigQuery

End Lab 01:09:23

Caution: When you are in the console, do not deviate from the lab instructions, doing so may cause your account to be blocked. [Learn more](#)

Open Google Console

Username: student-04-e79d9a963f81

Password: 98t8di0nI98

GCP Project ID: quicklabs-gcp-83-be3625

How many DISTINCT pages were visited by this user on 20170801?

109
 8
 101
 70

Submit

Click **Check my progress** to verify the objective.

Execute the query to see how many unique products were viewed

Assessment Completed!

Recap

You can do some pretty useful things with arrays like

Check complete. Points earned: 15. Message: Assessment Completed!

- finding the number of elements with `ARRAY_LENGTH(<array>)`

Checkpoints

- Create a new dataset and table to store the data **Check my progress** 20 / 20
- Execute the query to see how many unique products were viewed **Check my progress** 15 / 15
- Execute the query to use the `UNNEST()` on array field **Check my progress** 0 / 15
- Create a dataset and a table to ingest JSON data **Check my progress** 0 / 20

task 4: Querying datasets that already have arrays
 Task 5: Introduction to STRUCTs
 Task 6: Practice with STRUCTs and arrays
 Task 7: Lab question: STRUCT()
 Task 8: Lab question: Unpacking arrays with UNNEST()
 Task 9: Filtering within array values
 Congratulations!

Working with JSON, Arrays, and Structs in BigQuery

End Lab 01:09:53

Caution: When you are in the console, do not deviate from the lab instructions, doing so may cause your account to be blocked. [Learn more](#)

Open Google Console

Username: student-04-e79d9a963f81

Password: 98t8di0nI98

GCP Project ID: quicklabs-gcp-83-be3625

```
ARRAY_AGG(v2ProductName) AS products_viewed,
ARRAY_LENGTH(ARRAY_AGG(v2ProductName)) AS num_products_viewed,
ARRAY_AGG(pageTitle) AS pages_viewed,
ARRAY_LENGTH(ARRAY_AGG(pageTitle)) AS num_pages_viewed
FROM `data-to-insights.ecommerce.all_sessions`
WHERE visitId = 1501578398
GROUP BY fullVisitorId, date
ORDER BY date
```

How many pages were visited by this user on 20170801?

109
 70
 8
 101

Submit

6. Next, deduplicate the pages and products so you can see how many unique products were viewed by adding `DISTINCT` to `ARRAY_AGG()`:

```
SELECT
  fullVisitorId,
```

Checkpoints

- Create a new dataset and table to store the data **Check my progress** 20 / 20
- Execute the query to see how many unique products were viewed **Check my progress** 0 / 15
- Execute the query to use the `UNNEST()` on array field **Check my progress** 0 / 15
- Create a dataset and a table to ingest JSON data **Check my progress** 0 / 20

task 4: Querying datasets that already have arrays
 Task 5: Introduction to STRUCTs
 Task 6: Practice with STRUCTs and arrays
 Task 7: Lab question: STRUCT()
 Task 8: Lab question: Unpacking arrays with UNNEST()
 Task 9: Filtering within array values
 Congratulations!

Working with JSON, Arrays, and Structs in BigQuery

End Lab 01:10:23

Caution: When you are in the console, do not deviate from the lab instructions, doing so may cause your account to be blocked. [Learn more](#)

Open Google Console

Username: student-04-e79d9a963f81

Password: 98t8di0nI98

GCP Project ID: quicklabs-gcp-83-be3625

```
ARRAY_AGG(v2ProductName) AS products_viewed,
ARRAY_AGG(pageTitle) AS pages_viewed
FROM `data-to-insights.ecommerce.all_sessions`
WHERE visitId = 1501578398
GROUP BY fullVisitorId, date
ORDER BY date
```

4. Click Run and view the results

How many rows are returned?

63 - one for each day
 70 - one for each day
 2 - one for each day
 100 - one for each day

Submit

5. Next, use the `ARRAY_LENGTH()` function to count the number of pages and products that were viewed:

```
SELECT
```

Checkpoints

- Create a new dataset and table to store the data **Check my progress** 20 / 20
- Execute the query to see how many unique products were viewed **Check my progress** 0 / 15
- Execute the query to use the `UNNEST()` on array field **Check my progress** 0 / 15
- Create a dataset and a table to ingest JSON data **Check my progress** 0 / 20

task 4: Querying datasets that already have arrays
 Task 5: Introduction to STRUCTs
 Task 6: Practice with STRUCTs and arrays
 Task 7: Lab question: STRUCT()
 Task 8: Lab question: Unpacking arrays with UNNEST()
 Task 9: Filtering within array values
 Congratulations!

Google Cloud | BigQuery | Query results - BigQuery - qwiklabs

SANDBOX Set up billing to upgrade to the full BigQuery experience. Learn more

Explorer

Type to search

Viewing workspace resources. SHOW STARRED ONLY

- qwikelabs-gcp-03-ba3625968f93
 - External connections
 - fruit_store
 - fruit_details

Untitled

1 SELECT
2 fullVisitorId,
3 dimensions,
4 vProductname,
5 pageTitle
6 FROM `data-to-insights_ecommerce_all_sessions`
7 WHERE _id = 1001570998
8 ORDER BY date

Query results

Row	fullVisitorId	date	vProductName	pageTitle
1	571037925020908569	20170731	Google Snackpack Hat Black	Google RFID Journal
2	571037925020908569	20170731	Google Womens Lightweight...	Google Snackpack Hat Black
3	571037925020908569	20170801	Android Sticker Sheet Ultra Re...	Office / Google Merchandise St...
4	571037925020908569	20170801	Switch Tone Color Crayon Pen	Fun / Accessories / Google Mer...
5	571037925020908569	20170801	1 oz Hand Sanitizer	Office / Google Merchandise St...

RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

PERSONAL HISTORY PROJECT HISTORY

1042 6/15/2023

Google Cloud | BigQuery | Query results - BigQuery - qwiklabs-gcp-03-ba3625968f93

SANDBOX Set up billing to upgrade to the full BigQuery experience. Learn more

Explorer

Type to search

Viewing workspace resources. SHOW STARRED ONLY

- qwikelabs-gcp-03-ba3625968f93
 - External connections
 - fruit_store
 - fruit_details

Untitled

1 #standardSQL
2 SELECT person, fruit_array, total_cost FROM `data-to-insights_advanced.fruit_store`;

Query results

Row	person	fruit_array	total_cost
1	sally	raspberry blackberry strawberry cherry	10.99
2	frederick	orange	5.55

"fruit_details" created GOT TO TABLE

RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

PERSONAL HISTORY PROJECT HISTORY

1042 6/15/2023

Working with JSON, Arrays, and Structs in BigQuery

End Lab 01:11:25

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked.

Open Google Console

Username: student-84-e7ad9a963f81

Password: 98t6bd1QnT9B

GCP Project ID: qwikelabs-gcp-03-ba3625968f93

4. Call the new table fruit_details.

5. Check the checkbox of Schema (Auto detect).

6. Click Create table.

In the schema, note that fruit_array is marked as REPEATED which means it's an array.

Recap

- BigQuery natively supports arrays
- Array values must share a data type
- Arrays are called REPEATED fields in BigQuery

Click Check my progress to verify the objective.

Create a new dataset and table to store our data

Check my progress

Assessment Completed!

Checkpoints

- Create a new dataset and table to store the data
- Execute the query to see how many unique products were viewed
- Execute the query to use the UNNEST() on array field
- Create a dataset and a table to ingest JSON data

Check my progress

task 4. Querying datasets that already have arrays

Task 5. Introduction to STRUCTs

Task 6. Practice with STRUCTs and arrays

Task 7. Lab question: STRUCT()

Task 8. Lab question: Unpacking arrays with UNNEST()

Task 9. Filtering within array values

Congratulations!

Check complete. Points earned: 20. Message: Assessment Completed!

1042 6/15/2023

Working with JSON, Arrays, and Structs in BigQuery

End Lab 01:13:13

Question: When you run the command do not deviate from the lab instructions. Doing so may cause your session to be blocked.

Open Google Console

Username: student-84-e79fe9c1ff8

Password: 981b600e43b

GCP Project ID: quicklabs-gcp-03-ba3625f

standardSQL

```
SELECT
  ["raspberry", "blackberry", "strawberry", "cherry", 1234567] AS
  fruit_array;
```

You should get an error that looks like the following:

Error: Array elements of types {INT64, STRING} do not have a common supertype at [1:1]

Why did you get this error?

- Data in an array cannot exceed 4 elements
- Data in an array [] must all be the same type
- Data in an array must only be strings

Submit

Arrays can only share one data type (all strings, all numbers).

4. Here's the final table to query against:

standardSQL

```
SELECT person, fruit_array, total_cost FROM `data-to-insights.advanced.fruit_store`;
```

5. Click Run

10:40 6/15/2023

Working with JSON, Arrays, and Structs in BigQuery

End Lab 01:13:43

Question: When you run the command do not deviate from the lab instructions. Doing so may cause your session to be blocked.

Open Google Console

Username: student-84-e79fe9c1ff8

Password: 981b600e43b

GCP Project ID: quicklabs-gcp-03-ba3625f

In traditional relational database SQL, you would look at the repetition of names and immediately think of splitting the above table into two separate tables: Fruit Items and People. That process is called **normalization** (going from one table to many). This is a common approach for transactional databases like MySQL...

For data warehousing, data analysts often go the reverse direction (denormalization) and bring many separate tables into one large reporting table.

What are some potential issues if you stored all your data in one giant table?

- The table row size could be too large for traditional reporting databases
- Any changes to a value (like customer email) could impact many other rows (like all their orders)
- Data at differing levels of granularity could lead to reporting issues because less granular fields would be repeated
- All of the above

Submit

Now, you're going to learn a different approach that stores data at different levels of granularity all in one table using repeated fields:

Row	Fruit (array)	Person
1	raspberry	saif
	blackberry	
	strawberry	

10:39 6/15/2023

Google Cloud Skills Boost

Google Cloud

Paths Explore Profile Subscriptions

Continue learning path

Creating a Data Warehouse Through Joins and Unions

This lab focuses on how to create new reporting tables using SQL JOINS and UNIONS.

★★★★★ 1 hour Introductory 1 Credit

Lab Creating Date-Partitioned Tables in BigQuery

This lab focuses on how to query partitioned datasets and how to create your own dataset partitions to improve query performance, which reduces cost.

★★★★★ 1 hour Introductory 1 Credit

Lab Troubleshooting and Solving Data Join Pitfalls

This lab focuses on how to reverse-engineer the relationships between data tables and the pitfalls to avoid when joining them together.

★★★★★ 1 hour Introductory 1 Credit

Lab Working with JSON, Arrays, and Structs in BigQuery

09:38 6/16/2023