

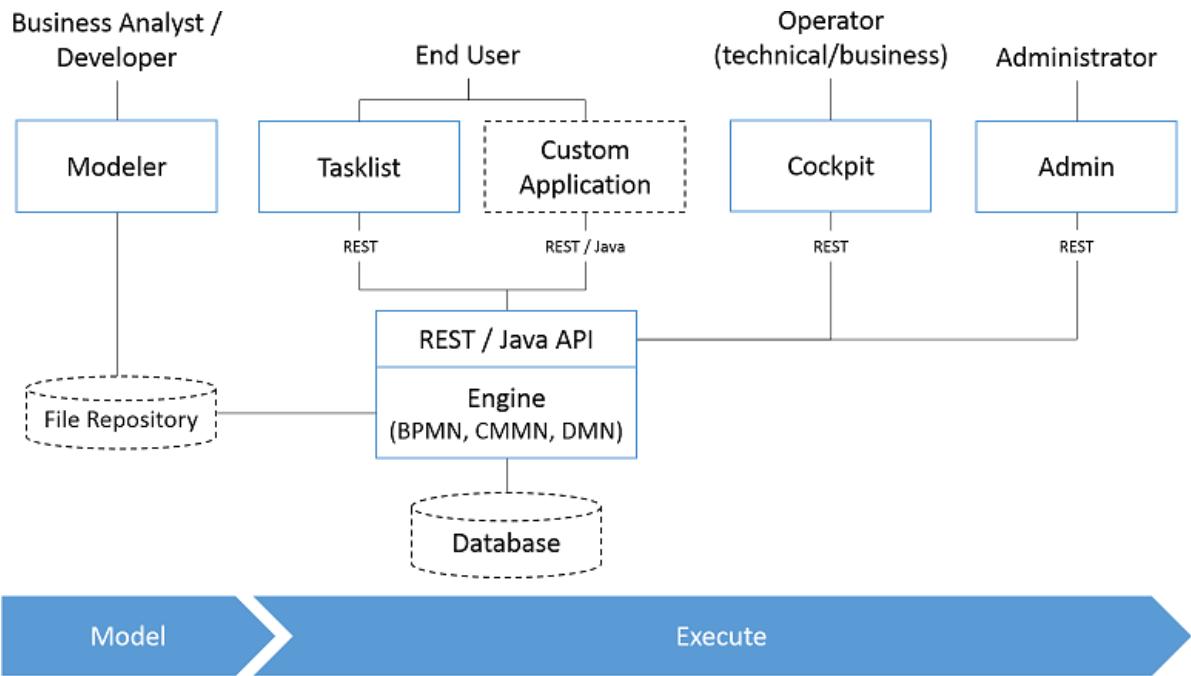
Internship Report

Aishvarya Akshaya Vishva Raman
(Firm Risk Management Intern)

Table of Contents

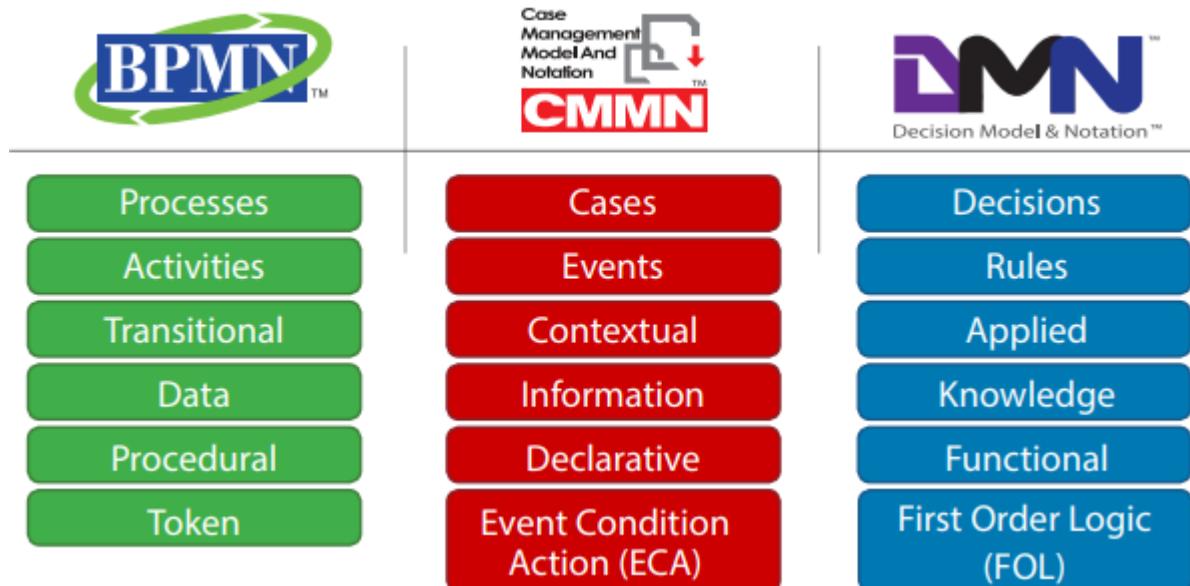
| | |
|-----------------------|----|
| Camunda..... | 3 |
| SingleStore SQL | 30 |
| Power Bi..... | 58 |

Camunda



- 1) **Process Engine:** The process engine is a Java library responsible for executing BPMN 2.0 processes, CMMN 1.1 cases and DMN 1.3 decisions. It has a lightweight POJO core and uses a relational database for persistence. ORM mapping is provided by the MyBatis mapping framework.
- 2) **Business Process Modelling Framework:** BPMN is a precise, complete and graphical notation for documenting well-defined business processes. It resolves many ambiguities found in textual process specifications by assigning activities to specific actors. Analysing the resulting models can be used to drive process improvement initiatives, regardless of whether processes are automated or manual. Because the graphical model is readily understandable by non-technical people, it serves as a bridge that allows collaboration between business stakeholders and IT personnel. OMG's BPMN 2.0.1 specification has been published as International Standard ISO/IEC 19510:2013.
- 3) **Case Management Model and Notation:** CMMN is a graphical notation used for capturing work methods that are based on the handling of “cases” requiring various activities that may be performed in an unpredictable order in response to evolving situations. Using an event-centred approach and the concept of a case file, CMMN expands the boundaries of what can be modelled with BPMN, including less structured work efforts and those driven by knowledge workers. Using a combination of BPMN and CMMN allows users to cover a much broader spectrum of work methods.
- 4) **Decision Model and Notation:** DMN is a notation for the precise specification of business decisions and rules. DMN is readily readable by the different types of people involved in decision management. These include business people who specify

the rules but also monitor their application; business analysts who transform user input into detailed decision models; and software developers who implement them in enterprise systems. DMN is designed to work alongside BPMN or CMMN providing a mechanism to model decision making within both process models and case models.



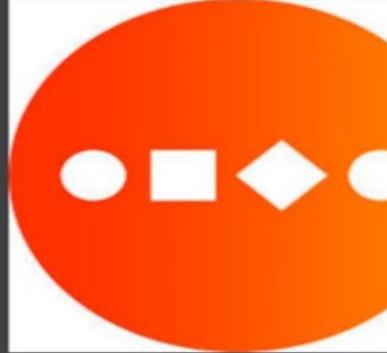
- 5) **REST API** The REST API allows you to use the process engine from a remote application or a JavaScript application. (Note: The documentation of the REST API is factored out into own documents.)
- 6) **Camunda Tasklist:** A web application for human workflow management and user tasks that allows process participants to inspect their workflow tasks and navigate to task forms in order to work on the tasks and provide data input.
- 7) **Camunda Cockpit:** A web application for process monitoring and operations that allows you to search for process instances, inspect their state and repair broken instances.
- 8) **Camunda Admin:** A web application that allows you to manage users, groups and authorizations.

Problem Statement: Steps for Deployment of an Application using Camunda

- 1) Camunda Platform Run is a pre-packaged distro of the Camunda Platform, including the Camunda webapps (Cockpit, Tasklist, Admin), the REST API and a bundled version of Swagger UI. Swagger UI is a web-GUI that allows you to explore the REST API endpoints of Camunda Platform Run. The idea behind Run is to provide a full Camunda Platform distro with a simple but powerful configuration mechanism that can be operated by everyone, regardless of their knowledge about Java or application server configuration.

Running Camunda BPM Run

- Self Contained Distribution
- Fast Startup
- Easy Configuration
- Easy to Customize
- No need to know any Java.



6

Usually, we prefer WildFly server which we have used in the further steps of deployment of Camunda BPM using Java code.

How to Run Camunda



OR



Spring Boot



- 2) Camunda Platform Run ships with two different configuration files which are both located in the configuration folder.
 - The default.yml configuration only contains necessary configuration like the H2 database, a demo user and CORS for REST calls from a client application.
 - The production.yml configuration is intended to provide the recommended properties according to the Security Instructions. When using Camunda Platform Run in a production environment, make sure to base your custom configuration on this one and carefully read through the security instructions.
- 3) Install Camunda- open-source community edition and Camunda open-source modeler to create the process diagrams and initiate the further steps of deployment:
 - i) First download the Open-Source Community edition, this will be useful for starting the localhost and deploying process diagrams created. Unzip the files.

CAMUNDA

Products Why Camunda? Solutions Services Developers About Contact Us Download Try Free 

Download Camunda

Open Source Modeler Open Source Community Edition 30-Day Trial Enterprise Edition Download for Enterprise Customers

Version: 7.15.0 Release Date: April 13, 2021

Distribution Type:  Camunda Platform Run

Contains:  Engines  Tasklist  Cockpit Basic

Download: [RUN ZIP](#) [RUN TAR](#)

ii) The configuration folder is the most important as it contains sql files and below is an example of a sql file. Camunda Platform run will be started using “start.bat” (Windows) or “start.sh” (Linux).

This PC > Downloads > camunda-bpm-run-7.15.0

| Name | Date modified | Type | Size |
|-------------------------|------------------|--------------------|--------|
| configuration | 06-04-2021 11:54 | File folder | |
| internal | 06-04-2021 11:54 | File folder | |
| LICENSE | 06-04-2021 11:54 | File | 11 KB |
| LICENSE_BOOK-7.15.0.txt | 06-04-2021 11:45 | Text Document | 311 KB |
| NOTICE | 06-04-2021 11:54 | File | 1 KB |
| README.txt | 06-04-2021 11:54 | Text Document | 2 KB |
| start.bat | 06-04-2021 11:54 | Windows Batch File | 3 KB |
| start.sh | 06-04-2021 11:54 | Shell Script | 3 KB |

```

Select C:\WINDOWS\system32\cmd.exe
Setting JAVA property to "C:\Program Files\Java\jdk-11.0.6\bin\java"
REST API enabled
WebApps enabled
Swagger UI enabled
classpath: C:\Users\abc\Downloads\camunda-bpm-run-7.15.0\internal\webapps,C:\Users\abc\Downloads\camunda-bpm-run-7.15.0\internal\rest,C:\Users\abc\Downloads\camunda-bpm-run-7.15.0\internal\swaggerui,C:\Users\abc\Downloads\camunda-bpm-run-7.15.0\configuration\userlib,C:\Users\abc\Downloads\camunda-bpm-run-7.15.0\configuration\keystore

[Camunda BPM Platform]

Spring-Boot: (v2.4.3)
Camunda Platform: (v7.15.0)

```

```

C:\WINDOWS\system32\cmd.exe
dminUserConfiguration[adminUser=AdminUserProperty[id=demo, firstName=Demo, lastName=Demo, email=demo@localhost, password='*****']], failedJobConfiguration, eventPublisherPlugin, SpringBootSpinProcessEnginePlugin] activated on process engine 'default'
2021-06-16 18:10:42.106 INFO 756 --- [           main] org.camunda.bpm.spring.boot          : STARTER-SB021 Auto-Deploying resources: []
2021-06-16 18:10:42.134 INFO 756 --- [           main] o.c.b.s.b.s.event.EventPublisherPlugin : EVENTING-001: Initialized Camunda Spring Boot Eventing Engine Plugin.
2021-06-16 18:10:42.134 INFO 756 --- [           main] o.c.b.s.b.s.event.EventPublisherPlugin : EVENTING-003: Task events will be published as Spring Events.
2021-06-16 18:10:42.136 INFO 756 --- [           main] o.c.b.s.b.s.event.EventPublisherPlugin : EVENTING-005: Execution events will be published as Spring Events.
2021-06-16 18:10:42.163 INFO 756 --- [           main] o.c.b.s.b.s.event.EventPublisherPlugin : EVENTING-007: History events will be published as Spring events.
2021-06-16 18:10:42.233 INFO 756 --- [           main] org.camunda.spin                    : SPIN-01010 Discovered Spin data format provider: org.camunda.spin.impl.json.jackson.format.JacksonJsonDataFormatProvider[name = application/json]
2021-06-16 18:10:43.158 INFO 756 --- [           main] org.camunda.spin                    : SPIN-01010 Discovered Spin data format provider: org.camunda.spin.impl.xml.dom.format.DomXmlDataFormatProvider[name = application/xml]
2021-06-16 18:10:43.257 INFO 756 --- [           main] org.camunda.spin                    : SPIN-01009 Discovered Spin data format provider: org.camunda.spin.impl.xml.dom.format.DomXmlDataFormat[name = application/xml]
2021-06-16 18:10:43.258 INFO 756 --- [           main] org.camunda.spin                    : SPIN-01009 Discovered Spin data format provider: org.camunda.spin.impl.json.jackson.format.JacksonJsonDataFormat[name = application/json]
2021-06-16 18:10:44.349 INFO 756 --- [           main] org.camunda.bpm.dmn.feel.scalacalculator          : FEEL/SCALA-01001 Spin value mapper detected
2021-06-16 18:10:44.759 INFO 756 --- [           main] org.camunda.feel.FeelEngine          : Engine created. [value-mapper: CompositeValueMapper(List(org.camunda.feel.impl.JavaValueMapper@5754de72, org.camunda.spin.plugin.impl.feel.integration.SpinValueMapper@31ee95f4)), function-provider: org.camunda.bpm.dmn.feel.impl.scala.function.CustomFunctionTransformer@78e89bfe, clock: SystemClock, configuration: Configuration(false)]
2021-06-16 18:10:45.159 INFO 756 --- [           main] com.zaxxer.hikari.HikariDataSource          : HikariPool-1 - Starting...
2021-06-16 18:10:46.266 INFO 756 --- [           main] com.zaxxer.hikari.pool.PoolBase          : HikariPool-1 - Driver does not support get/set network timeout for connections. (Receiver class org.h2.jdbc.JdbcConnection does not define or inherit an implementation of the resolved method abstract getNetworkTimeout()I of interface java.sql.Connection.)
2021-06-16 18:10:46.307 INFO 756 --- [           main] com.zaxxer.hikari.HikariDataSource          : HikariPool-1 - Start completed.

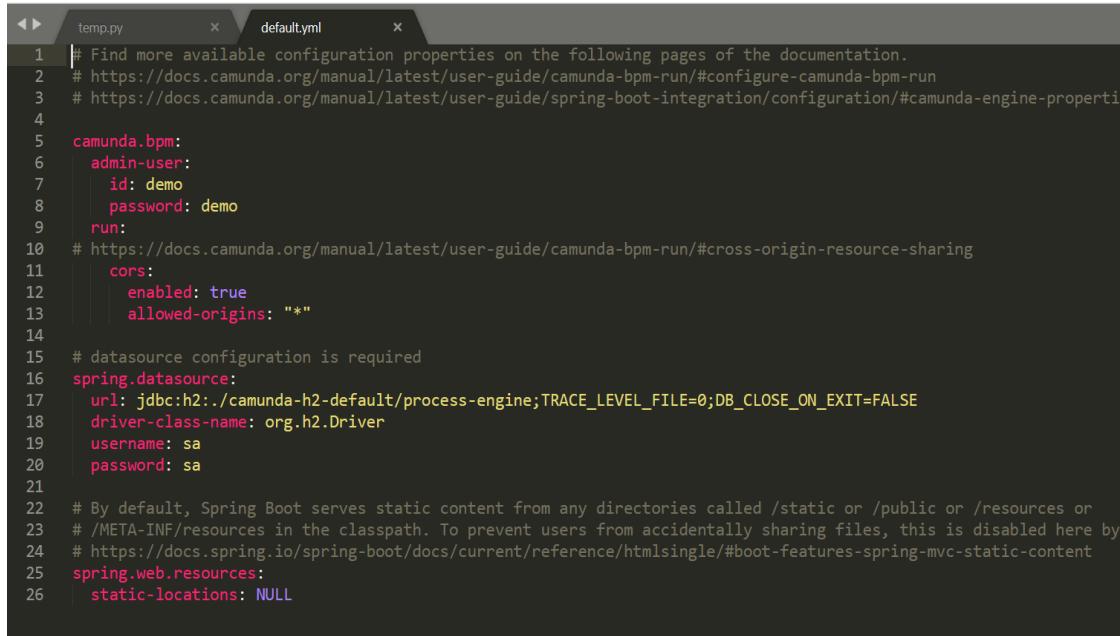
```

- iii) Go to localhost:8080 once “start.bat” starts running. In “default.yml” file we can see that the id is “demo” and the password is “demo”. Type that in our login page.

The screenshot shows a web browser window with the following details:

- Address Bar:** localhost:8080/camunda/app/welcome/default/#!/login
- Page Title:** CAMUNDA
- Content:**

Welcome



```
temp.py      default.yml
1 # Find more available configuration properties on the following pages of the documentation.
2 # https://docs.camunda.org/manual/latest/user-guide/camunda-bpm-run/#configure-camunda-bpm-run
3 # https://docs.camunda.org/manual/latest/user-guide/spring-boot-integration/configuration/#camunda-engine-properties
4
5 camunda.bpm:
6   admin-user:
7     id: demo
8     password: demo
9   run:
10  # https://docs.camunda.org/manual/latest/user-guide/camunda-bpm-run/#cross-origin-resource-sharing
11  cors:
12    enabled: true
13    allowed-origins: "*"
14
15 # datasource configuration is required
16 spring.datasource:
17   url: jdbc:h2:./camunda-h2-default/process-engine;TRACE_LEVEL_FILE=0;DB_CLOSE_ON_EXIT=FALSE
18   driver-class-name: org.h2.Driver
19   username: sa
20   password: sa
21
22 # By default, Spring Boot serves static content from any directories called /static or /public or /resources or
23 # /META-INF/resources in the classpath. To prevent users from accidentally sharing files, this is disabled here by
24 # https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#boot-features-spring-mvc-static-content
25 spring.web.resources:
26   static-locations: NULL
```

- iv) We can see the Cockpit and the Tasklist once we are logged-in.

Camunda Welcome

Applications

- Cockpit
- Tasklist
- Admin

Profile

Demo Demo
demo@localhost

Edit profile Change password

Groups

- camunda BPM Administrators

Links

Documentation
Camunda webapps user documentation

Camunda Cockpit

Processes Decisions Human Tasks More ▾

Right Now

| Running Process Instances | Open Incidents | Open Human Tasks |
|---------------------------|----------------|------------------|
| 0 | 0 | 0 |

Deployed

| Process Definitions | Decision Definitions | Case Definitions | Deployments |
|---------------------|----------------------|------------------|-------------|
| 0 | 0 | 0 | 0 |

- v) Download the Open-Source Modeler to create process diagrams. Once downloaded open the CamundaModeler.exe file.

| | | | |
|---------------------|-------------------------------|---------------------------------|-----------------------------------|
| Open Source Modeler | Open Source Community Edition | 30-Day Trial Enterprise Edition | Download for Enterprise Customers |
|---------------------|-------------------------------|---------------------------------|-----------------------------------|

Version: 4.8.1

Release Date: May 12, 2021

Supports:



BPMN



DMN

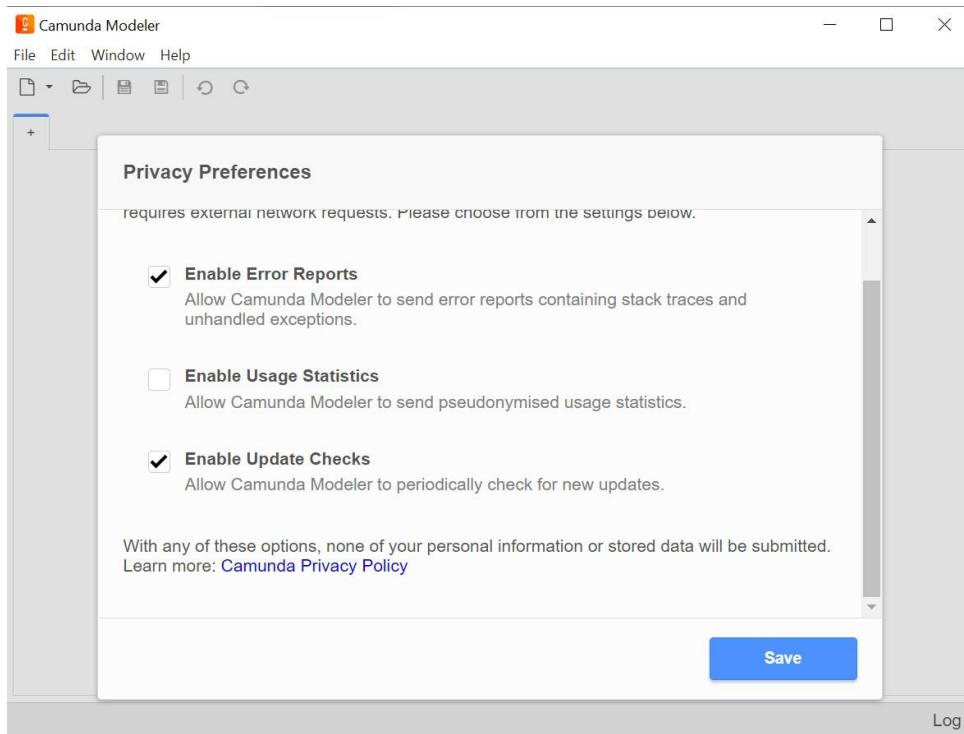
Download:

Mac OS .zip ▾

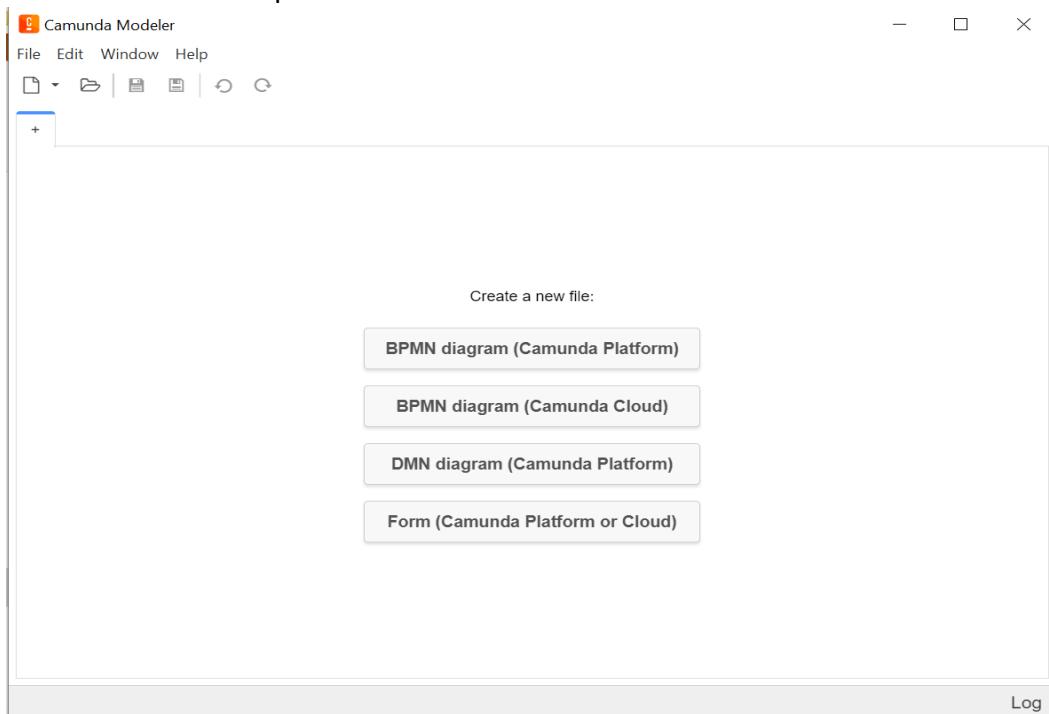
Mac OS .dmg ▾

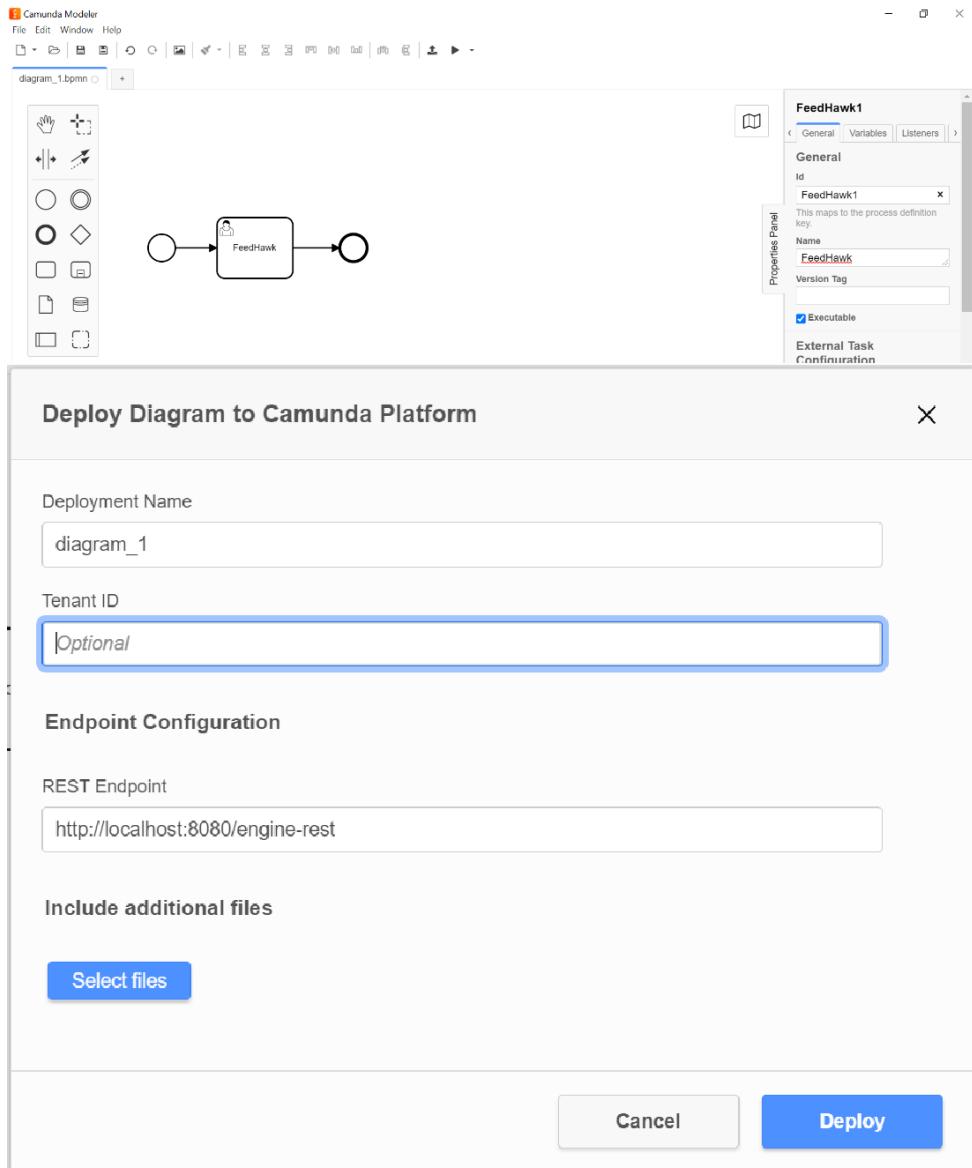
Windows 32bit ▾

Windows 64bit ▾

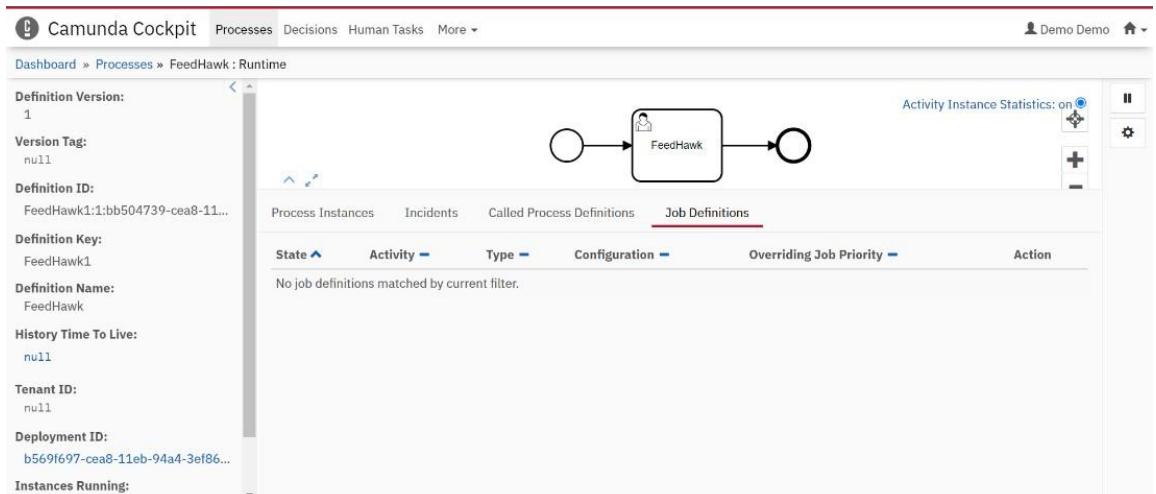


- vi) Create the BPMN diagram, give a unique-id and name to it and deploy it by giving the link for REST endpoint.



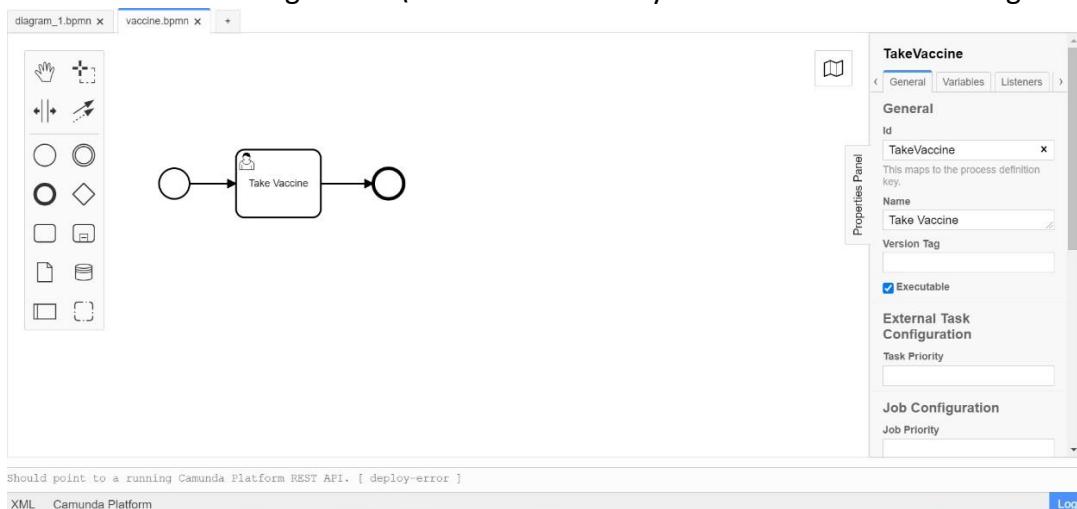


- vii) Refresh the Cockpit page and we can see our process diagram has been deployed. Further go to the Tasklist page to create an instance of the process.



The screenshot shows the Camunda Cockpit interface. At the top, there's a modal window titled "Start process" with a "Business Key" input field and a "Start" button. Below it, the main dashboard shows "2 process definitions deployed". A table lists two definitions: "FeedHawk" (State: green checkmark, Incidents: 0, Running Instances: 0) and another "FeedHawk" (State: green checkmark, Incidents: 0, Running Instances: 1). On the right, the "Process Instances" tab is selected, showing a single instance named "dd73ca7d-cea9-11eb-94a4-3ef862082519" with a start time of "2021-06-16T19:20:52". The process diagram on the right shows a task named "Feed your Pet Hawk" with an ID of 1.

- viii) One can also directly deploy the process diagram by saving the “. bpmn” file in the Configuration\Resources directory and start the “start.bat” again.



File Explorer:

| Name | Date modified | Type | Size |
|--------------|------------------|-----------|------|
| vaccine.bpmn | 16-06-2021 19:34 | BPMN File | 3 KB |

Command Prompt (C:\Windows\System32\cmd.exe):

```
Setting JAVA property to "C:\Program Files\Java\jdk-11.0.6\bin\java"
REST API enabled
WebApps enabled
Swagger UI enabled
classpath: C:\Users\abc\Downloads\camunda-bpm-run-7.15.0\internal\webapps,C:\Users\abc\Downloads\camunda-bpm-run-7.15.0\internal\rest,C:\Users\abc\Downloads\camunda-bpm-run-7.15.0\internal\swaggerui,C:\Users\abc\Downloads\camunda-bpm-run-7.15.0\configuration\userlib,C:\Users\abc\Downloads\camunda-bpm-run-7.15.0\configuration\keystore
```

Browser (Camunda Cockpit):

Processes

| State | Incidents | Running Instances | Name | Tenant ID |
|-------|-----------|-------------------|--------------|-----------|
| ✓ | 0 | 0 | FeedHawk | |
| ✓ | 0 | 1 | FeedHawk | |
| ✓ | 0 | 0 | Take Vaccine | |

localhost:8080/engine-rest/engine

Apps Instructors and Stu... Multithreaded Sock

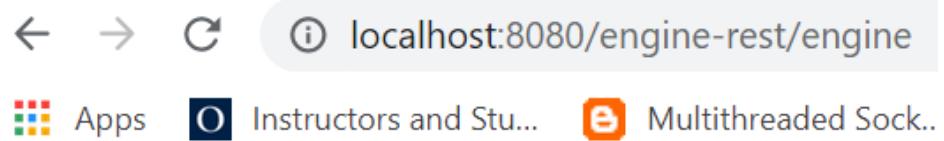
[{"name": "default"}]

ix) One can add the variable process-engine-name and set it to some value. Hence, we will be able to see the name set in the below link.

```

1 # Find more available configuration properties on the following pages of the documentation.
2 # https://docs.camunda.org/manual/latest/user-guide/camunda-bpm-run/#configure-camunda-bpm-run
3 # https://docs.camunda.org/manual/latest/user-guide/spring-boot-integration/configuration/#camunda-engine-properties
4
5 camunda.bpm:
6   admin-user:
7     id: demo
8     password: demo
9   run:
10  # https://docs.camunda.org/manual/latest/user-guide/camunda-bpm-run/#cross-origin-resource-sharing
11    cors:
12      enabled: true
13      allowed-origins: "*"
14    process-engine-name: "Aishu"
15
16  # datasource configuration is required
17 spring.datasource:
18   url: jdbc:h2:./camunda-h2-default/process-engine;TRACE_LEVEL_FILE=0;DB_CLOSE_ON_EXIT=FALSE
19   driver-class-name: org.h2.Driver
20   username: sa
21   password: sa
22
23  # By default, Spring Boot serves static content from any directories called /static or /public or /resources or
24  # /META-INF/resources in the classpath. To prevent users from accidentally sharing files, this is disabled here by
25  # https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#boot-features-spring-mvc-static-content
26  spring.web.resources:
27    static-locations: NULL

```



Hence, we saw above how to run Camunda Run BPM and how to deploy our process model diagram. We will further see the steps of how a java project can be deployed using Camunda Enterprise platform and WildFly server.

- 4) Install WildFly server for Camunda Enterprise edition. We will be using IntelliJ IDE.
- i) For the java code and establishing a connection with Camunda BPM we will be require to create an archetype initially while creating a new Java-Maven project.

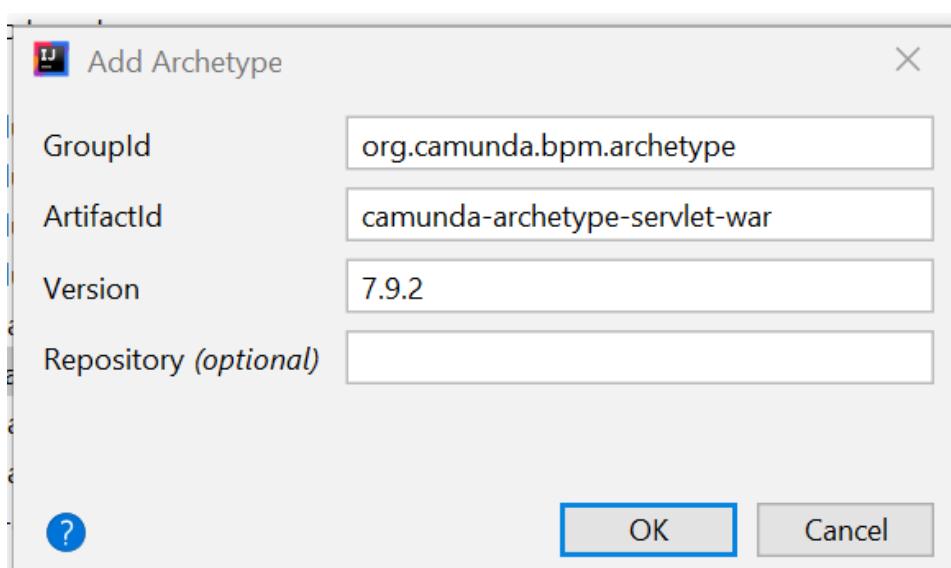
The screenshot shows the Sonatype Nexus Repository Manager interface. The top navigation bar includes the logo, the text "Sonatype Nexus Repository Manager OSS 3.30.1-01", and a search bar labeled "Search components". The left sidebar has sections for "Browse", "Welcome", and "Search". The main content area is titled "Browse / camunda-bpm". It shows an "HTML View" of the archetype catalog structure, which includes folders for "com", "io", and "org", and files for "archetype-catalog.xml", "archetype-catalog.xml.md5", "archetype-catalog.xml.sha1", "archetype-catalog.xml.sha256", and "archetype-catalog.xml.sha512". Below the tree view is a large text block containing the XML content of the archetype catalog file. At the bottom of the page, there is a message stating "This XML file does not appear to have any style information associated with it. The document tree is shown below." The XML content is as follows:

```

<archetype-catalog xmlns="http://maven.apache.org/plugins/maven-archetype-plugin/archetype-catalog/1.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/plugins/maven-archetype-plugin/archetype-catalog/1.0.0 http://maven.apache.org/xsd/archetype-catalog-1.0.0.xsd">
  <archetype>
    <groupId>org.camunda.bpm.archetype</groupId>
    <artifactId>camunda-archetype-cockpit-plugin</artifactId>
    <version>0.1-test</version>
    <description>Plugin for Camunda Cockpit, contains REST-Backend, MyBatis database query, HTML and JavaScript frontend, Ant build script for one-click deployment</description>
  </archetype>
  <archetype>
    <groupId>org.camunda.bpm.archetype</groupId>
    <artifactId>camunda-archetype-cockpit-plugin</artifactId>
    <version>1.0.0</version>
    <description>Plugin for Camunda Cockpit, contains REST-Backend, MyBatis database query, HTML and JavaScript frontend, Ant build script for one-click deployment</description>
  </archetype>
  <archetype>
    <groupId>org.camunda.bpm.archetype</groupId>
    <artifactId>camunda-archetype-cockpit-plugin</artifactId>
    <version>1.1.2</version>
    <description>Plugin for Camunda Cockpit, contains REST-Backend, MyBatis database query, HTML and JavaScript frontend, Ant build script for one-click deployment</description>
  </archetype>
  <archetype>
    <groupId>org.camunda.bpm.archetype</groupId>
    <artifactId>camunda-archetype-cockpit-plugin</artifactId>
    <version>7.1.3</version>
    <description>Plugin for Camunda Cockpit, contains REST-Backend, MyBatis database query, HTML and JavaScript frontend, Ant build script for one-click deployment</description>
  </archetype>
  <archetype>
    <groupId>org.camunda.bpm.archetype</groupId>
    <artifactId>camunda-archetype-cockpit-plugin</artifactId>
  </archetype>

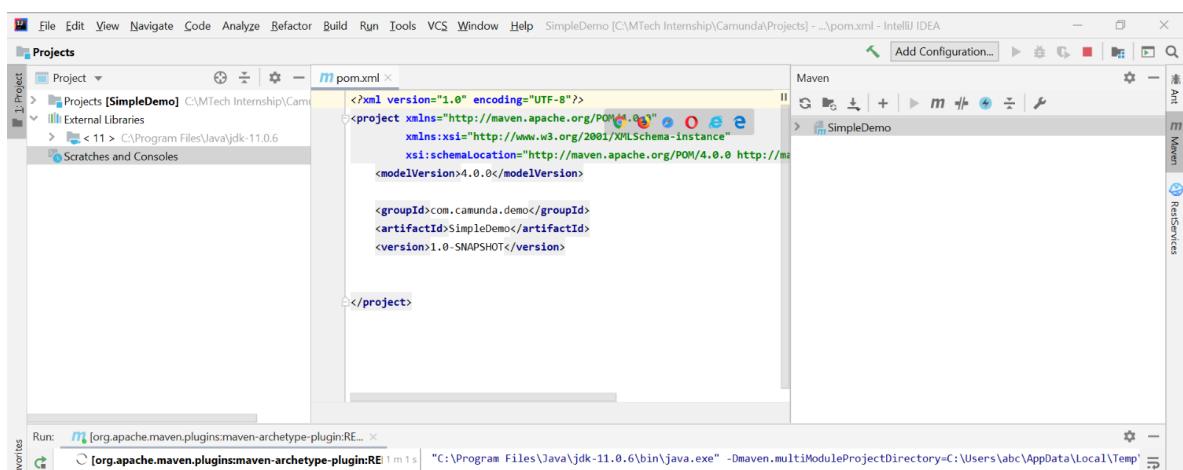
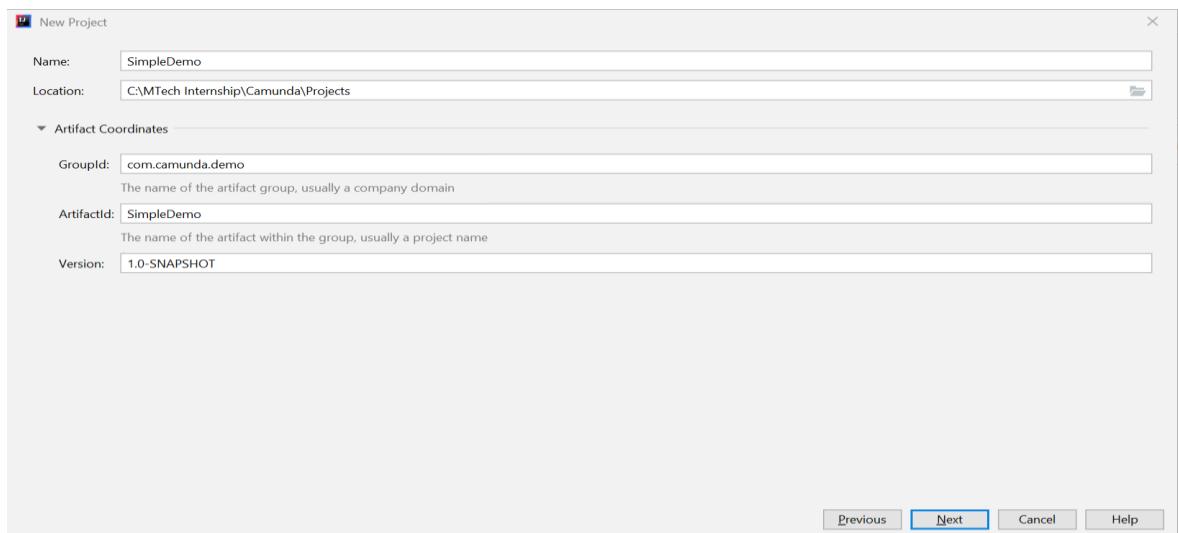
```

II) Enter the group-id, artifact-id and the version for the plugin.

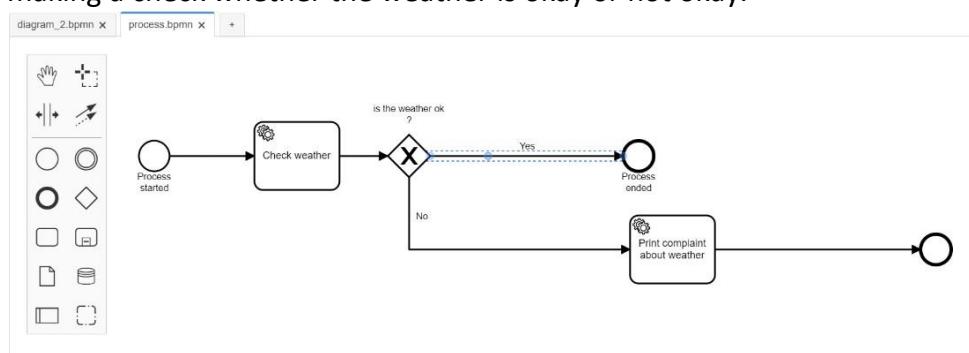


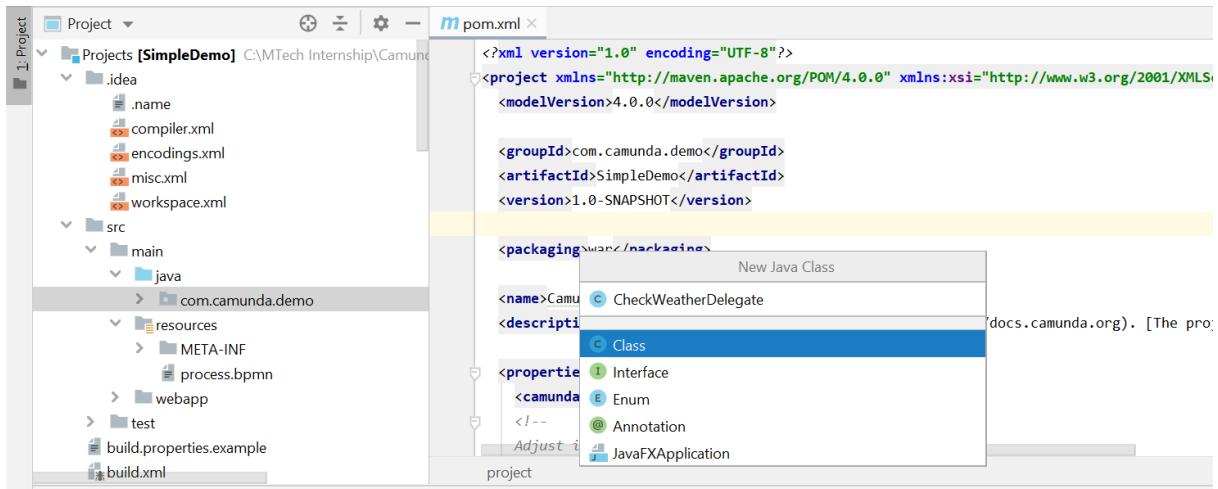
- III) The name of the project considered here is “SimpleDemo”. The co-ordinates should

be named as per convenience.



IV) Open the process. bnpn file and make the changes wrt the project. In our case we are making a check whether the weather is okay or not okay.





V) Create two java files and based on those variables of “weatherOk” and the class path of the files rename the same in “proces.bnppn” diagram as well in the properties section.

CheckWeatherDelegate.java - Notepad

File Edit Format View Help

```
package com.camunda.demo;
```

```
import org.camunda.bpm.engine.delegate.DelegateExecution;
import org.camunda.bpm.engine.delegate.JavaDelegate;
```

```
import java.util.Random;
```

```
public class CheckWeatherDelegate implements JavaDelegate {
    @Override
    public void execute(DelegateExecution execution) throws Exception{
```

```
        Random rando = new Random();
        execution.setVariable("name", "Aishvarya");
        execution.setVariable("weatherOk", rando.nextBoolean());
```

```
}
```

```
}
```

VI) Before beginning to open “start-camunda.bat” in Enterprise Camunda BPM Edition make sure to install JDK and add the path “**C:\Windows\System32**” to the PATH value of Environmental variables.

The screenshot shows the Camunda BPM Studio interface. On the left is a process diagram with a start event, a transition to a boundary event labeled "Print complaint about weather", and another transition to an end event labeled "Process ended". The "Properties Panel" on the right shows the activity's configuration:

- General**: Id = Activity_1s8p60b, Name = Print complaint about weather.
- Details**: Implementation = Java Class, Java Class = com.camunda.demo.LoggerDelegate.
- Asynchronous Continuations**: Both "Asynchronous Before" and "Asynchronous After" checkboxes are unchecked.
- Documentation**: Element Documentation is present.

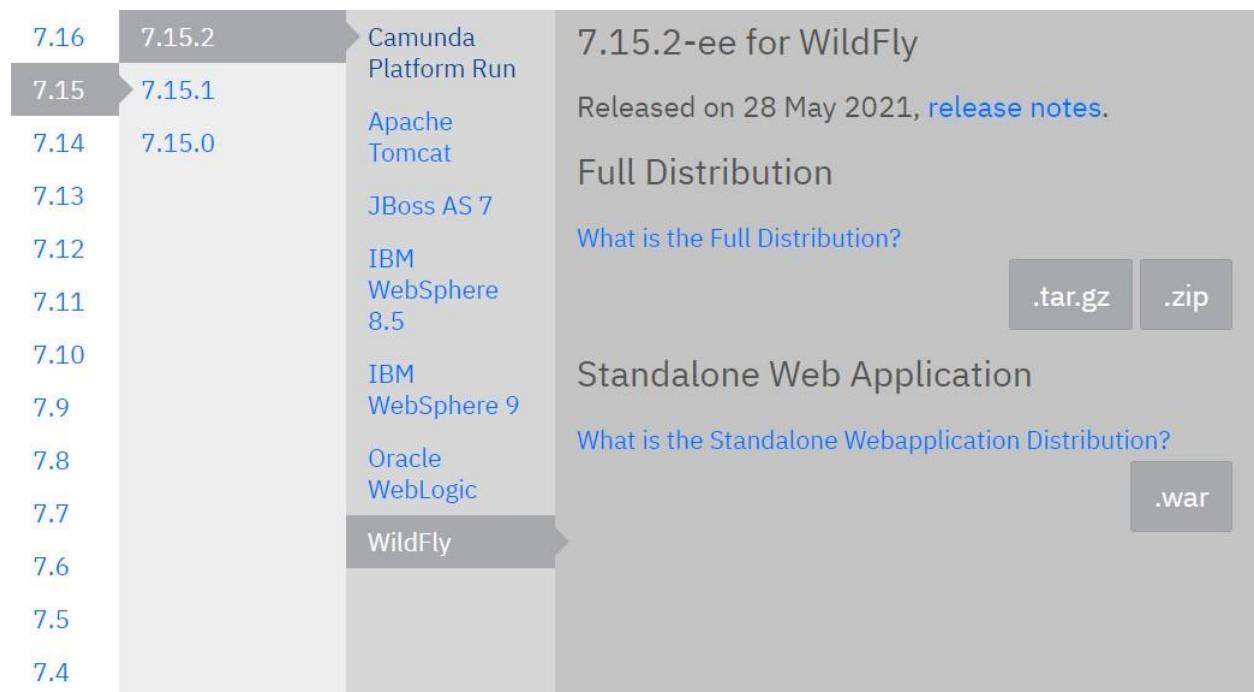
The bottom part of the interface is a code editor with tabs for pom.xml, CheckWeatherDelegate.java, and LoggerDelegate.java. The code editor displays the following Java code:

```

private final Logger LOGGER = LoggerFactory.getLogger(LoggerDelegate.class.getName());

public void execute(DelegateExecution execution) throws Exception {
    LOGGER.info(" ... " + execution);
}
}
}
}

```



VII) Once you start “start-camunda.bat” it will directly take you to
“<https://localhost:8080/camunda-welcome/index.html>”

▶ This PC ▶ Downloads ▶ camunda-bpm-ee-wildfly-7.15.2-ee

| <input type="checkbox"/> Name | Date modified | Type | Size |
|---|------------------|--------------------|--------|
| <input type="checkbox"/> lib | 26-05-2021 16:09 | File folder | |
| <input type="checkbox"/> modules | 26-05-2021 16:18 | File folder | |
| <input type="checkbox"/> server | 26-05-2021 16:23 | File folder | |
| <input type="checkbox"/> sql | 26-05-2021 16:23 | File folder | |
| <input type="checkbox"/> LICENSE | 26-05-2021 16:23 | File | 1 KB |
| <input type="checkbox"/> LICENSE_BOOK-7.15.2-ee.txt | 26-05-2021 16:13 | Text Document | 311 KB |
| <input type="checkbox"/> NOTICE | 26-05-2021 16:23 | File | 1 KB |
| <input type="checkbox"/> README.txt | 26-05-2021 16:23 | Text Document | 2 KB |
| <input type="checkbox"/> start-camunda.bat | 26-05-2021 16:10 | Windows Batch File | 1 KB |
| <input type="checkbox"/> start-camunda.sh | 26-05-2021 16:10 | Shell Script | 1 KB |

VIII) Enter the user name, password and the license-key.

CAMUNDA

Cockpit

Log in

Date and Time displayed in local timezone: *Asia/Calcutta*

Powered by [Camunda Platform](#) / v7.15.2-ee

Camunda Admin Users Groups Tenants Authorizations System **Demo Demo**

Dashboard » System Settings

- General
- Telemetry Settings
- Execution Metrics
- License Key**
- Operation Log

License Key

No license key has been provided.

Enter your License Key:

```
CjrsHh/ksz7j2JBRzinkstJ3R94fvg77BjQAFvm9UvAQB6Hew/jZV/+H
IzsjikFrws2luV2ejc1j5x4j69ix3vofG8Sizn158XGOpv8NS9k6gfrmD
aiM588nxgX15PB60m+Ogd5Q+8AQgVmNc7i37N4L2Wc5vh9Hlus4JgtvBt
PTXc501lwCRKw+xxn07ZqSGyMUzk8zqochpIafv/ji77Ia68702gPY1b
kitR4QZ/wHQAYYQJdwelci21AmuX1bcdTGbh50aELt5n0nB326+x/FBQ
YcCV37ql0nxymslQ0kN9hz007zmKHhiyqtK8BxrmVdNLT2XWCHhk0tn8
dI6006oov710gJ25eu0513WJy9eNS9v+SrIBeyHaB8BE1Ha@CVo5gX7vc
```

Save Key Do not have a license key?

IX) Start the Camunda-Cockpit and Camunda-Tasklist, we can see there will be already 2 processes pre-deployed.

Camunda Cockpit Processes Decisions Cases Human Tasks More **Demo Demo**

Right Now

| Running Process Instances | Open Incidents | Open Human Tasks |
|---------------------------|----------------|------------------|
| 8 | 0 | 6 |

Metrics

| Executed Activity Instances | | Evaluated Decision Instances | | Executed Jobs | |
|-----------------------------|-------|------------------------------|-------|---------------|--------|
| Evaluated | Ended | Evaluated | Ended | Successful | Failed |
| 19 ↑ | 15 ↑ | 42 ↑ | | 7 ↑ | 0 → |

Deployed

| Process Definitions | Decision Definitions | Case Definitions | Deployments |
|---------------------|----------------------|------------------|-------------|
| 2 | 2 | 0 | 2 |

Camunda Cockpit Processes Decisions Cases Human Tasks More **Demo Demo**

Dashboard » Processes

| Process Instance ID | Start Time | End Time | Definition ID | Business Key |
|--------------------------------------|------------------|----------|---|--------------|
| 72986e-d-ceca-11eb-8917-3ef862082519 | 2021-06-16T23... | | ReviewInvoice:1:659ebf0a-ceca-11eb-8917-3e... | |
| 723df2b5-ceca-11eb-8917-3ef862082519 | 2021-06-16T23... | | invoice:2:66449c01-ceca-11eb-8917-3ef8620... | |
| 722ed765-ceca-11eb-8917-3ef862082519 | 2021-06-16T23... | | ReviewInvoice:1:659ebf0a-ceca-11eb-8917-3e... | |
| 676dd08-ceca-11eb-8917-3ef862082519 | 2021-06-16T23... | | invoice:1:659dd4a8-ceca-11eb-8917-3ef8620... | |
| 727e7da3-ceca-11eb-8917-3ef862082519 | 2021-06-11T23... | | invoice:2:66449c01-ceca-11eb-8917-3ef8620... | |
| 721115fb-ceca-11eb-8917-3ef862082519 | 2021-06-11T23... | | invoice:1:659dd4a8-ceca-11eb-8917-3ef8620... | |
| 7250b797-ceca-11eb-8917-3ef862082519 | 2021-06-02T23... | | invoice:2:66449c01-ceca-11eb-8917-3ef8620... | |
| 71a81aae-ceca-11eb-8917-3ef862082519 | 2021-06-02T23... | | invoice:1:659dd4a8-ceca-11eb-8917-3ef8620... | |

2 process definitions deployed

| State | Incidents | Running Instances | Name | Tenant ID | History view | Report | Action |
|-----------------|-----------|-------------------|-----------------|-----------|--------------|--------|--------|
| Green checkmark | 0 | 6 | Invoice Receipt | | History view | Report | Action |

Date and Time displayed in local timezone: Asia/Calcutta

Powered by Camunda Platform / v7.15.2-ee

The screenshot shows the Camunda Cockpit interface. On the left, there's a sidebar with 'Deployment Time' and a 'Deploy' button. Below it, there are two project entries:

- camunda-invoice**: Time: 2021-06-16T23:13:45, Source: process application, Tenant ID: null. It lists two items: 64d8d6f4-ceca-11eb-8917-3ef862082519 and another entry with the same details.
- invoiceBusinessDecisions.dmn**

A search bar at the top right says "Select a resource in the list". At the bottom, it says "Date and Time displayed in local timezone: Asia/Calcutta" and "Powered by Camunda Platform /v7.15.2-ee".

X) Build the Java code in IntelliJ IDE, run the command “mvn clean install” and click on war plugin’s “**war: war**” to generate the WAR file for the project that will be given to Camunda BPM

The screenshot shows the IntelliJ IDEA interface with the Maven tool window open. The project structure on the left shows a 'target' folder containing various Maven artifacts. The central editor shows a Java class named 'LoggerDelegate'. In the bottom 'Run' tool window, a Maven build is in progress, showing the command 'SimpleDemo [org.apache.maven.plugins:maven-war:2.4:war]' and its execution details. The output log shows the assembly of a webapp and the creation of a war file at 'C:\MTech Internship\Camunda\Projects\target\SimpleDemo.war'. The status bar at the bottom indicates the build was successful.

XI) Place the war file generated in the below said path of the WildFly server folder.

| This PC > Downloads > camunda-bpm-ee-wildfly-7.15.2-ee > server > wildfly-22.0.1.Final > standalone > deployments | | | | |
|---|------------------|---------------|----------|--|
| Name | Date modified | Type | Size | |
| camunda-welcome.war | 26-05-2021 16:10 | File folder | | |
| camunda-engine-rest-7.15.2-ee-wildfly.war | 26-05-2021 16:13 | WAR File | 2,876 KB | |
| camunda-engine-rest-7.15.2-ee-wildfly.wa... | 26-05-2021 16:13 | DEPLOYED File | 1 KB | |
| camunda-example-invoice-7.15.2-ee.war | 26-05-2021 16:16 | WAR File | 120 KB | |
| camunda-example-invoice-7.15.2-ee.war.d... | 26-05-2021 16:16 | DEPLOYED File | 1 KB | |
| camunda-h2-webapp-7.15.2-ee.war | 26-05-2021 16:17 | WAR File | 3 KB | |
| camunda-h2-webapp-7.15.2-ee.war.deploy... | 26-05-2021 16:17 | DEPLOYED File | 1 KB | |
| camunda-webapp-ee-jboss-7.15.2-ee.war | 26-05-2021 16:23 | WAR File | 8,609 KB | |
| camunda-webapp-ee-jboss-7.15.2-ee.war.... | 26-05-2021 16:23 | DEPLOYED File | 1 KB | |
| camunda>Welcome.war.deployed | 26-05-2021 16:10 | DEPLOYED File | 1 KB | |
| README.txt | 11-02-2021 18:49 | Text Document | 9 KB | |
| SimpleDemo.war | 17-06-2021 00:12 | WAR File | 9 KB | |
| SimpleDemo.war.deployed | 17-06-2021 00:12 | DEPLOYED File | 1 KB | |

XII) Refresh the cockpit page and we can see the process definition “Service Demo” being deployed.

The screenshot shows the Camunda Cockpit interface for managing processes. At the top, there's a navigation bar with links for Processes, Decisions, Cases, Human Tasks, and More. On the far right, it shows a user profile for 'Demo' and a home icon. Below the navigation is a breadcrumb trail: Dashboard > Processes. The main area is titled 'Search process instances' and features a search bar with 'Add criteria' and a red 'Execute batch' button. A large section below is titled '3 process definitions deployed'. It lists three entries:

| State | Incidents | Running Instances | Name | Tenant ID | History view | Report | Action |
|-------|-----------|-------------------|-----------------|-----------|--------------|--------|--------|
| ✓ | 0 | 6 | Invoice Receipt | | History view | | |
| ✓ | 0 | 2 | Review Invoice | | History view | | |
| ✓ | 0 | 0 | SimpleDemo | | History view | | |

At the bottom of the page, there's a note: 'Date and Time displayed in local timezone: Asia/Calcutta' and a footer: 'Powered by Camunda Platform / v7.15.2-ee'.

Camunda Cockpit Processes Decisions Cases Human Tasks More ▾

Dashboard > Processes > SimpleDemo : Runtime | History

Definition Version: 1
Version Tag: null
Definition ID: SimpleDemo:1:72492679-ced3-1...
Definition Key: SimpleDemo
Definition Name: SimpleDemo
History Time To Live: null
Tenant ID: null
Deployment ID: 7076d777-ced3-11eb-8917-3ef8...
Instances Running:
Date and Time displayed in local timezone: Asia/Calcutta

Activity Instance Statistics: on

Process Instances Incidents Called Process Definitions Job Definitions Modify Documentation

Add criteria No process instances matched by current filter.

Powered by Camunda Platform / v7.15.2-ee

Start process

Search by process name.

Click on the process to start.

Invoice Receipt

SimpleDemo

[Close](#)

Camunda Cockpit Processes Decisions Cases Human Tasks More ▾

Dashboard > Processes > SimpleDemo : 60407e0a-ced4-11eb-8917-3ef862082519 : History | Runtime

Information Filter

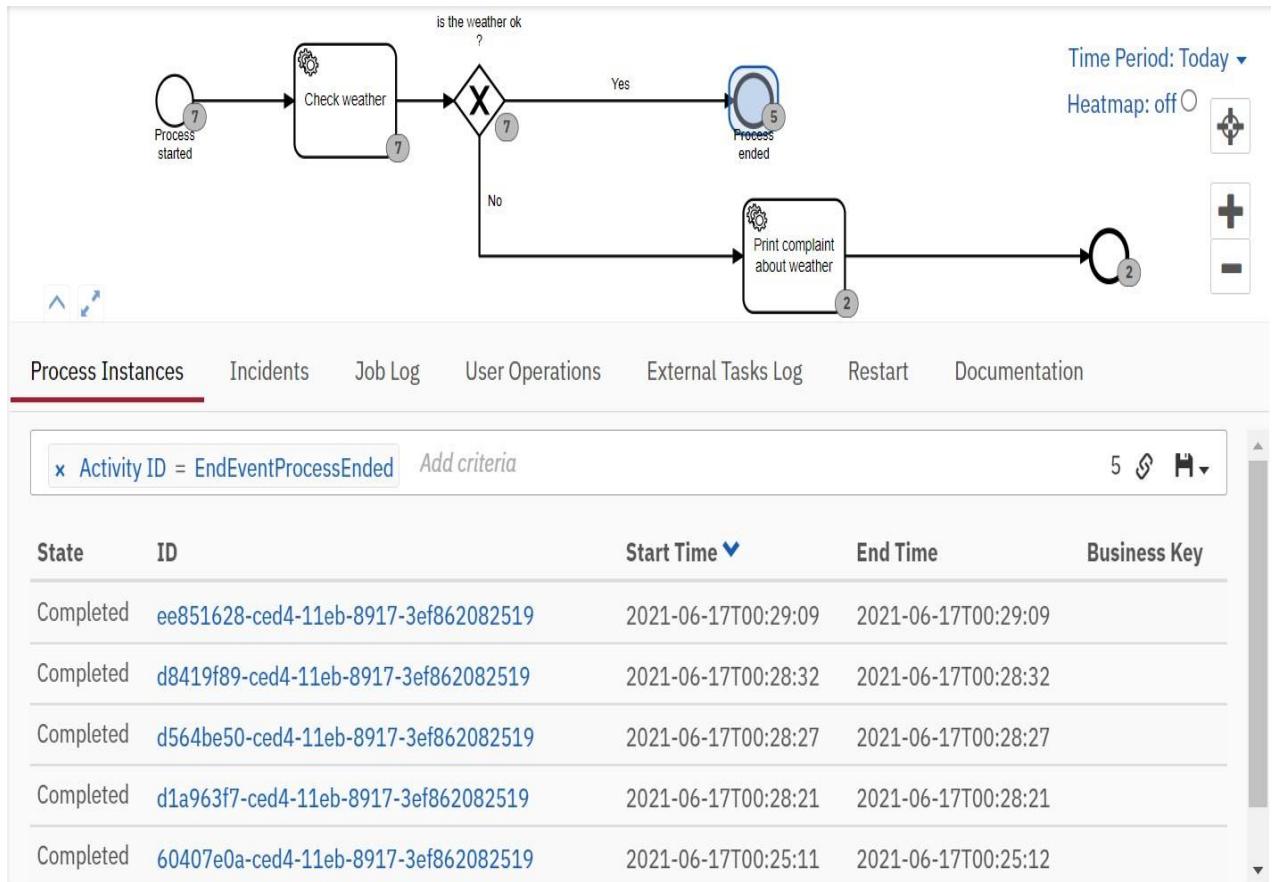
Instance ID: 60407e0a-ced4-11eb-8917-3ef8...
Business Key: null
Definition Version: 1
Definition ID: SimpleDemo:1:72492679-ced3-1...
Definition Key: SimpleDemo
Definition Name: SimpleDemo
Tenant ID: null
Deployment ID: 7076d777-ced3-11eb-8917-3ef8...
Date and Time displayed in local timezone: Asia/Calcutta

Audit Log Variables Called Process Instances Called Case Instances Executed Decision Instances Incidents User Tasks

| State | Activity | Start Time | End Time | Activity Instance ID |
|-------|---------------------|---------------------|---------------------|-------------------------------|
| ○ | Process started | 2021-06-17T00:25:11 | 2021-06-17T00:25:11 | StartEventProcessStarted:6... |
| ○ | Check weather | 2021-06-17T00:25:11 | 2021-06-17T00:25:12 | Activity_1sbomea:60c0355... |
| ○ | is the weather ok ? | 2021-06-17T00:25:12 | 2021-06-17T00:25:12 | Gateway_0vo5ixi:6107297... |
| ○ | Process ended | 2021-06-17T00:25:12 | 2021-06-17T00:25:12 | EndEventProcessEnded:61... |

Powered by Camunda Platform / v7.15.2-ee

XIII) Start multiple instances of the process “Simple Demo” (say 7). Once can see below that five times the weather is okay and the remaining two times the weather is not okay.



Observations/Insights:

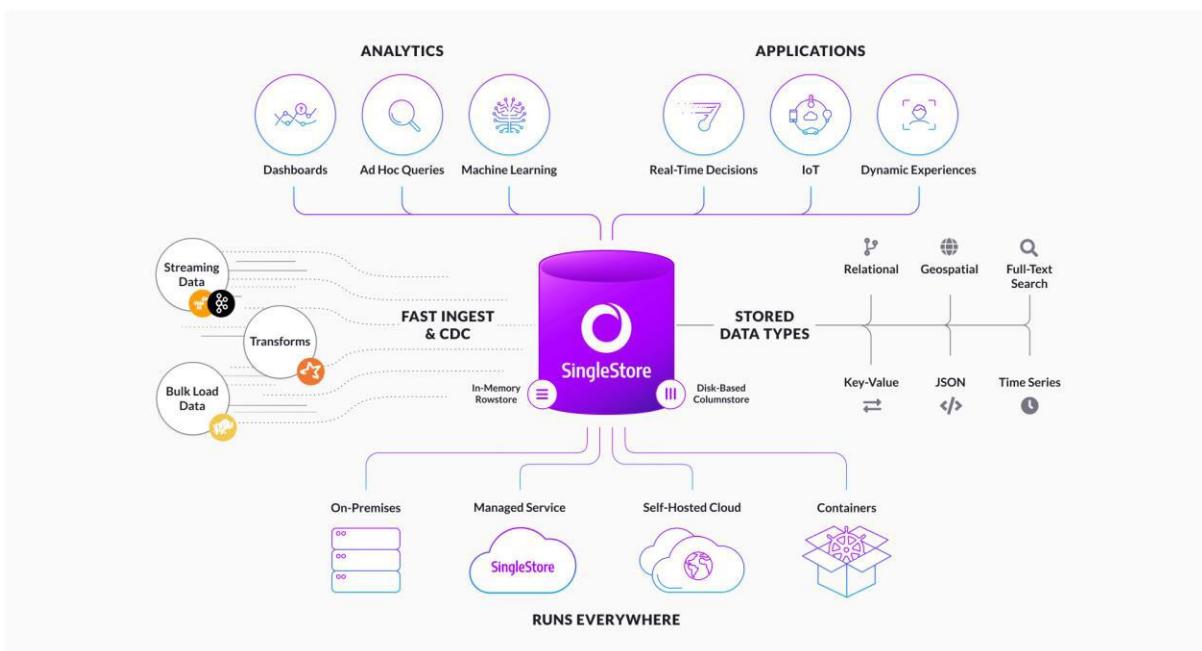
Thus using Camunda BPM tools once can deploy several process diagrams such as diagram for sending emails or invoices and sever other real-time examples.

SingleStore SQL

What is SingleStore (Formerly known as MEMSQL)

SingleStore DB is a distributed, relational database that handles both transactions and real-time analytics at scale. Querying is done through standard SQL drivers and syntax, leveraging a broad ecosystem of drivers and applications.

SingleStore is a scalable SQL database that ingests data continuously to perform operational analytics for the front lines of your business. Ingest millions of events per second with ACID transactions while simultaneously analysing billions of rows of data in relational SQL, JSON, geospatial, and full-text search formats.



What is an In-Memory Database

In-memory databases primarily rely on main memory for storage. They are durable to disk (and respect to ACID properties). In-memory databases can spill on disk or pin data in memory (and take advantage of it). Trade-offs are suitable to systems with lots of memory. These databases tend to be distributed systems because modern-day systems are distributed systems and one can stream memory cost-effectively with a lot of distributed systems. It becomes computationally expensive in the case of vertically scaling a single machine. Also, in-memory databases have a different set of bottlenecks.

All database workloads will be running on in-memory databases because of the following reasons:

- Memory is getting cheaper (about 40% every year).
- In-memory databases leverage SSD's (hence to avoid random writes).

Why and when to use SingleStore

- 1) **Both a transactional database and analytical data warehouse:** SingleStore is designed to act as both a transactional (row store) database, and an analytical (column store) data warehouse. This unique technology eliminates the need for an organization to spin up a transactional database for row-level writes, and a separate analytical data warehouse for historical analysis. Organizations can concurrently write and read from the same distributed system using ANSI SQL.
- 2) **Real-time analytics:** The ability to stream operational data into relational format while concurrently reading data means SingleStore can discover insights on the latest events or compare those with historical data. This approach enables real-time financial reporting or insights on live customer interactions with a single unified database. The integrated architecture can replace more complex systems that require copying of data from a transactional database or data lake into a dedicated data warehouse for analysis. Most relational database management systems (RDBMSs) cannot write and read data at the same time. This unique architecture makes SingleStore an attractive technology for organizations that want to analyse their business continuously while providing a historical reference store for predicting events or for making strategic decisions.
- 3) **Scalability:** SingleStore scales similar to a Hadoop architecture in that SingleStore doesn't require exotic hardware; it scales linearly on commodity hardware. This allows SingleStore to run on-premises, in the cloud, or as a managed service. It's also possible to scale SingleStore instances without taking the cluster offline, which means SingleStore can be easily scaled "just-in-time" when new performance or storage is necessary.

How SingleStore DB Works:

- 1) **Data Ingestion:** i) SingleStore DB can load data continuously or in bulk from a variety of sources. Popular loading sources include: files, a Kafka cluster, cloud repositories like Amazon S3, HDFS, or from other databases. As a distributed system, SingleStore DB ingests data streams using parallel loading to maximize throughput.
- ii) SingleStore Pipelines is an easy-to-use built-in capability that extracts, transforms, and loads external data using sources such as Kafka, S3, Azure Blob, and file systems. It is ideal for scenarios where data from a supported source must be ingested and processed in real-time. This makes SingleStore Pipelines a good alternative to third-party middleware for basic ETL operations that must be executed as fast as possible, thus eliminating traditional long-running processes such as overnight batch jobs.
- iii) For bulk data load operations, SingleStore DB offers a LOAD DATA function that imports files in parallel to maximize performance.

- 2) **High-Performance for OLTP and OLAP Workloads:** i) Single Store DB is a highly-scalable distributed system. Data is sharded automatically amongst nodes in a cluster. Sharding optimizes query performance for both distributed aggregate queries and filtered queries with equality predicates. You can add nodes and redistribute shards (also known as partitions) as needed to scale your workload.
ii) SingleStore DB also supports storing and processing data using an in-memory row store or a disk-based column store. The in-memory row store provides optimum real-time performance for transactional workloads. The disk-based column store is best for analytical workloads across large historical datasets. Row stores are the default table type, but column stores can be created by specifying a column store index type. A combination of the SingleStore DB row store and column store engines simplifies your technology stack by allowing you to merge real-time and historical data in a single query.
iii) In addition to handling mixed workloads, SingleStore DB supports **high concurrency** for simultaneous users. A distributed query optimizer evenly divides the processing workload to maximize the efficiency of CPU usage. Query plans are compiled to machine code and cached to expedite subsequent executions. A key feature of these compiled query plans is they do not pre-specify values for the parameters. This allows SingleStore DB to substitute the values upon request, which enables subsequent queries of the same structure to run quickly. Moreover, with SingleStore DB using Multi-Version Concurrency Control (MVCC) and lock-free data structures, data remains highly accessible, even amidst a high volume of concurrent reads and writes.

- 3) **Analysing Data:** SingleStore DB queries data using scalable standard SQL. The JDBC/ODBC interface enables widely-available BI tools such as Looker, Tableau, and Micro strategy to query and interact with data. Users or applications can connect and write SQL queries for custom visualizations or embedded analytic requirements.

SingleStore DB delivers a number of query performance capabilities to accelerate response time. These include:

- a) True shared-nothing horizontal scaling and distributed parallel query execution
 - b) Column store query processing for fast aggregations and optimized disk consumption
 - c) Query compilation for faster execution: Query vectorization for highly efficient processing by leveraging streamlined, array-oriented processing, and on-chip CPU instructions
- 4) **Deploy anywhere:** SingleStore DB can be deployed anywhere in the cloud or on-premises without requiring specialty hardware to perform effectively; however, it will take advantage of single-instruction/multiple-data (SIMD) CPU instructions, specifically AVX2, and it is NUMA-aware to increase performance on newer machines. It can be deployed on bare metal, in VMs, and in the cloud.

In addition to being fast, consistent, and scalable, SingleStore DB is also durable.

Transactions are committed to disk as log records and periodically compressed as snapshots

of the entire database. If any node goes down, it can restart and use its logs to recover committed transactions.

SingleStore DB also supports high availability through the use of availability groups where nodes in the cluster share copies of data amongst each other in paired configurations. Nodes can failover as needed without the cluster going offline.

- 5) Highly Compatible:** SingleStore DB is an ODBC-compatible database. It is wire protocol compatible with MySQL so that applications that use a MySQL driver can connect to and use SingleStore DB transparently. SingleStore DB supports a broad subset of the MySQL syntax and built-in functions, so most queries that run against MySQL will work unchanged against SingleStore DB. SingleStore DB also supports advanced features such as distributed architecture, high performance via compiled queries, column stores, extensibility, Geospatial, JSON, Window Functions, pipelines, full-text search, time series extensions, approximate fast aggregate functions, and S3, Azure Blob Store, and GCS integration.

In addition to MySQL wire compatibility, SingleStore DB also integrates well with the stream processing framework Apache Spark to provide a simple way to create and manage real-time data pipelines.

How to Deploy a Cluster in SingleStore using Docker:

- 1) Create an account in SingleStore and further steps is to install it.

Follow the steps below to deploy a SingleStore DB cluster in any environment. For more information please read our [in-depth installation and configuration guides](#).

① Select Installation Type

Quick Start 3 mins

Deploy a single-host cluster in a preconfigured Docker container

Full Installation ~30 mins

Deploy a multi-node cluster that is production ready

② Select where you want to deploy SingleStore

Windows

Mac OS X

Linux

Install on Windows with Docker

To run SingleStore on Windows, you will need to run it in a container. This guide will install a preconfigured Docker image that contains a simple one node cluster. [View troubleshooting docs article for help.](#)

① Install Docker

Docker enables you to run SingleStore in a container. Select an edition of Docker to install below. If you already have Docker installed you can skip this step.



For older operating systems we recommend Docker Toolbox.

② Configure Docker

Go to **Settings > Resources > Advanced** to change the configuration to:

4 Cores **4 GB RAM**

③ Set Password

We've generated a random password below, but it is recommended that you change this. You will use this to log in to your cluster. The default user is set as **root**.

[Set Password](#)

The screenshot shows the Docker Desktop documentation page. The left sidebar has a tree view under 'Docker Desktop' with 'Windows' selected, which further branches into 'Install Docker Desktop for Windows', 'User manual', 'Networking', 'Logs and troubleshooting', 'Release notes', and 'Docker Desktop WSL 2 backend'. The main content area title is 'Install Docker Desktop on Windows' with an estimated reading time of 6 minutes. It welcomes users to Docker Desktop for Windows and provides information on system requirements, download URL, installation instructions, and automatic updates. A large blue button labeled 'Docker Desktop for Windows' is present. On the right, there's a sidebar with options like 'Edit this page', 'Request docs changes', and a toggle for dark mode. Below that is a 'On this page:' section with links to various Docker Desktop topics.

② Plan

Free Trial Credits
Test out SingleStore with free trial credits

On-Demand
Spin up a cluster of any size. Cancel at any time.

Proof of Concept
Test a larger workload for a longer period of time with the help of a SingleStore Technical Specialist.

③ Cloud Provider & Region

The screenshot shows the SingleStore cloud provider interface. The top navigation bar includes 'SingleStore', 'Clusters > Aishvarya Raman's Cluster', 'Last updated: 7:06:32 PM', '+ Load Data', 'Connect', and a help icon. The left sidebar has sections for 'CLUSTERS' (Cluster Overview, Aishvarya Raman's Cluster), 'QUERY' (SQL Editor, Visual Explain), and a detailed view for 'AISHVARYA RAMAN (AISHV...)' with a copy icon. The main content area shows '0% / Starting Deployment' and the 'Overview' tab for the cluster. It displays 'Cluster Properties' including 'Cluster Type: Production', 'Region: AWS US East 1 (N. Virginia)', 'Expiration: -', 'SingleStore DB Version: 7.3.11', 'Size: -', 'Created: 6/15/21 at 7:06:29 PM', and 'High Availability: On'. At the bottom are tabs for 'User Access', 'Load Data', and 'Connect'.

The screenshot shows the SingleStore Cluster Overview page. At the top, a green banner displays the message: "Your cluster has been created successfully and is now online". Below the banner, the navigation bar includes "Clusters > Aishvarya Raman's Cluster", "Upgrade cluster", "+ Load Data", "Connect", and a three-dot menu. The main content area features tabs for "Overview", "Firewall", "Security", "Updates", and "Monitoring", with "Overview" selected. A large section titled "How can we help?" contains two cards: "Interactive Tutorials" (with sub-options like TEST, LOAD DATA, Create Database, Create Table, Create View) and "Solution Guides" (with sub-options like IMPLEMENT, AUGMENTING, HOW TO REPLACE, BUILD NEW). On the right side, there is a "Chat with an engineer" button.

2) Installing Docker desktop and creating a container

The screenshot shows the Docker Desktop 3.4.0 setup window. The title bar says "Installing Docker Desktop 3.4.0 (65384)". The main content area displays the progress of unpacking files, which includes:

- Unpacking file: frontend/Docker Desktop.exe
- Unpacking file: frontend/d3dcompiler_47.dll
- Unpacking file: DockerCli.exe
- Unpacking file: Docker.WPF.dll
- Unpacking file: Docker.HttpApi.dll
- Unpacking file: Docker.Core.dll
- Unpacking file: Docker.Backend.dll
- Unpacking file: Docker.ApiServices.dll
- Unpacking file: Docker Desktop.exe
- Unpacking file: Docker Desktop Installer.exe
- Unpacking file: courgette64.exe
- Unpacking file: Bugsnag.dll
- Unpacking file: BITSReference5_0.dll
- Unpacking file: resources/wsl/wsl-data.tar

Click on PowerShell and run the following commands to start a container

```
PS C:\WINDOWS\system32> docker run --help

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container

Options:
  --add-host list           Add a custom host-to-IP mapping
                            (host:ip)
  -a, --attach list         Attach to STDIN, STDOUT or STDERR
  --blkio-weight uint16     Block IO (relative weight),
                            between 10 and 1000, or 0 to
                            disable (default 0)
  --blkio-weight-device list Block IO weight (relative device
                            weight) (default [])
  --cap-add list            Add Linux capabilities
  --cap-drop list           Drop Linux capabilities
  --cgroup-parent string   Optional parent cgroup for the
                            container
  --cgroupns string         Cgroup namespace to use
                            (host|private)
                            'host': Run the container in
                            the Docker host's cgroup namespace
                            'private': Run the container in
                            its own private cgroup namespace
                            ''': Use the cgroup
                            namespace as configured by the
                            default-cgroupns-mode
                            option on the daemon (default)
  --cidfile string          Write the container ID to the file
  --cpu-period int          Limit CPU CFS (Completely Fair
                            Scheduler) period
  --cpu-quota int           Limit CPU CFS (Completely Fair
```

```
PS C:\WINDOWS\system32> docker run -d -p 80:80 docker/getting-started
```

SETUP DOCKER FOR SINGLESTORE ADMINISTRATOR LABS



Windows Client and Ubuntu Server: Part 1 of 4

Requirements for your Windows Client

- Must be running 64-bit Windows 7 or higher
- Must have 4 cores and at least 4 Gb RAM
- OSX must be 10.13 or newer

Install Docker

1. Navigate to <https://hub.docker.com/editions/community/docker-ce-desktop-windows/>
2. Click Get Docker.
3. Choose a location for Docker Desktop Installer.exe and click Save.
4. Using File Explorer, navigate to the location of Docker Desktop Installer.exe and double-click it.
5. If requested to allow Docker to make changes click Yes.
6. Follow the instructions for installation.
7. From the Start menu folder click Docker Desktop.
8. The Docker icon will appear in the notifications area.
9. Right-click the Docker icon and select Settings.
10. In the Resources section select 4 for CPUs and 4 GB for Memory.
11. Verify that Docker is installed by running Docker run hello-world at a Terminal prompt and verify that Hello from Docker! is displayed without error.

SETUP DOCKER FOR SINGLESTORE ADMINISTRATOR LABS



Windows Client and Ubuntu Server: Part 2 of 4

Download and Run Docker Containers

1. Open a Windows PowerShell window.
2. At the PowerShell prompt type the following commands. The containers will download and start. In running the master container the “-p” parameter sets port mapping. To the left of the colon is the client port and to the right is the port used by the container.

```
docker run -d -p 22:22 -p 3306:3306 -p 8080:8080 -p 8081:8081 --name master singlestoretraining/centos_master:latest
docker run -d --name child singlestoretraining/ubuntu_child:latest
docker run -d --name leaf1 singlestoretraining/ubuntu_leaf1:latest
docker run -d --name leaf2 singlestoretraining/ubuntu_leaf2:latest
```

1. At the PowerShell prompt type: docker ps.
2. The results should look something like this. The status for all containers should be “up”:

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAMES |
|--------------|--|------------------------|----------------|---------------|--|--------|
| a415b65f93fb | singlestoretraining/ubuntu_leaf2:latest | "/usr/bin/supervisord" | 10 seconds ago | Up 10 seconds | 22/tcp, 3306/tcp, 8080/tcp | leaf2 |
| 1cb57296a235 | singlestoretraining/ubuntu_leaf1:latest | "/usr/bin/supervisord" | 26 seconds ago | Up 25 seconds | 22/tcp, 3306/tcp, 8080/tcp | leaf1 |
| 2da8988a2268 | singlestoretraining/ubuntu_child:latest | "/usr/bin/supervisord" | 42 seconds ago | Up 42 seconds | 22/tcp, 3306/tcp, 8080/tcp | child |
| 9ed9a50b6488 | singlestoretraining/ubuntu_master:latest | "/usr/bin/supervisord" | 55 seconds ago | Up 56 seconds | 0.0.0.0:22->22/tcp, 0.0.0.0:3306->3306/tcp, 0.0.0.0:8080->8080/tcp | master |

SETUP DOCKER FOR SINGLESTORE ADMINISTRATOR LABS



Windows Client and Ubuntu Server: Part 3 of 4

Get IP Addresses of Containers

1. At a Terminal prompt type the following command:

```
docker network inspect bridge | Select-String -Pattern 'Name', 'IPv4Address' | % {
```

2. The result should look like this. You'll need these IPAddresses for the labs:

```
"Name": "bridge",
  "Name": "child",
  "IPv4Address": "172.17.0.3/16",
  "Name": "leaf2",
  "IPv4Address": "172.17.0.5/16",
  "Name": "leaf1",
  "IPv4Address": "172.17.0.4/16",
  "Name": "master",
  "IPv4Address": "172.17.0.2/16",
  "com.docker.network.bridge.name": "docker0",
```

SETUP DOCKER FOR SINGLESTORE ADMINISTRATOR LABS



Windows Client and Ubuntu Server: Part 4 of 4

Verify that you can SSH to the Master node

1. Download the SSH key from <https://drive.google.com/file/d/1uxawQ3weAFs1I24-InL5Pgl2BdKAvt5Gr/view?usp=sharing>
2. Create the .ssh directory by typing `mkdir ~/.ssh` at a PowerShell prompt.
3. Copy the downloaded file to `~/ssh/`
If the file was downloaded to the Downloads folder you can copy the file at a PowerShell prompt by typing:
`cp ~/Downloads/singlestorekey.pem ~/.ssh/singlestorekey.pem`
4. At a Terminal prompt type: `ssh -i ~/.ssh/singlestorekey.pem ubuntu@localhost`
5. Verify that you are at a `ubuntu@master $` prompt. If so, you are connected to the container that will become your master aggregator and you are ready to continue with your lab. If not, click the button below.

Observation/Insights:

The ability to combine the scalability of NoSQL databases and the familiar structure, transaction capabilities, and SQL support of traditional relational databases.

Problem Statement: How is JSON used in SingleStore database

JSON in SingleStore DB:

- 1) SingleStore DB exposes a JavaScript Object Notation (JSON) column-type that implements the JSON standard. You can define columns in SingleStore DB tables as the JSON data type in order to efficiently store and query JSON values. A JSON value is a JavaScript object, which means it is represented as a series of name/value pairs that are written according to a structured format and sequence.

```
2) {  
    "key-name": "string Value",  
    "another-key": "hello world"  
}
```

myJSON[“key-name”] => “string Value”

- 3) **Structured and Schema-less:** While structured in key-value pairs, JSON is schema less by nature. Values can be numbers, strings, Booleans, nulls or even nested JSON objects or arrays.
- 4) **Defining JSON Columns:** Defining a JSON column in a SingleStore DB table is as simple as specifying the JSON data type in the CREATE TABLE command:

```
CREATE TABLE assets (  
    tag_id BIGINT PRIMARY KEY,  
    name TEXT NOT NULL,  
    description TEXT,  
    properties JSON NOT NULL  
);
```

Visual Explain

10);

- 5) **DML- Accessing data in JSON columns:** By default, SingleStore DB does not preserve columnstore JSON NULL values and empty arrays. To change this behaviour and preserve original columnstore JSON, set the **preserve_original_colstore_json** engine variable to ON. This flag preserves the original columnstore JSON for any new data that is loaded; it does not update the existing data. By default, this variable is set to AUTO (same as OFF).



SingleStore

- CLUSTERS**
- Cluster Overview
- Aishvarya Raman's Cluster
- Aishvarya Raman's Cluster

- QUERY**
- SQL Editor**
- Visual Explain

AISHVARYA RAMAN (AISHVA... ↗)

Aishvarya Raman's Cluster • json_db

```

13  INSERT INTO humans VALUES(
14    1,
15    'Aishvarya',
16    'Raman',
17    25,
18    {"teams": [
19      {"Cricket": "India"}, 
20      {"IPL": "Chennai Super Kings"}, 
21      {"football": "Giants"}, 
22      {"hockey": "India"} 
23    ]})
24 );
25 INSERT INTO humans VALUES(
26    2,
27    'Samiksha',
28    'Rathod',
29    30,
30    {"teams": [
31      {"Cricket": "Australia"}, 
32      {"IPL": "Royal Challengers Bangalore"}, 
33      {"football": "Arsenal"}, 
34      {"hockey": "India"} 
35    ]})
36 );

```

37 SELECT * FROM humans ORDER BY human_id;

38 SELECT * FROM humans WHERE human_id=1;

| human_id | first_name | last_name | age | favorite_teams |
|----------|------------|-----------|-----|---|
| 1 | Aishvarya | Raman | 25 | [{"teams": [{"Cricket": "India"}, {"IPL": "Chennai Super Kings"}, {"football": "Giants"}, {"hockey": "India"}]}] |
| 2 | Samiksha | Rathod | 30 | [{"teams": [{"Cricket": "Australia"}, {"IPL": "Royal Challengers Bangalore"}, {"football": "Arsenal"}, {"hockey": "India"}]}] |

Message Logs

Save as CSV

| first_name | last_name | favorite_teams::teams |
|------------|-----------|---|
| Aishvarya | Raman | [{"Cricket": "India"}, {"IPL": "Chennai Super Kings"}, {"football": "Giants"}, {"hockey": "India"}] |

```
39  SELECT first_name, last_name, favorite_teams::teams::`2` FROM humans
40  WHERE human_id=1;
```

Message Logs

| first_name | last_name | favorite_teams::teams::`2` |
|------------|-----------|----------------------------|
| Aishvarya | Raman | {"football": "Giants"} |

```
39  SELECT first_name, last_name, favorite_teams::teams::`1`::$IPL FROM humans
40  WHERE human_id=1;
41  |
```

Message Logs

| first_name | last_name | favorite_teams::teams::`1`::\$IPL |
|------------|-----------|-----------------------------------|
| Aishvarya | Raman | Chennai Super Kings |

```
42  UPDATE humans SET favorite_teams::teams::`1`::$IPL = 'Mumbai Indians'
43  WHERE human_id=2;|
```

Message Logs

| Insert ID | Affected Rows | Changed Rows |
|-----------|---------------|--------------|
| 0 | 1 | 1 |

```
44  SELECT * FROM humans ORDER BY human_id;
```

Message Logs

290 ms, Showing 2 rows

| human_id | first_name | last_name | age | favorite_teams |
|----------|------------|-----------|-----|--|
| 1 | Aishvarya | Raman | 25 | {"teams": [{"Cricket": "India"}, {"IPL": "Chennai Super Kings"}, {"football": "Giants"}, {"hockey": "India"}]} |
| 2 | Samiksha | Rathod | 30 | {"teams": [{"Cricket": "Australia"}, {"IPL": "Mumbai Indians"}, {"football": "Arsenal"}, {"hockey": "India"}]} |

Morgan Stanley

```
45  SELECT human_id, first_name, last_name, age, JSON_EXTRACT_STRING(favorite_teams, "teams", 1, "IPL")
46  FROM humans ORDER BY human_id;
```

The screenshot shows a database interface with a toolbar at the top. The toolbar includes tabs for 'Message Logs' (selected), 'SELECT huma...', '① SELECT huma...', 'SELECT * FRO...', 'SELECT * FRO...', and other icons. Below the toolbar is a button labeled 'Save as CSV'. The status bar indicates '329 ms, Showing 2 rows'. The main area displays a table with the following data:

| human_id | first_name | last_name | age | JSON_EXTRACT_STRING(favorite_teams, "teams", 1, "IPL") |
|----------|------------|-----------|-----|--|
| 1 | Aishvarya | Raman | 25 | Chennai Super Kings |
| 2 | Samiksha | Rathod | 30 | Mumbai Indians |

- 6) The ::, ::\$, ::% operators are actually just convenient aliases to be used for the JSON_EXTRACT_<type>

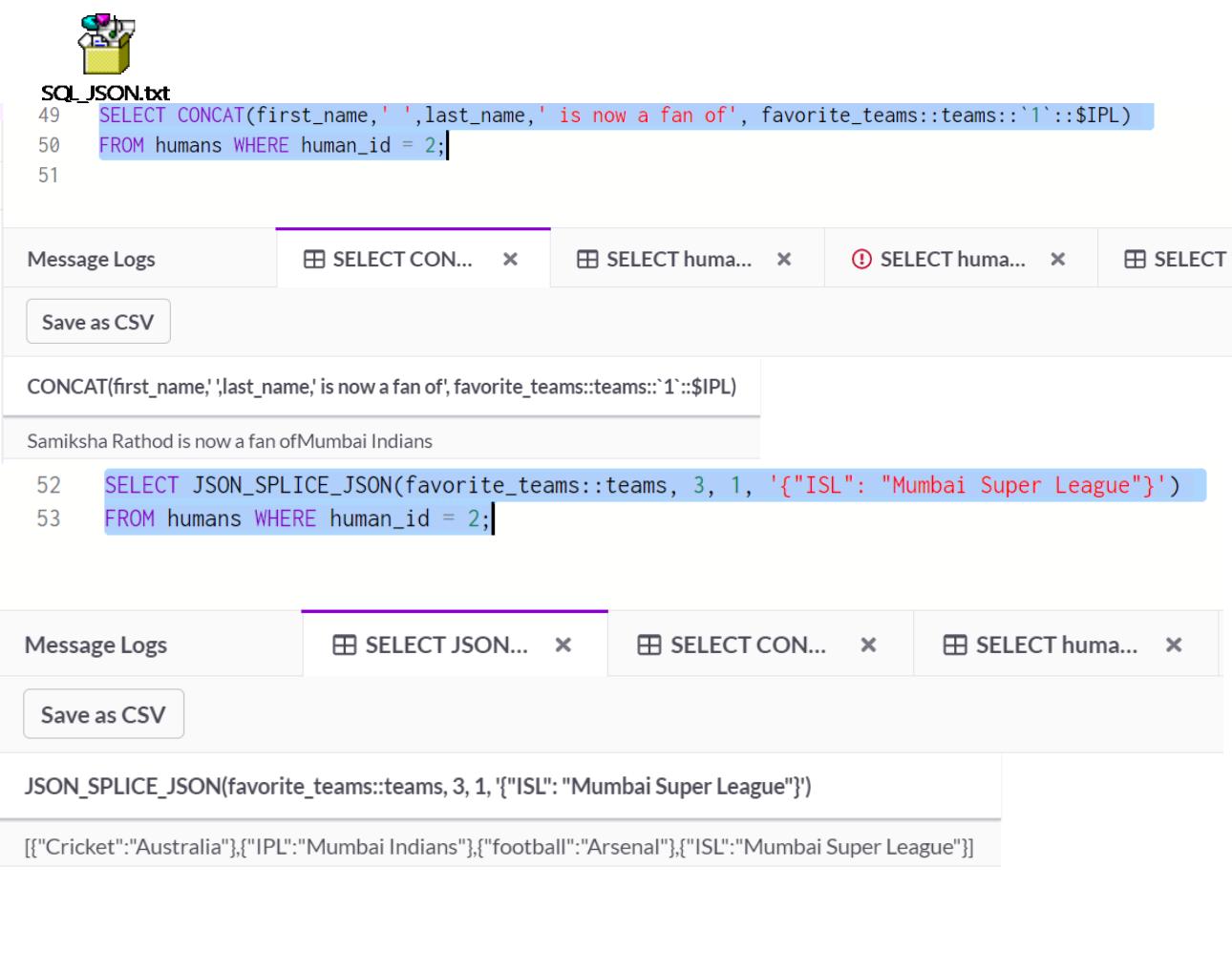
Syntax- i) JSON_EXTRACT_<type> (json, keypath)

ii) JSON_SET_<type> (json, keypath)

iii) <type> can be

- a) Double: For all numeric data, equivalent to the ::% operator.
- b) String: For all text and binary types, equivalent to the ::\$ operator.
- c) JSON: For all valid JavaScript types.

- 7) **JSON_SPICE_<type> (json, start, length, value, [value, ...])** – Deletes the given element indices from an array, and optionally slices new values in.



```

SQL_JSON.txt
49 SELECT CONCAT(first_name, ' ', last_name, ' is now a fan of', favorite_teams::teams::`1`::$IPL)
50 FROM humans WHERE human_id = 2;
51

Message Logs      SELECT CON... x      SELECT huma... x      SELECT huma... x      SELECT

Save as CSV

CONCAT(first_name, ' ', last_name, ' is now a fan of', favorite_teams::teams::`1`::$IPL)
Samiksha Rathod is now a fan of Mumbai Indians

52 SELECT JSON_SPICE_JSON(favorite_teams::teams, 3, 1, '{"ISL": "Mumbai Super League"})'
53 FROM humans WHERE human_id = 2;

Message Logs      SELECT JSON... x      SELECT CON... x      SELECT huma... x

Save as CSV

JSON_SPICE_JSON(favorite_teams::teams, 3, 1, {"ISL": "Mumbai Super League"})
[{"Cricket": "Australia"}, {"IPL": "Mumbai Indians"}, {"football": "Arsenal"}, {"ISL": "Mumbai Super League"}]

```

Problem Statement: Build a SingleStore Web Application using Flask and Python

What is Flask?

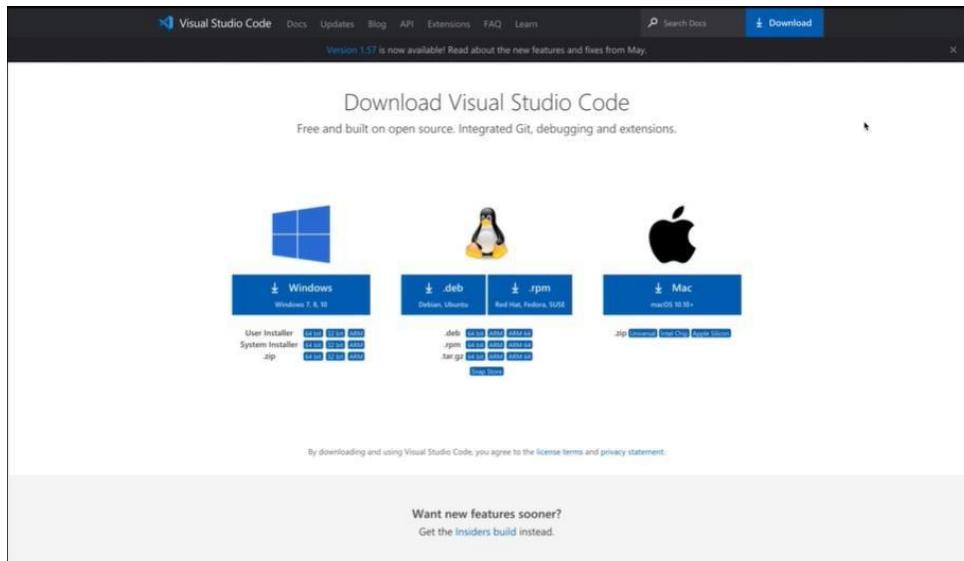
- Flask is a **micro web framework** written in Python. Flask depends on the Jinja template engine and the Werkzeug WSGI toolkit. “Micro” does not mean that your whole web application has to fit into a single Python file (although it certainly can), nor does it mean that Flask is lacking in functionality. The “micro” in micro

framework means Flask aims to keep the core simple but extensible. Flask won't make many decisions for you, such as what database to use. Those decisions that it does make, such as what templating engine to use, are easy to change. Everything else is up to you, so that Flask can be everything you need and nothing you don't.

- By default, Flask does not include a database abstraction layer, form validation or anything else where different libraries already exist that can handle that. Instead, Flask supports extensions to add such functionality to your application as if it was implemented in Flask itself. Numerous extensions provide database integration, form validation, upload handling, various open authentication technologies, and more. Flask may be "micro", but it's ready for production use on a variety of needs.

Building a demo Flask application using Python

- 1) Install Visual Studio Code or Sublime Text Editor



anaconda / packages / flask 1.1.2

A microframework based on Werkzeug, Jinja2 and good intentions.

Conda Files Labels Badges

📄 License: [BSD 3-Clause](#)
🏡 Home: <http://flask.pocoo.org>
🔗 Development: <https://github.com/pallets/flask>
📘 Documentation: <http://flask.pocoo.org/docs/1.0/>
⬇️ 206275 total downloads
📅 Last upload: 5 months and 30 days ago

Installers

conda install [?](#)

| | |
|------------------|--------|
| linux-ppc64le | v1.0.3 |
| linux-64 | v1.0.3 |
| win-32 | v1.1.1 |
| apple-mac noarch | v1.1.2 |
| osx-64 | v1.0.3 |
| linux-32 | v1.0.2 |

2) Install Flask and Python3 in virtual environment using Anaconda Prompt

```
(base) C:\WINDOWS\system32>conda install -c anaconda flask
Collecting package metadata (current_repodata.json): done
Solving environment: |
The environment is inconsistent, please check the package plan carefully
The following packages are causing the inconsistency:

- defaults/win-64::aiohttp==3.7.4=py38h2bbff1b_1
- defaults/win-64::anaconda-client==1.8.0=py38haa95532_0
- defaults/win-64::anaconda-navigator==1.9.12=py38_0
- defaults/win-64::anyio==2.2.0=py38haa95532_2
- defaults/win-64::argon2-cffi==20.1.0=py38h2bbff1b_1
- defaults/win-64::astor==0.8.1=py38haa95532_0
- defaults/noarch::astunparse==1.6.3=py_0
- defaults/win-64::async-timeout==3.0.1=py38haa95532_0
- defaults/noarch::async_generator==1.10=pyhd3eb1b0_0
- defaults/noarch::attrs==21.2.0=pyhd3eb1b0_0
- defaults/noarch::babel==2.9.1=pyhd3eb1b0_0
- defaults/noarch::backcall==0.2.0=pyhd3eb1b0_0
- defaults/noarch::backports==1.0=pyhd3eb1b0_2
- defaults/noarch::backports.functools_lru_cache==1.6.4=pyhd3eb1b0_0
- defaults/noarch::backports.tempfile==1.0=pyhd3eb1b0_1
- defaults/noarch::backports.weakref==1.0.post1=py_1
- defaults/noarch::beautifulsoup4==4.9.3=pyha847dfd_0
- defaults/noarch::bleach==3.3.0=pyhd3eb1b0_0
- defaults/win-64::blinker==1.4=py38haa95532_0
- defaults/win-64::brotlipy==0.7.0=py38h2bbff1b_1003
- defaults/noarch::cachetools==4.2.2=pyhd3eb1b0_0
- defaults/win-64::certifi==2021.5.30=py38haa95532_0
- defaults/win-64::cffi==1.14.5=py38hcd4344a_0
- defaults/win-64::chardet==3.0.4=py38haa95532_1003

The following NEW packages will be INSTALLED:

absl-py      anaconda/win-64::absl-py-0.10.0-py38_0
decorator    anaconda/noarch::decorator-4.4.2-py_0
flask        anaconda/noarch::flask-1.1.2-py_0
grpcio       anaconda/win-64::grpcio-1.31.0-py38he7da953_0
h5py         anaconda/win-64::h5py-2.10.0-py38h5e291fa_0
itsdangerous anaconda/noarch::itsdangerous-1.1.0-py_0
numpy-base   pkgs/main/win-64::numpy-base-1.20.1-py38haf7ebc8_0
setuptools   anaconda/win-64::setuptools-58.3.0-py38h949ed1a_1
six          anaconda/noarch::six-1.15.0-py_0
typing_extensions pkgs/main/noarch::typing_extensions-3.10.0.0-pyh06a4308_0

The following packages will be SUPERSEDED by a higher-priority channel:

ca-certificates  pkgs/main::ca-certificates-2021.5.25--> anaconda::ca-certificates-2020.10.14-0

The following packages will be DOWNGRADED:

numpy          1.20.2-py38ha4e8547_0 --> 1.20.1-py38h34a8a5c_0

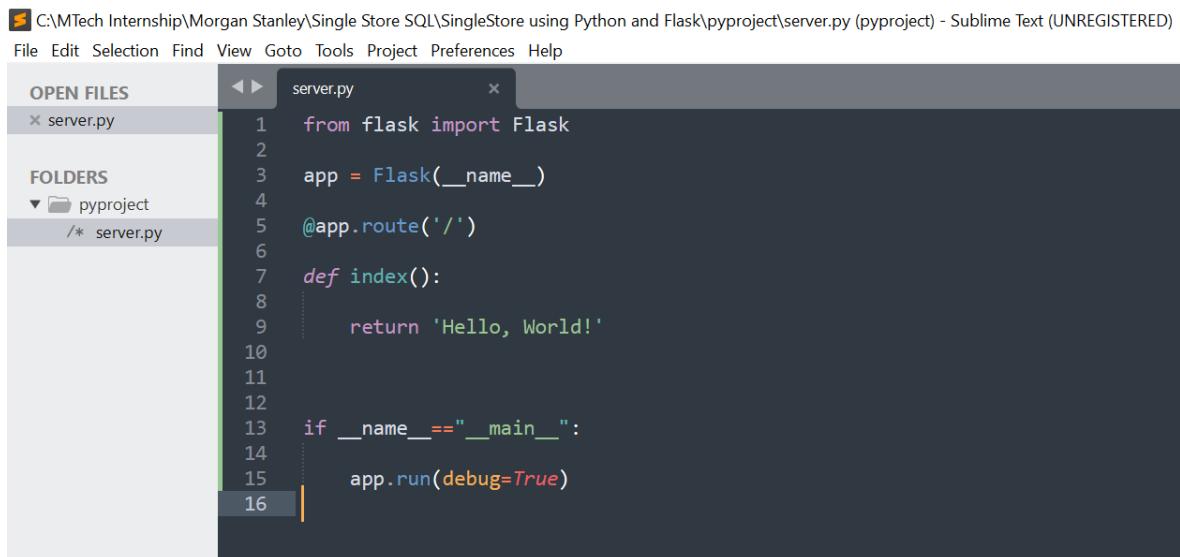
Proceed ([y]/n)?
```

Downloading and Extracting Packages

| | | | |
|-------------------|--------|-------|------|
| decorator-4.4.2 | 14 KB | ##### | 100% |
| six-1.15.0 | 13 KB | ##### | 100% |
| numpy-base-1.20.1 | 4.2 MB | ##### | 99% |

```
(base) C:\WINDOWS\system32>pip install flask
Collecting flask
  Downloading Flask-2.0.1-py3-none-any.whl (94 kB)
    |
    |████████████████████████████████| 94 kB 268 kB/s
Requirement already satisfied: Jinja2>=3.0 in c:\users\abc\anaconda3\lib\site-packages (from flask) (3.0.1)
Collecting itsdangerous>=2.0
  Downloading itsdangerous-2.0.1-py3-none-any.whl (18 kB)
Collecting Werkzeug>=2.0
  Downloading Werkzeug-2.0.1-py3-none-any.whl (288 kB)
    |
    |████████████████████████████████| 288 kB 2.2 MB/s
Requirement already satisfied: click>=7.1.2 in c:\users\abc\anaconda3\lib\site-packages (from flask) (8.0.1)
Requirement already satisfied: colorama in c:\users\abc\anaconda3\lib\site-packages (from click>=7.1.2->flask) (0.4.4)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\abc\anaconda3\lib\site-packages (from Jinja2>=3.0->flask) (2.0.1)
Installing collected packages: Werkzeug, itsdangerous, flask
  Attempting uninstall: Werkzeug
    Found existing installation: Werkzeug 1.0.1
    Uninstalling Werkzeug-1.0.1:
      Successfully uninstalled Werkzeug-1.0.1
Successfully installed Werkzeug-2.0.1 flask-2.0.1 itsdangerous-2.0.1
```

3) Create a “server.py” code in your python directory



The screenshot shows a Sublime Text window with the following details:

- File Path:** C:\MTech Internship\Morgan Stanley\Single Store SQL\SingleStore using Python and Flask\pyproject\server.py (pyproject) - Sublime Text (UNREGISTERED)
- Menu Bar:** File Edit Selection Find View Goto Tools Project Preferences Help
- Open Files:** server.py
- Folders:** pyproject /* server.py
- Code Content:**

```
 1  from flask import Flask
 2
 3  app = Flask(__name__)
 4
 5  @app.route('/')
 6
 7  def index():
 8
 9      return 'Hello, World!'
10
11
12
13  if __name__=="__main__":
14
15      app.run(debug=True)
16
```

4) Run “server.py” from Anaconda prompt

```
(base) C:\WINDOWS\system32>cd C:\MTech Internship\Morgan Stanley\Single Store SQL\SingleStore using Python and Flask\pypyproject  
(base) C:\MTech Internship\Morgan Stanley\Single Store SQL\SingleStore using Python and Flask\pypyproject>python server.py  
* Serving Flask app 'server' (lazy loading)  
* Environment: production  
WARNING: This is a development server. Do not use it in a production deployment.  
Use a production WSGI server instead.  
* Debug mode: on  
* Restarting with stat  
* Debugger is active!  
* Debugger PIN: 349-995-757  
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)  
127.0.0.1 - - [04/Aug/2021 19:43:58] "GET / HTTP/1.1" 200 -  
127.0.0.1 - - [04/Aug/2021 19:44:00] "GET /favicon.ico HTTP/1.1" 404 -
```



← → ⌂ ⓘ 127.0.0.1:5000

Hello, World!

Dependencies with Flask

- **Werkzeug** implements WSGI, the standard Python interface between applications and servers.
- **Jinja** is a template language that renders the pages your application serves.
- **MarkupSafe** comes with Jinja. It escapes untrusted input when rendering templates to avoid injection attacks.
- **ItsDangerous** securely signs data to ensure its integrity. This is used to protect Flask's session cookie.
- Click is a framework for writing command line applications. It provides the flask command and allows adding custom management commands.

How to render an entire HTML page instead of a simple string?

- 1) i) We need to always create HTML pages in the main root directory under "templates" folder because flask checks for the exact folder name and if not found an error occurs.
- ii) Created index.html, dashboard.html and edit.html
- iii) Dashboard.html: Displays the registration and login page
- iv) Edit.html: Displays the To-Do item list that can be edited by the registered user

```
index.html x dashboard.html x | edit.html x

<!DOCTYPE html>
<html>
<head>
    <title>SingleStore Home</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"
        integrity="sha384-HSMxcRTRxnN+Bdg0JdbxYKrThecOKuH5zCYotlSAcp1+c8xmyTe9GYg1l9a69psu" crossorigin="anonymous">
</head>

<body>
    <div class="container">
        <div class="text-center">
            <h1 class="display-4">Welcome to SingleSpace!</h1>
            <h3>The place to manage your To Dos</h3>
        </div>

        <div class="register box">
            <h3>Register</h3>
            <form action="/register" method="post" autocomplete="off">
                <label for="uname">Username:</label>
                <input id="uname" type="text" name="Username"><br>

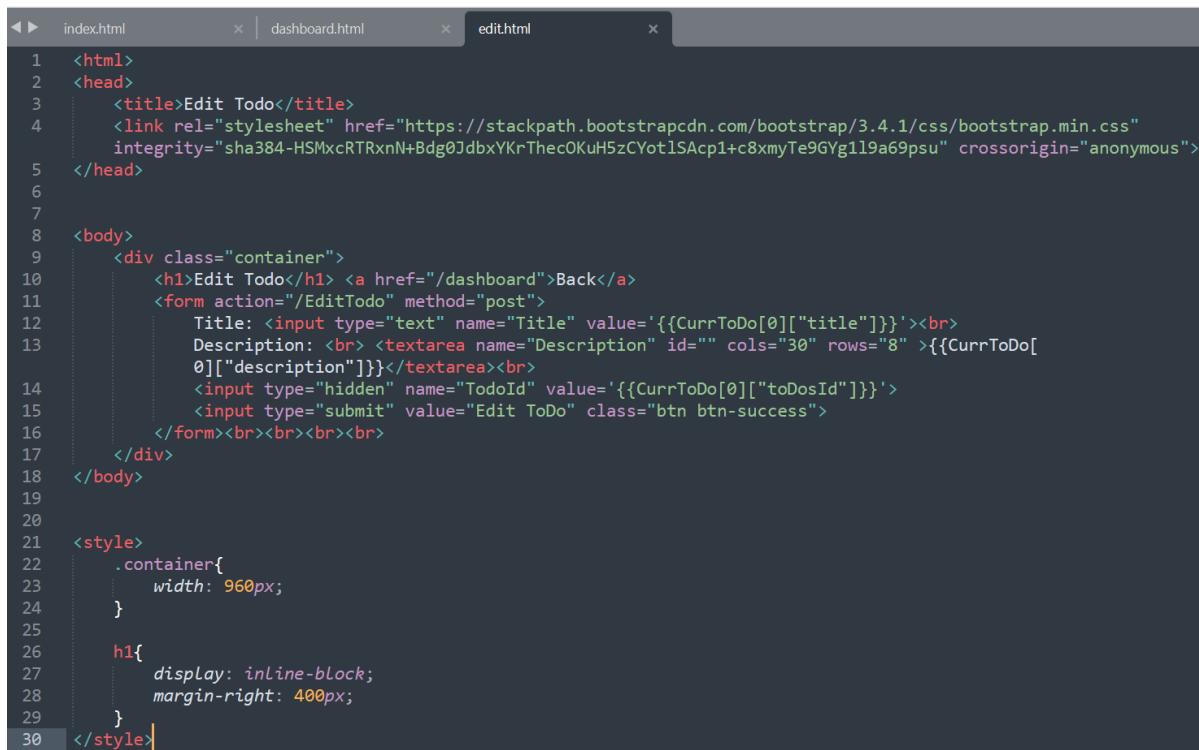
                <label for="email">Email:</label>
                <input id="email" type="email" name="Email"><br>

                <label for="pw">Password:</label>
                <input id="pw" type="password" name="Password"><br>

                <label for="pw_conf">Password Confirm:</label>
                <input id="pw_conf" type="password" name="PasswordConfirmation"><br><br>

                <input type="submit" value="Register" class="btn btn-success">
            </form>
        </div>
    </div>
</body>
```

```
index.html      x dashboard.html      x edit.html      x
1  <html>
2  <head>
3      <title>SingleStore Dashboard</title>
4      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"
5          integrity="sha384-HSMxcRTRxnN+Bdg0JdbxYKrThecOKuH5zCYotlSAcp1+c8xmyTe9GYg1l9a69psu" crossorigin="anonymous">
6  </head>
7
8  <body>
9      <div class="container">
10         <div class="dash">
11             <h1>SingleSpace Dashboard</h1> <a href="/logout">Log Out</a>
12             <h3>Welcome, {{CurrUser[0]["username"]}}</h3>
13             <br><br>
14
15             <div class="post">
16                 <h3>Let's Create some ToDos!</h3>
17                 <form action="/addToDo" method="post" autocomplete="off">
18                     Title: <input type="text" name="Title"><br>
19                     Description: <br> <textarea name="Description" id="" cols="30" rows="8"></textarea><br>
20                     <input type="submit" value="Add ToDo" class="btn btn-success">
21                 </form><br><br><br><br>
22         </div>
23
24         <div class="all">
25             {% for todo in AllToDos %}
26                 <p><b>Title:</b> {{ todo['title'] }}</p>
27                 <p><b>Description:</b> {{ todo['description'] }}</p>
28                 <a href="/edit/{{todo['toDosId']}}">Edit</a> <span> - </span> <a href="/delete/{{todo['toDosId']}}">Delete</a>
29                 <br>
30             {% endfor %}
31         </div>
32     </div>
33 </body>
34 </html>
```



```
1 <html>
2 <head>
3     <title>Edit Todo</title>
4     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"
5         integrity="sha384-HSMxcRTRxnN+Bdg0JdbxYKrThecOKuH5zCYotlSAcp1+c8xmyTe9GYg1l9a69psu" crossorigin="anonymous">
6 </head>
7
8 <body>
9     <div class="container">
10        <h1>Edit Todo</h1> <a href="/dashboard">Back</a>
11        <form action="/EditTodo" method="post">
12            Title: <input type="text" name="Title" value='{{CurrToDo[0]["title"]}}'><br>
13            Description: <br> <textarea name="Description" id="" cols="30" rows="8" >{{CurrToDo[
14                0]["description"]}}</textarea><br>
15            <input type="hidden" name="TodoId" value='{{CurrToDo[0]["toDosId"]}}'>
16            <input type="submit" value="Edit ToDo" class="btn btn-success">
17        </form><br><br><br><br>
18    </div>
19 </body>
20
21 <style>
22     .container{
23         width: 960px;
24     }
25
26     h1{
27         display: inline-block;
28         margin-right: 400px;
29     }
30 </style>
```

- 2) Install “pymysql” in Anaconda Virtual Environment for creating database connectivity between flask and SingleStore

```
(base) C:\Windows\System32>pip install pymysql
Collecting pymysql
  Downloading PyMySQL-1.0.2-py3-none-any.whl (43 kB)
    |████████| 43 kB 109 kB/s
Installing collected packages: pymysql
Successfully installed pymysql-1.0.2
```

- 3) Create the SingleStore Managed Service cluster if not created

Your trial has ended. Tell us how it went and what we could improve. [Send Feedback](#) or [Set up a POC](#)

SingleStore

Clusters

Contact Us [Create Cluster](#)

CLUSTERS

Cluster Overview

QUERY

SQL Editor

Visual Explain

AISHVARYA RAMAN (AISHVARYA...)

Software Licenses

Members

Billing & Usage

Settings

Sign Out

Active Recently Terminated

Try SingleStore Managed Service®

Spin-up a managed cloud database. No manual setup, no cloud provider fees, just a resilient database online in minutes.

[View QuickStart Guide](#) [Create Managed Cluster](#)

1 Create Cluster — 2 Security Settings

Create a Database Cluster

Learn more about pricing details and plan types

1 Cluster Name

Create a unique name for your cluster

2 Plan

Free Trial Credits
Test out SingleStore with free trial credits

On-Demand
Spin up a cluster of any size. Cancel at any time.

Prove-of-Concept

Summary

Cluster Name: Aishvarya Akshaya Vishva Raman's Cluster

Plan: Free Trial

Provider and Region: AWS US East 1 (N. Virginia) - 2

Cluster Size: S-00

Current balance: \$650.00

Price per Hour:

Secure this Database Cluster

Set a cluster password and specify which IP addresses can access this cluster.

① Set Cluster Password

② Security Settings

Please save this password because you will need it to login to the cluster as a user after it is deployed

Admin Password: (copy)

Generate Strong Password

② Configure Cluster Access Restriction

Access Restriction:

- Allow access from anywhere
- Only allow access from specified IP addresses Recommended

IP Allow List

| User Access | Load Data | Connect |
|--|--|---|
| Master Username: admin Password: <input type="password"/> | LOAD DATA FROM THESE SOURCES AND MORE: <ul style="list-style-type: none"> AWS S3 Bucket Microsoft Azure Blob Storage Google Cloud Storage + Load Data | CONNECT TO YOUR CLUSTER FROM: <ul style="list-style-type: none"> Endpoint CLI or SQL IDEs Your App or BI Tools Connect |

Also, set the admin password for the cluster which is mandatory for the database connectivity credentials in Flask application.

4) Create the database and the tables required in the SQL query editor of SingleStore cluster

Aishvarya Akshaya Vishva Raman's Cluster • py_db ✓

```

1  create database py_db;
2  use py_db;
3
4  CREATE TABLE `users` (
5      `userId` int(11) NOT NULL AUTO_INCREMENT,
6      `username` varchar(255) DEFAULT NULL,
7      `email` varchar(255) DEFAULT NULL,
8      `password` varchar(45) DEFAULT NULL,
9      PRIMARY KEY (`userId`)
10 );
11
12 CREATE TABLE `toDos` (
13     `toDosId` int(11) NOT NULL AUTO_INCREMENT,
14     `title` varchar(255) DEFAULT NULL,
15     `description` varchar(255) DEFAULT NULL,
16     `userId` int(11) NOT NULL,
17     PRIMARY KEY (`toDosId`),
18     KEY `fk_toDos_users_idx` (`userId`)
19 );
20
21 SELECT * FROM `users`;
22 SELECT * FROM `toDos`;

```

- 5) Copy the host and the database credentials under SQL IDE of the SingleStore cluster into the python file “server.py” and “dbconnector.py”

Connect to Your Cluster ×

CLI Client **SQL IDE** Your App BI Tools

Sequel Pro ▼

Use the connection parameters below and follow the [docs](#) instructions.

```

Host: svc-55894b31-99bb-4be2-801a-4f9a663d9580-ddl.aws-virginia-
2.svc.singlestore.com
Port: 3306
Username: admin
Password: <Admin Password>
Database: <Database Name>

```

```

1  from flask import Flask, render_template, request, session, redirect
2  from dbconnector import connectToS2MS
3  app = Flask(__name__)
4  app.secret_key = 'keep it secret, keep it safe'
5
6
7  @app.route('/')
8  def index():
9      print("in the index method")
10     return render_template('index.html')
11
12
13 @app.route('/register', methods=['POST'])
14 def register():
15     s2ms = connectToS2MS('py_db')
16     # *** REGISTRATION 1 ***
17     # Check Email To See If It Already Exists - Retrieve a User Using the Provided Email
18     query = "SELECT * FROM users WHERE email = %(email)s"
19     data = {"email": request.form['Email']}
20     UserExist = s2ms.query_db(query, data)
21
22     if UserExist == []:
23         print("NEW USER")
24         if request.form['Password'] != request.form['PasswordConfirmation']:
25             regerror = "Passwords do not match!"
26             return render_template("index.html", regerror = regerror)
27         else:
28             # Validate for empty
29             if request.form['Username'] == '' or request.form['Email'] == '' or request.form['Password'] == '':
30                 regerror = "Please complete all required fields"
31                 return render_template("index.html", regerror = regerror)

```

```

1  import pymysql.cursors
2
3  class DbConnector:
4      def __init__(self, db):
5          connection = pymysql.connect(host = 'svc-55894b31-99bb-4be2-801a-4f9a663d9580-ddl.aws-virginia-2.svc.single',
6                                      user = 'admin',
7                                      password = 'SHINchan$1234',
8                                      db = db,
9                                      charset = 'utf8mb4',
10                                     cursorclass = pymysql.cursors.DictCursor,
11                                     autocommit = True)
12
13         self.connection = connection
14
15     def query_db(self, query, data=None):
16         with self.connection.cursor() as cursor:
17             try:
18                 query = cursor.mogrify(query, data)
19                 print("Running Query:", query)
20
21                 executable = cursor.execute(query, data)
22                 if query.lower().find("insert") >= 0:
23                     # INSERT queries will return the ID NUMBER of the row inserted
24                     self.connection.commit()
25                     return cursor.lastrowid
26                 elif query.lower().find("select") >= 0:
27                     # SELECT queries will return the data from the database as a LIST OF DICTIONARIES
28                     result = cursor.fetchall()
29                     return result
30                 else:
31                     # UPDATE and DELETE queries will return nothing
32                     self.connection.commit()

```

- 6) Run “server.py” from anaconda prompt and reload the <http://127.0.0.1/5000> web page again

```
Administrator: Anaconda Prompt (anaconda3) - python server.py
(base) C:\MTech Internship\Morgan Stanley\Single Store SQL\SingleStore using Python and Flask\pyproject>python server.py
* Serving Flask app 'server' (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 349-995-757
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
in the index method
127.0.0.1 - - [05/Aug/2021 00:47:31] "GET / HTTP/1.1" 200 -
Running Query: SELECT * FROM users WHERE email = 'a.raman@gmail.com';
NEW USER
CREATING NEW USER!!!
Running Query: INSERT INTO users (username, email, password) VALUES('Aishvarya Raman', 'a.raman@gmail.com', 'SHINchan#124');
Running Query: SELECT * FROM users WHERE email= 'a.raman@gmail.com'
1
127.0.0.1 - - [05/Aug/2021 00:48:41] "POST /register HTTP/1.1" 302 -
retrieving user
Running Query: SELECT * FROM users WHERE userId = '1';
retriving todos
Running Query: SELECT * FROM toDos WHERE userId = '1';
127.0.0.1 - - [05/Aug/2021 00:48:43] "GET /dashboard HTTP/1.1" 200 -
sdasdadasd
Running Query: INSERT INTO toDos (title, description, userId) VALUES ('Buy From Shop', 'Bread\r\nCookies\r\nChocolate', 1);
127.0.0.1 - - [05/Aug/2021 00:49:57] "POST /addToDo HTTP/1.1" 302 -
retrieving user
Running Query: SELECT * FROM users WHERE userId = '1';
retriving todos
Running Query: SELECT * FROM toDos WHERE userId = '1';
127.0.0.1 - - [05/Aug/2021 00:49:59] "GET /dashboard HTTP/1.1" 200 -
```

Welcome to SingleSpace!

The place to manage your To Dos

Register

| | |
|-------------------|-------------------|
| Username: | Aishvarya Raman |
| Email: | a.raman@gmail.com |
| Password: | ***** |
| Password Confirm: | ***** |

[Register](#)

Log In

| | |
|-----------|--------------------------|
| Email: | <input type="text"/> |
| Password: | <input type="password"/> |

[Log In](#)

SingleSpace Dashboard

[Log Out](#)

Welcome, Aishvarya Raman

Let's Create some ToDos!

Title:

Description:

Bread
Cookies
Chocolate

[Add ToDo](#)

[Log Out](#)

SingleSpace Dashboard

Welcome, Aishvarya Raman

Let's Create some ToDos!

Title:

Description:

[Add ToDo](#)

Title: Buy From Shop

Description: Bread Cookies Chocolate

[Edit - Delete](#)

7) Verify the table entries in the SingleStore SQL query results

| Message Logs | | SELECT * FRO... | X | SELECT * F |
|--------------|---------------|-------------------------|--------|------------|
| Save as CSV | | | | |
| toDosId | title | description | userId | |
| 1 | Buy From Shop | Bread Cookies Chocolate | 1 | |

| Message Logs | | SELECT * FRO... | X | SELECT * F |
|--------------|-----------------|-------------------|---------|------------|
| Save as CSV | | | | |
| userId | username | email | pass... | |
| 1 | Aishvarya Ra... | a.raman@gmail.com | SHI... | |

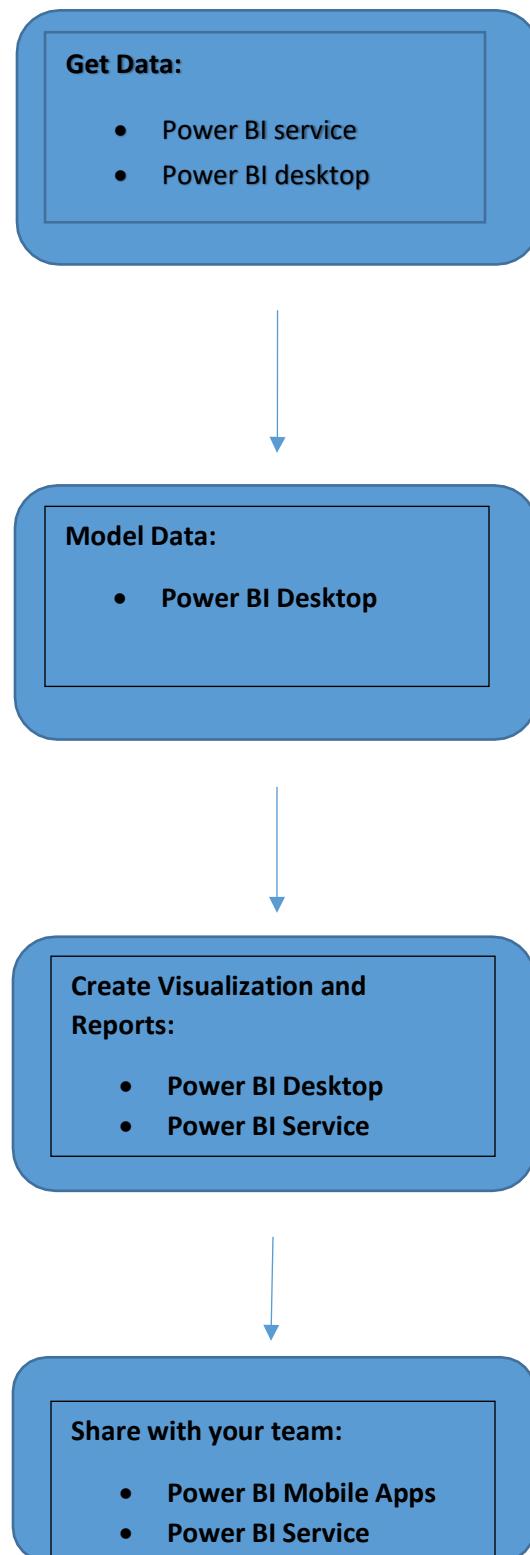
Observation/ Insights:

We can build flask applications (as well as Django framework) in python using SingleStore cluster by database connectivity.

Power Bi

Overview:

A simple understanding of business intelligence is that business intelligence is about connecting business decision making to facts about the business and its environment, to take a deep dive and understand the data underneath your business, so that you can make better business decisions.



What is Power BI

- Power BI is a business analytics service by Microsoft. It aims to provide interactive visualizations and business intelligence capabilities with an interface simple enough for end users to create their own reports and dashboards. It is part of the Microsoft Power Platform.
- Power BI provides cloud-based BI (business intelligence) services, known as "Power BI Services", along with a desktop based interface, called "Power BI Desktop". It offers data warehouse capabilities including data preparation, data discovery and interactive dashboards. In March 2016, Microsoft released an additional service called Power BI Embedded on its Azure cloud platform. One main differentiator of the product is the ability to load custom visualizations.

Features of Power BI

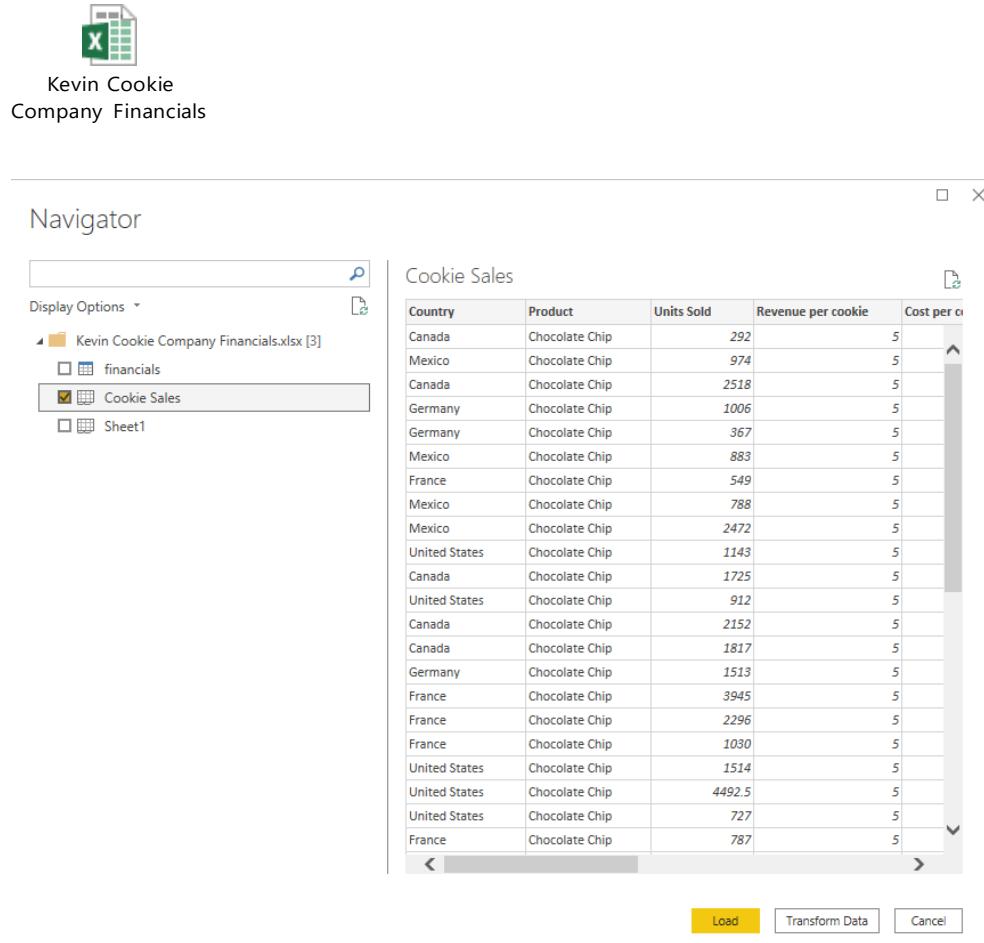
- 1) Allows to create datasets from one or multiple sources of data.
- 2) Helps to build reports such as bar-charts, slicer, geo-spatial maps, matrix, line charts and so on based on the attributes of the data.
- 3) Once can create dashboards and publish it to their respective teams.

Power BI Components

- **Power BI Desktop:** The Windows-desktop-based application for PCs and desktops, primarily for designing and publishing reports to the Service.
- **Power BI Service:** The SaaS-based (software as a service) online service. This was formerly known as Power BI for Office 365, now referred to as PowerBI.com, or simply Power BI.
- **Power BI Mobile Apps:** The Power BI Mobile apps for Android and iOS devices, as well as for Windows phones and tablets.
- **Power BI Gateway:** Gateways used to sync external data in and out of Power BI and are required for automated refreshes. In Enterprise mode, can also be used by Flows and Power Apps in Office 365.
- **Power BI Embedded:** Power BI REST API can be used to build dashboards and reports into the custom applications that serve Power BI users, as well as non-Power BI users.
- **Power BI Report Server:** An On-Premises Power BI Reporting solution for companies that won't or can't store data in the cloud-based Power BI Service.
- **Power BI Premium:** Capacity-based offering that includes flexibility to publish reports broadly across an enterprise, without requiring recipients to be licensed individually per user. Greater scale and performance than shared capacity in the Power BI service
- **Power BI Visuals Marketplace:** A marketplace of custom visuals and R-powered visuals

Getting Started with Power BI Desktop:

- 1) Download Power BI and Power BI desktop from the Microsoft store or from the link
<https://powerbi.microsoft.com/en-us/desktop/>
- 2) Consider the excel workbook to be loaded to the Power BI desktop below:



The screenshot shows the Power BI Desktop interface. On the left, the Navigator pane displays a list of files: 'Kevin Cookie Company Financials.xlsx [3]', 'financials', 'Cookie Sales' (which is selected and highlighted in yellow), and 'Sheet1'. The main area is titled 'Cookie Sales' and contains a table with the following data:

| Country | Product | Units Sold | Revenue per cookie | Cost per cookie |
|---------------|----------------|------------|--------------------|-----------------|
| Canada | Chocolate Chip | 292 | 5 | 5 |
| Mexico | Chocolate Chip | 974 | 5 | 5 |
| Canada | Chocolate Chip | 2518 | 5 | 5 |
| Germany | Chocolate Chip | 1006 | 5 | 5 |
| Germany | Chocolate Chip | 367 | 5 | 5 |
| Mexico | Chocolate Chip | 883 | 5 | 5 |
| France | Chocolate Chip | 549 | 5 | 5 |
| Mexico | Chocolate Chip | 788 | 5 | 5 |
| Mexico | Chocolate Chip | 2472 | 5 | 5 |
| United States | Chocolate Chip | 1143 | 5 | 5 |
| Canada | Chocolate Chip | 1725 | 5 | 5 |
| United States | Chocolate Chip | 912 | 5 | 5 |
| Canada | Chocolate Chip | 2152 | 5 | 5 |
| Canada | Chocolate Chip | 1817 | 5 | 5 |
| Germany | Chocolate Chip | 1513 | 5 | 5 |
| France | Chocolate Chip | 3945 | 5 | 5 |
| France | Chocolate Chip | 2296 | 5 | 5 |
| France | Chocolate Chip | 1030 | 5 | 5 |
| United States | Chocolate Chip | 1514 | 5 | 5 |
| United States | Chocolate Chip | 4492.5 | 5 | 5 |
| United States | Chocolate Chip | 727 | 5 | 5 |
| France | Chocolate Chip | 787 | 5 | 5 |

At the bottom of the interface, there are three buttons: 'Load' (highlighted in yellow), 'Transform Data', and 'Cancel'.

- 3) Click on “Transform data” for changing the column name, changing the data type of “Units Sold” from a decimal number to a whole number, or filtering out only specific countries.
- 4) For performing join operation one can load data (from excel, web...). For example we have to load the data from the web page
https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population.
Below given are the steps.

Merge

Select a table and matching columns to create a merged table.

Cookie Sales

| Country | Product | Units Sold | Revenue per cookie | Cost per cookie | Revenue | Cost | Profit | Date |
|---------|----------------|------------|--------------------|-----------------|---------|------|--------|------------|
| Canada | Chocolate Chip | 292 | 5 | 2 | 1460 | 584 | 876 | 01-01-2018 |
| Mexico | Chocolate Chip | 974 | 5 | 2 | 4870 | 1948 | 2922 | 01-01-2018 |
| Canada | Chocolate Chip | 2518 | 5 | 2 | 12590 | 5036 | 7554 | 01-01-2018 |
| Germany | Chocolate Chip | 1006 | 5 | 2 | 5030 | 2012 | 3018 | 01-01-2018 |
| France | Chocolate Chip | 1000 | 5 | 2 | 5000 | 2000 | 3000 | 01-01-2018 |

Country Populations

| Country or dependent territory | Population |
|--------------------------------|------------|
| United States | 331837048 |
| Mexico | 126014024 |
| Germany | 83190556 |
| France | 67413000 |
| Canada | 38296723 |

Join Kind

Left Outer (all from first, matching from second)

Use fuzzy matching to perform the merge

Fuzzy matching options

The selection matches 700 of 700 rows from the first table.

OK **Cancel**

The screenshot shows two overlapping dialog boxes from Microsoft Power BI.

Merge Dialog (Top):

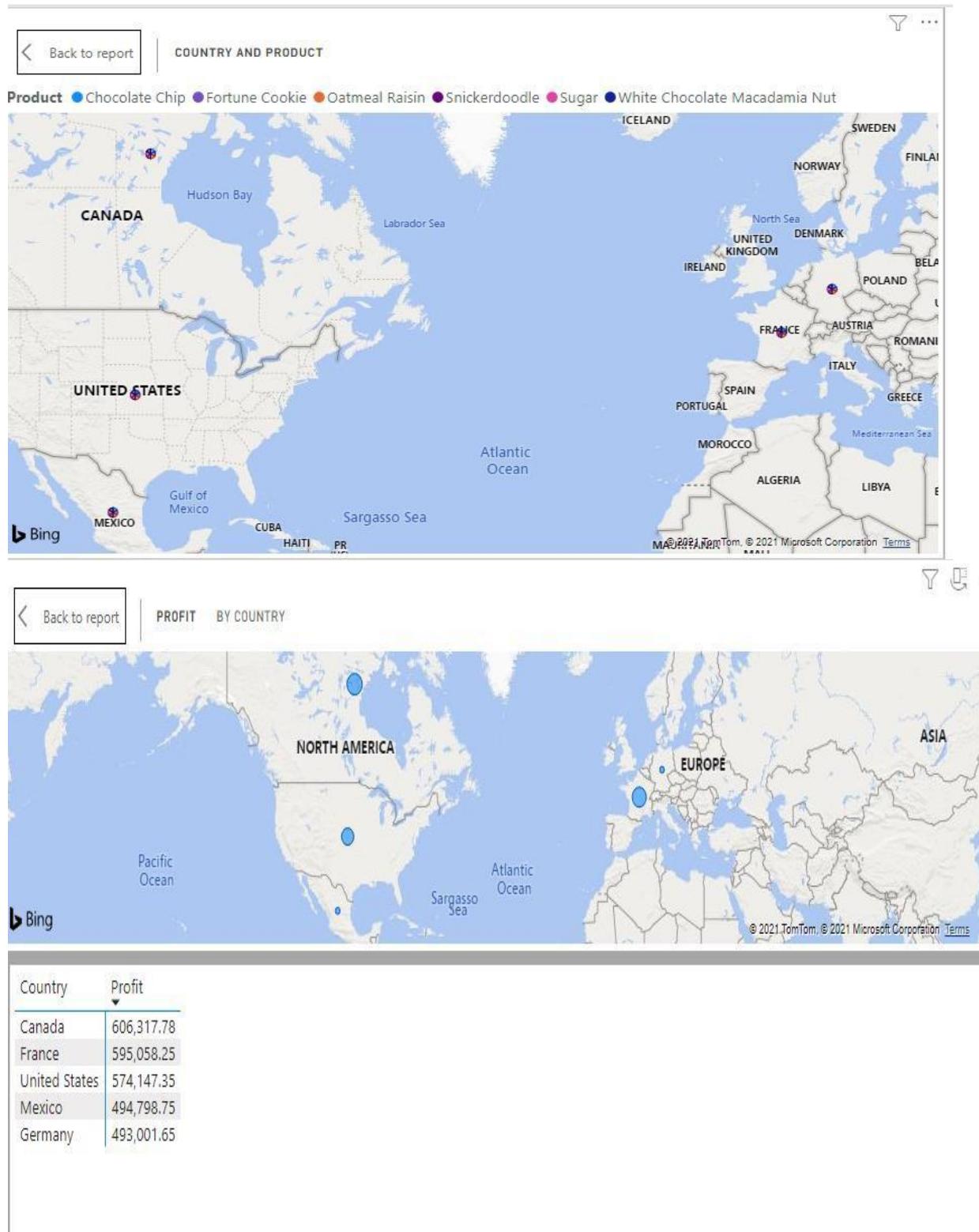
- Privacy levels:** A warning message states: "The privacy level is used to ensure data is combined without undesirable data transfer. Incorrect privacy levels may lead to sensitive data being leaked outside of a trusted scope. More information on privacy levels can be found [here](#)".
- Checkboxes:** One checkbox is checked: "Ignore Privacy Levels checks for this file. Ignoring Privacy Levels could expose sensitive or confidential data to an unauthorized person."
- Buttons:** Save and Cancel.
- Bottom Options:**
 - Left Outer (all from first, matching from second)
 - Use fuzzy matching to perform the merge
 - Fuzzy matching options
 - Estimating matches based on data previews
- Buttons:** OK and Cancel.

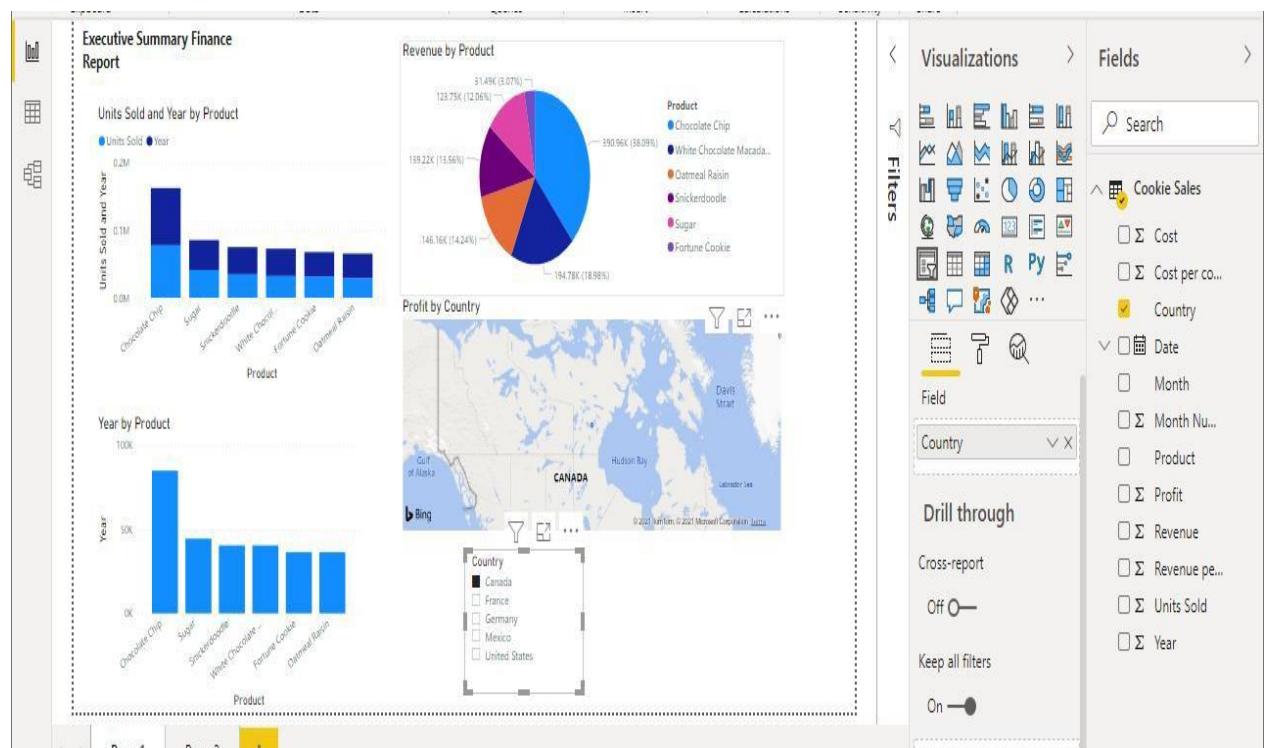
Expand Country Populations Dialog (Bottom):

- Title:** Expand Country Populations
- Text:** Select the columns to expand.
- Checklist:**
 - (Select All Columns)
 - Country or dependent territory
 - Population
- Default column name prefix (optional):** Country Populations
- Buttons:** OK and Cancel.

- 5) Create different types of reports by selecting the relevant attributes from the data: Reports in Power BI can have multiple pages. In each page there might be multiple visualization elements. Slicing and dicing, hovering and highlighting is possible in the Report. We can drill down through a hierarchical data structure, or select a particular

column in a column chart and see the related data to it in other visualization elements. All of these means Report in Power BI is for slicing and dicing. Report built for end users to play with different handles (slicers or visuals or filters), and achieve what they want. For example, they might want to understand why Sales Amount in month August is lower than other month? or Which product is selling best, and how it is distributed through branches. Here is an example of Power BI report.





Report1.pbix

- 6) Thus we have explored that we can get data, transform data, build and format visuals, save publish and share visuals. Slicer are valuable tools for filtering visuals on a report page to a specific selection. For Publishing report one can go to: Publish->My Workspace-> Share->Report.

Exploring DAX in Power BI:

- 1) DAX is the analytical language in Power BI. Learning this language, however, comes with some difficulties. Part of that challenge is the way that functions work in DAX. Learning the output of functions is an important element of learning how and where you can use them.
- 2) Consider the following excel files below which will be used to perform the functions:



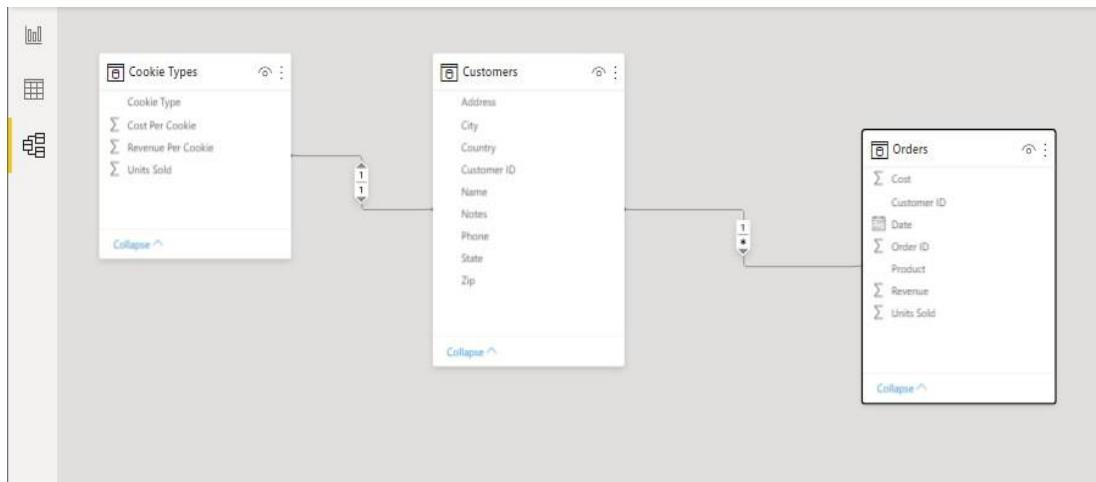
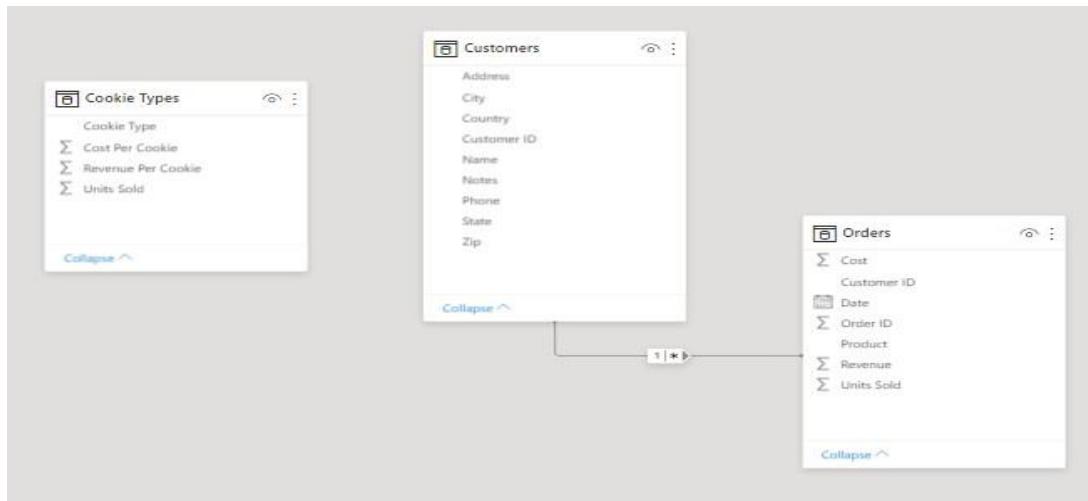
Orders.xlsx



Customers.xlsx



Cookie Types.xlsx



- 3) Scalar function in a function that returns one single value. This value can be of any data type; Date, Numeric, Text, etc. But it is always one single value. One of the most basic and simple functions in this category is SUM. Consider the measure below:

Example: Sales = Sum (FactInternetSales [Sales Amount]). This calculation will return one single value. The steps for creating a measure is given below:

The screenshot shows the Power BI Desktop interface. On the right, the 'Fields' pane is open, displaying a tree view of data sources: 'Cookie Types', 'Customers', and 'Orders'. Under 'Orders', several measures are listed: Cost, Customer ID, Date, Measure, Order ID, Product, Revenue, and Units Sold. A measure named 'Total Number of Units Sold = sum('Cookie Types'[Units Sold])' is selected. On the left, the report canvas shows a visual representation of the data flow from the 'Cookie Types' table through a 'Summarize' step to the final measure.

This screenshot shows a table visualization on the report canvas. The table has one column labeled 'Name' and another labeled 'Total Number of U...'. The data in the table is:

| Name | Total Number of U... |
|--------------------------------|----------------------|
| Home table | Orders |
| ACME Bites | 329,676.50 |
| Park & Shop Convenience Stores | 179,447.50 |
| Tres Delicious | 128,768.00 |
| Wholesome Foods | 266,690.00 |
| Total | 1,125,806.00 |

The 'Measure tools' tab is selected in the ribbon. The 'Fields' pane on the right is identical to the one in the first screenshot, showing the same hierarchy and measure definitions.

- 4) i) **COUNTX**: Counts the number of rows that contain a non-blank value or an expression that evaluates to a non-blank value, when evaluating an expression over a table.

Syntax: COUNTX(<table>, <expression>)

Return value: An integer.

ii) The COUNTX function takes two arguments. The first argument must always be a table, or any expression that returns a table. The second argument is the column or expression that is searched by COUNTX.

iii) The COUNTX function counts only values, dates, or strings. If the function finds no rows to count, it returns a blank. If you want to count logical values, use the **COUNTAX** function.

- 5) i) The **COUNTROWS** function counts the number of rows in the specified table, or in a table defined by an expression: COUNTROWS(<table>). Return value: A whole number.

ii) This function can be used to count the number of rows in a base table, but more often is used to count the number of rows that result from filtering a table, or applying context to a table. Whenever there are no rows to aggregate, the function returns a blank. However, if

there are rows, but none of them meet the specified criteria, the function returns 0.

Microsoft Excel also returns a zero if no rows are found that meet the conditions.

| 1 Count Of Orders = COUNTROWS(Orders) | | |
|---------------------------------------|--------------|-----------|
| ACME Bites | 329,676.50 | 1,125,806 |
| Park & Shop Convenience Stores | 179,447.50 | |
| Tres Delicious | 128,769.00 | |
| Wholesome Foods | 266,890.00 | |
| Total | 1,125,806.00 | |

- 6) **DISTINCT(column):** Returns a one-column table that contains the distinct values from the specified column. In other words, duplicate values are removed and only unique values are returned. The results of DISTINCT are affected by the current filter context. For example, if you use the formula in the following example to create a measure, the results would change whenever the table was filtered to show only a particular region or a time period.

File Home Help Table tools Column tools

Name: Column
Data type: Whole number

Format: \$
Summarization: Data categories

Structure Formatting

Customer ID Order ID Product Units Sold Date Revenue Cost Profit

| Customer ID | Order ID | Product | Units Sold | Date | Revenue | Cost | Profit |
|-------------|----------|----------------|------------|-------------------|---------|------|--------|
| 3 | 266868 | Chocolate Chip | 292 | 01 February 2020 | 1460 | 584 | 876 |
| 3 | 140794 | Chocolate Chip | 974 | 01 February 2020 | 4870 | 1948 | 2922 |
| 3 | 684759 | Chocolate Chip | 2518 | 01 June 2020 | 12590 | 5036 | 7554 |
| 3 | 183251 | Chocolate Chip | 1513 | 01 December 2020 | 7565 | 3026 | 4539 |
| 3 | 600124 | Chocolate Chip | 1030 | 01 May 2020 | 5150 | 2060 | 3090 |
| 3 | 562219 | Chocolate Chip | 1514 | 01 February 2020 | 7570 | 3028 | 4542 |
| 3 | 387444 | Chocolate Chip | 787 | 01 June 2020 | 3935 | 1574 | 2361 |
| 3 | 365463 | Chocolate Chip | 1728 | 01 May 2020 | 8640 | 3456 | 5184 |
| 3 | 505159 | Chocolate Chip | 2145 | 01 October 2019 | 10725 | 4290 | 6435 |
| 3 | 300303 | Chocolate Chip | 1084 | 01 December 2020 | 5420 | 2168 | 3252 |
| 3 | 578401 | Chocolate Chip | 689 | 01 June 2020 | 3445 | 1378 | 2067 |
| 3 | 365552 | Chocolate Chip | 1570 | 01 June 2020 | 7850 | 3140 | 4710 |
| 3 | 239419 | Chocolate Chip | 4251 | 01 January 2020 | 21255 | 8502 | 12753 |
| 3 | 549329 | Chocolate Chip | 2918 | 01 May 2020 | 14590 | 5836 | 8754 |
| 3 | 288851 | Chocolate Chip | 2988 | 01 July 2020 | 14940 | 5976 | 8964 |
| 3 | 255145 | Chocolate Chip | 2074 | 01 September 2020 | 10370 | 4148 | 6222 |
| 3 | 538134 | Chocolate Chip | 1514 | 01 October 2019 | 7570 | 3028 | 4542 |
| 3 | 817134 | Chocolate Chip | 274 | 01 December 2020 | 1370 | 548 | 822 |
| 3 | 697568 | Chocolate Chip | 1138 | 01 December 2020 | 5690 | 2276 | 3414 |
| 3 | 508782 | Chocolate Chip | 2992 | 01 March 2020 | 14960 | 5984 | 8976 |
| 3 | 697895 | Chocolate Chip | 1228 | 01 October 2019 | 6140 | 2456 | 3684 |

Table: Orders (700 rows) Column: Profit (657 distinct values)

1 Total Profit = SUM(Orders[Revenue]) - SUM(Orders[Cost])

| | | | | |
|-----|--|--|--------------------------|--|
| X ✓ | 1 Distinct Customers = DISTINCTCOUNT(Orders[Customer ID]) 2 // This tells me how many customers we have | ABC Grocerie ACME Bites Park & Shop Convenience Stores | 329,676.50 179,447.50 | |
| X ✓ | 1 Percentage Of Profit = [Total Profit] / SUM([Orders[Revenue]]) | | | |

7) **SUMX**: Returns the sum of an expression evaluated for each row in a table.

Syntax: SUMX(<table>, <expression>). Return value: A decimal number. Remarks: The SUMX function takes as its first argument a table, or an expression that returns a table. The second argument is a column that contains the numbers you want to sum, or an expression that evaluates to a column. Only the numbers in the column are counted. Blanks, logical values, and text are ignored.

| |
|--|
| 1 Total Profit = sumx('Cookie Types', 'Cookie Types'[Units Sold] * ('Cookie Types'[Revenue Per Cookie] - 'Cookie Types'[Cost Per Cookie])) |
|--|

8) **WEEKDAY**: Returns a number from 1 to 7 identifying the day of the week of a date.

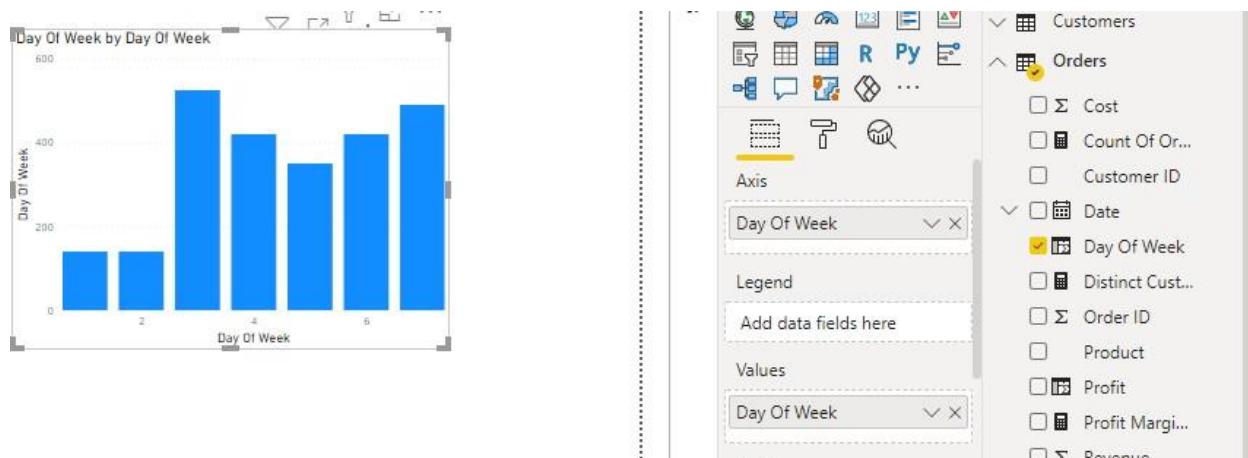
By default, the day ranges from 1 (Sunday) to 7 (Saturday). Syntax:

WEEKDAY(<date>, <return_type>). Return value: An integer number from 1 to 7.

Remarks:

- i) In contrast to Microsoft Excel, which stores dates as serial numbers, DAX works with dates and times in a datetime format. If you need to display dates as serial numbers, you can use the formatting options in Excel.
- ii) You can also type dates in an accepted text representation of a date, but to avoid unexpected results, it is best to convert the text date to a datetime format first.
- iii) When the date argument is a text representation of the date, the function uses the locale and date/time settings of the client computer to understand the text value in order to perform the conversion. If the current date/time settings represent dates in the format of Month/Day/Year, then the string, "1/8/2009", is interpreted as a datetime value equivalent to January 8th of 2009. However, if the current date/time settings represent dates in the format of Day/Month/Year, then the same string would be interpreted as a datetime value equivalent to August 1st of 2009.

| | |
|-----|--|
| X ✓ | 1 Day Of Week = WEEKDAY(Orders[Date], 1) |
|-----|--|



- 9) **FIND:** Returns the starting position of one text string within another text string. FIND is case-sensitive. `FIND(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])`. Whereas Microsoft Excel has multiple versions of the FIND function to accommodate single-byte character set (SBCS) and double-byte character set (DBCS) languages, DAX uses Unicode and counts each character the same way; therefore, you do not need to use a different version depending on the character type.

```
1 Has_Chocolate = FIND("Chocolate", Orders[Product], 1, 0)
```

The figure shows a Power BI table view. The table has columns: Customer ID, Order ID, Product, Units Sold, Date, Revenue, Cost, Profit, Day Of Week, and Has_Chocolate. The Has_Chocolate column contains values 0 or 1. A screenshot of the Power BI Fields pane is shown on the right, with the "Orders" node expanded to show fields like Cost, Count Of Orders, Customer ID, Day Of Week, Distinct Customers, Has_Chocolate, Order ID, Product, Profit, Profit Margin Per..., Revenue, and Total Number of... The "Has_Chocolate" field is highlighted.

- 10) **IF:** Checks a condition, and returns one value when it's TRUE, otherwise it returns a second value. Syntax: `IF(<logical_test>, <value_if_true>[, <value_if_false>])`

A screenshot of the Power BI formula bar. The formula is `1 Has_Chocolate = IF(FIND("Chocolate", Orders[Product], 1, 0) > 0, "Chocolate present", "No Chocolate")`. The condition part of the IF function, `FIND("Chocolate", Orders[Product], 1, 0) > 0`, is highlighted in red. The formula bar also shows a checkmark icon indicating the formula is valid.

Table: Orders (700 rows) Columns: Has_Chocolate (2 distinct values)

| Customer ID | Order ID | Product | Units Sold | Date | Revenue | Cost | Profit | Day Of Week | Has_Chocolate |
|-------------|----------|----------------|------------|-------------------|---------|------|--------|-------------|-------------------|
| 3 | 358353 | Chocolate Chip | 260 | 01 February 2020 | 1300 | 520 | 780 | 7 | Chocolate present |
| 3 | 864409 | Chocolate Chip | 2470 | 01 September 2019 | 12350 | 4940 | 7410 | 1 | Chocolate present |
| 3 | 898886 | Chocolate Chip | 2914 | 01 October 2020 | 14570 | 5828 | 8742 | 5 | Chocolate present |
| 3 | 763666 | Chocolate Chip | 1177 | 01 November 2020 | 5885 | 2354 | 3531 | 1 | Chocolate present |
| 3 | 216326 | Chocolate Chip | 1614 | 01 April 2020 | 8070 | 3228 | 4842 | 4 | Chocolate present |
| 3 | 844763 | Chocolate Chip | 2535 | 01 April 2020 | 12675 | 5070 | 7605 | 4 | Chocolate present |
| 3 | 408804 | Chocolate Chip | 2559 | 01 August 2020 | 12795 | 5118 | 7677 | 7 | Chocolate present |
| 3 | 609851 | Chocolate Chip | 267 | 01 October 2019 | 1335 | 534 | 801 | 3 | Chocolate present |
| 3 | 708450 | Chocolate Chip | 2151 | 01 November 2019 | 10755 | 4302 | 6453 | 6 | Chocolate present |
| 3 | 203604 | Fortune Cookie | 921 | 01 March 2020 | 921 | 460 | 461 | 1 | No Chocolate |
| 3 | 830805 | Fortune Cookie | 2518 | 01 June 2020 | 2518 | 1259 | 1259 | 2 | No Chocolate |
| 3 | 440487 | Fortune Cookie | 958 | 01 August 2020 | 958 | 479 | 479 | 7 | No Chocolate |
| 3 | 227728 | Fortune Cookie | 615 | 01 December 2020 | 615 | 308 | 307 | 3 | No Chocolate |
| 3 | 539666 | Fortune Cookie | 2498 | 01 September 2019 | 2498 | 1249 | 1249 | 1 | No Chocolate |
| 3 | 652401 | Fortune Cookie | 1142 | 01 June 2020 | 1142 | 571 | 571 | 2 | No Chocolate |
| 3 | 676869 | Fortune Cookie | 690 | 01 November 2020 | 690 | 345 | 345 | 1 | No Chocolate |
| 3 | 637451 | Fortune Cookie | 1287 | 01 December 2020 | 1287 | 644 | 643 | 3 | No Chocolate |
| 3 | 494228 | Fortune Cookie | 1706 | 01 December 2020 | 1706 | 853 | 853 | 3 | No Chocolate |
| 3 | 539522 | Fortune Cookie | 1859 | 01 August 2020 | 1859 | 930 | 929 | 7 | No Chocolate |
| 3 | 873031 | Fortune Cookie | 2851 | 01 October 2019 | 2851 | 1426 | 1425 | 3 | No Chocolate |
| 3 | 150704 | Fortune Cookie | 1159 | 01 October 2019 | 1159 | 580 | 579 | 3 | No Chocolate |

11) CALCULATE: CALCULATE(<expression>[, <filter1> [, <filter2> [, ...]]])

Parameters: i) expression: The expression to be evaluated.

- ii) filter1, filter2, ... (Optional) Boolean expressions or table expressions that defines filters, or filter modifier functions. The expression used as the first parameter is essentially the same as a measure.
- iii) Filters can be: Boolean filter expressions, Table filter expressions, Filter modification functions. When there are multiple filters, they can be evaluated by using the AND (&&) logical operator, meaning all conditions must be TRUE, or by the OR (||) logical operator, meaning either condition can be true.

Table: Orders (700 rows) Columns: Chocolate Chip with over 500 units (1 distinct value)

| Customer ID | Order ID | Product | Units Sold | Date | Revenue | Cost | Profit | Day Of Week | Chocolate Chip with over 500 units |
|-------------|----------|----------------|------------|-------------------|---------|------|--------|-------------|------------------------------------|
| 3 | 358353 | Chocolate Chip | 260 | 01 February 2020 | 1300 | 520 | 780 | 7 | 1 |
| 3 | 864409 | Chocolate Chip | 2470 | 01 September 2019 | 12350 | 4940 | 7410 | 1 | 1 |
| 3 | 898886 | Chocolate Chip | 2914 | 01 October 2020 | 14570 | 5828 | 8742 | 5 | 1 |
| 3 | 763666 | Chocolate Chip | 1177 | 01 November 2020 | 5885 | 2354 | 3531 | 1 | 1 |
| 3 | 216326 | Chocolate Chip | 1614 | 01 April 2020 | 8070 | 3228 | 4842 | 4 | 1 |
| 3 | 844763 | Chocolate Chip | 2535 | 01 April 2020 | 12675 | 5070 | 7605 | 4 | 1 |
| 3 | 408804 | Chocolate Chip | 2559 | 01 August 2020 | 12795 | 5118 | 7677 | 7 | 1 |
| 3 | 609851 | Chocolate Chip | 267 | 01 October 2019 | 1335 | 534 | 801 | 3 | 1 |
| 3 | 708450 | Chocolate Chip | 2151 | 01 November 2019 | 10755 | 4302 | 6453 | 6 | 1 |
| 3 | 203604 | Fortune Cookie | 921 | 01 March 2020 | 921 | 460 | 461 | 1 | 0 |
| 3 | 830805 | Fortune Cookie | 2518 | 01 June 2020 | 2518 | 1259 | 1259 | 2 | 0 |
| 3 | 440487 | Fortune Cookie | 958 | 01 August 2020 | 958 | 479 | 479 | 7 | 0 |
| 3 | 227728 | Fortune Cookie | 615 | 01 December 2020 | 615 | 308 | 307 | 3 | 0 |
| 3 | 539666 | Fortune Cookie | 2498 | 01 September 2019 | 2498 | 1249 | 1249 | 1 | 0 |
| 3 | 652401 | Fortune Cookie | 1142 | 01 June 2020 | 1142 | 571 | 571 | 2 | 0 |
| 3 | 676869 | Fortune Cookie | 690 | 01 November 2020 | 690 | 345 | 345 | 1 | 0 |
| 3 | 637451 | Fortune Cookie | 1287 | 01 December 2020 | 1287 | 644 | 643 | 3 | 0 |
| 3 | 494228 | Fortune Cookie | 1706 | 01 December 2020 | 1706 | 853 | 853 | 3 | 0 |
| 3 | 539522 | Fortune Cookie | 1859 | 01 August 2020 | 1859 | 930 | 929 | 7 | 0 |
| 3 | 873031 | Fortune Cookie | 2851 | 01 October 2019 | 2851 | 1426 | 1425 | 3 | 0 |
| 3 | 150704 | Fortune Cookie | 1159 | 01 October 2019 | 1159 | 580 | 579 | 3 | 0 |

Aggregations:

- 1) Aggregations in Power BI let you reduce table sizes so you can focus on important data and improve query performance. Aggregations enable interactive analysis over big data in ways that aren't possible otherwise, and can dramatically reduce the cost of unlocking large datasets for decision making.
- 2) Some advantages of using aggregations include:
 - i) The data you want to analyse or report on doesn't fit in memory.
 - ii) With aggregations we can load a small set of aggregated data in memory that one can update and reload every minute or so. That aggregated data can take care of the majority of queries but will still be quick to load. One has to be careful to make sure the data is in sync.

iii) Balanced architectures. The Power BI in-memory cache can handle aggregated queries, limiting queries sent in Direct Query mode and helping you meet concurrency limits. The remaining detail-level queries tend to be filtered, transactional-level queries, which data warehouses and big-data systems normally handle well.

- 3) Consider the below example performed using quick measure for finding the maximum profit date-wise:

The screenshot shows the Power BI interface with three main sections: a code editor at the top, a 'Quick measures' configuration pane in the middle, and a report view at the bottom.

Code Editor:

```

1 Profit max per Date =
2 MAXX(
3     KEEPFILTERS(VALUES('Orders'[Date])),
4     CALCULATE(SUM('Orders'[Profit]))
5 )

```

Quick measures pane:

- Calculation:** Max per category
- Base value:** Sum of Profit
- Category:** Date
- Fields:** A list of available fields from the 'Orders' table, including Profit, Date, Day Of Week, and various counts and totals.

Report View:

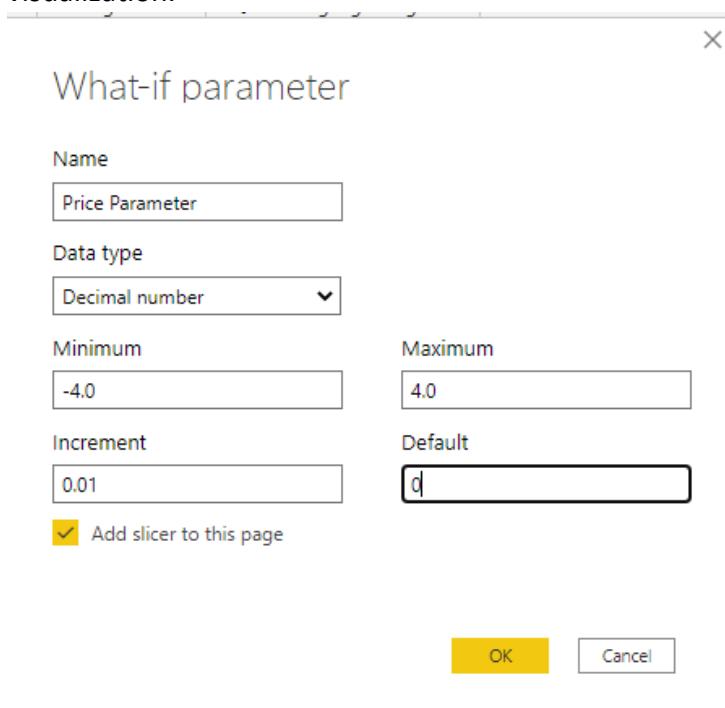
| Year | Profit max per Date | Day Of Week |
|-----------|---------------------|-------------|
| 2019 | 222,992.00 | 490 |
| Qtr 3 | 122,828.00 | 35 |
| September | 122,828.00 | 35 |
| 1 | 122,828.00 | 35 |
| Qtr 4 | 222,992.00 | 455 |
| October | 222,992.00 | 210 |
| 1 | 222,992.00 | 210 |
| November | 157,883.00 | 210 |
| 1 | 157,883.00 | 210 |
| December | 135,068.00 | 35 |
| 1 | 135,068.00 | 35 |
| 2020 | 251,918.00 | 1995 |
| Qtr 1 | 168,994.50 | 420 |
| January | 168,994.50 | 140 |
| 1 | 168,994.50 | 140 |
| February | 134,347.00 | 245 |
| March | 126,998.00 | 35 |
| Qtr 2 | 246,596.00 | 490 |
| Qtr 3 | 170,749.50 | 490 |
| Qtr 4 | 251,918.00 | 595 |
| Total | 251,918.00 | 2485 |



Understanding DAX requires you to change one's mind set from programming languages or even expression languages. the difficulty is mainly understanding the filter context. However, another part is understanding the output type of each function. One should know how and where to use tabular or scalar functions in DAX.

What-If Analysis

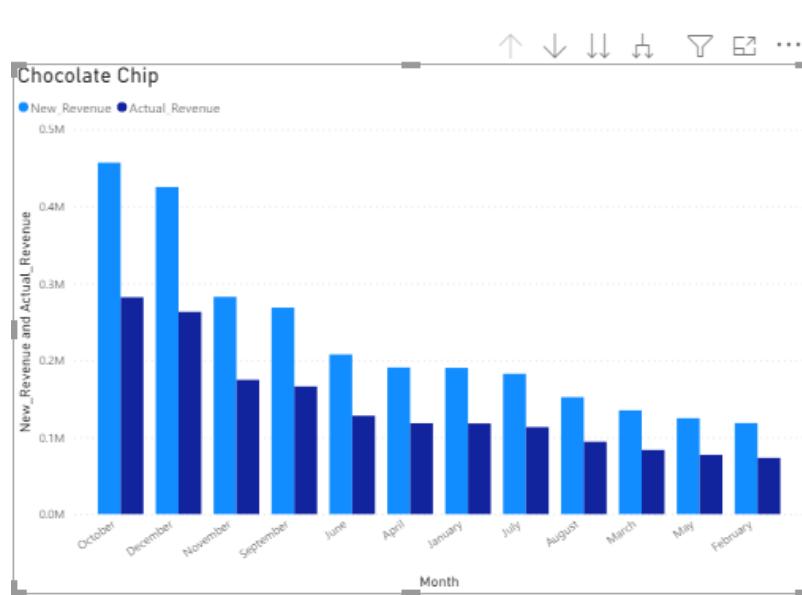
- 1) We can create what-if variables for your reports, interact with the variable as a slicer, and visualize and quantify different key values in your reports.
- 2) Consider, the Cookie sales data, we want to create a what-if variable based on the cost of each type of cookie. The slope and the cost-coefficient is also considered. We create a new measure called "New Revenue" and create a price-parameter as a what-if variable that scales from -4 to +4. Customized charts are prepared for visualization.



```
. New_Revenue = SUMX('Cookie Sales', IF('Cookie Sales'[Slope] + ((1 + 'Price Parameter'[Price Parameter Value]) * 'Cookie Sales'[Cost] * 'Cookie Sales'[Price Coefficient]) > 0, 'Cookie Sales'[Slope] + ((1 + 'Price Parameter'[Price Parameter Value]) * 'Cookie Sales'[Cost] * 'Cookie Sales'[Price Coefficient]), 0))
```

Morgan Stanley

| | Units Sold | Revenue per cookie | Cost per cookie | Revenue | Cost | Profit | Date | Month Number | Month | Year | Price Coefficient | Slope |
|----|------------|--------------------|-----------------|---------|------|---------|-------------------|--------------|-----------|------|-------------------|-------|
| 1 | 292 | 5 | 2 | 1460 | 584 | 876 | 01 February 2019 | 2 | February | 2019 | -25 | 145 |
| 2 | 974 | 5 | 2 | 4870 | 1948 | 2922 | 01 February 2019 | 2 | February | 2019 | -25 | 145 |
| 3 | 2518 | 5 | 2 | 12590 | 5036 | 7554 | 01 June 2019 | 6 | June | 2019 | -25 | 145 |
| 4 | 1006 | 5 | 2 | 5030 | 2012 | 3018 | 01 June 2019 | 6 | June | 2019 | -25 | 145 |
| 5 | 367 | 5 | 2 | 1835 | 734 | 1101 | 01 July 2019 | 7 | July | 2019 | -25 | 145 |
| 6 | 883 | 5 | 2 | 4415 | 1766 | 2649 | 01 August 2019 | 8 | August | 2019 | -25 | 145 |
| 7 | 549 | 5 | 2 | 2745 | 1098 | 1647 | 01 September 2018 | 9 | September | 2018 | -25 | 145 |
| 8 | 788 | 5 | 2 | 3940 | 1576 | 2364 | 01 September 2018 | 9 | September | 2018 | -25 | 145 |
| 9 | 2472 | 5 | 2 | 12360 | 4944 | 7416 | 01 September 2019 | 9 | September | 2019 | -25 | 145 |
| 10 | 1143 | 5 | 2 | 5715 | 2286 | 3429 | 01 October 2019 | 10 | October | 2019 | -25 | 145 |
| 11 | 1725 | 5 | 2 | 8625 | 3450 | 5175 | 01 November 2018 | 11 | November | 2018 | -25 | 145 |
| 12 | 912 | 5 | 2 | 4560 | 1824 | 2736 | 01 November 2018 | 11 | November | 2018 | -25 | 145 |
| 13 | 2152 | 5 | 2 | 10760 | 4304 | 6456 | 01 December 2018 | 12 | December | 2018 | -25 | 145 |
| 14 | 1817 | 5 | 2 | 9085 | 3634 | 5451 | 01 December 2019 | 12 | December | 2019 | -25 | 145 |
| 15 | 1513 | 5 | 2 | 7565 | 3026 | 4539 | 01 December 2019 | 12 | December | 2019 | -25 | 145 |
| 16 | 3945 | 5 | 2 | 19725 | 7890 | 11835 | 01 January 2019 | 1 | January | 2019 | -25 | 145 |
| 17 | 2296 | 5 | 2 | 11480 | 4592 | 6888 | 01 February 2019 | 2 | February | 2019 | -25 | 145 |
| 18 | 1030 | 5 | 2 | 5150 | 2060 | 3090 | 01 May 2019 | 5 | May | 2019 | -25 | 145 |
| 19 | 1514 | 5 | 2 | 7570 | 3028 | 4542 | 01 February 2019 | 2 | February | 2019 | -25 | 145 |
| 20 | 4492 | 5 | 2 | 22462.5 | 8985 | 13477.5 | 01 April 2019 | 4 | April | 2019 | -25 | 145 |



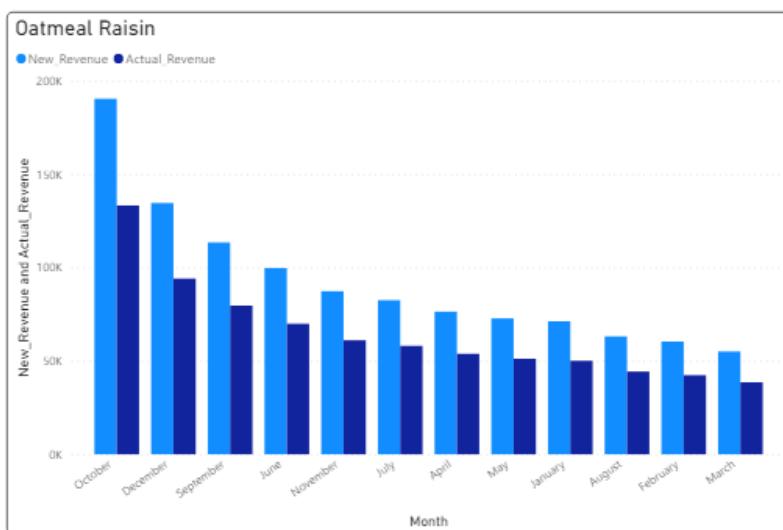
Price Parameter

-1.16



Product

- Chocolate Chip
- Fortune Cookie
- Oatmeal Raisin
- Snickerdoodle
- Sugar
- White Chocolate Macadamia Nut



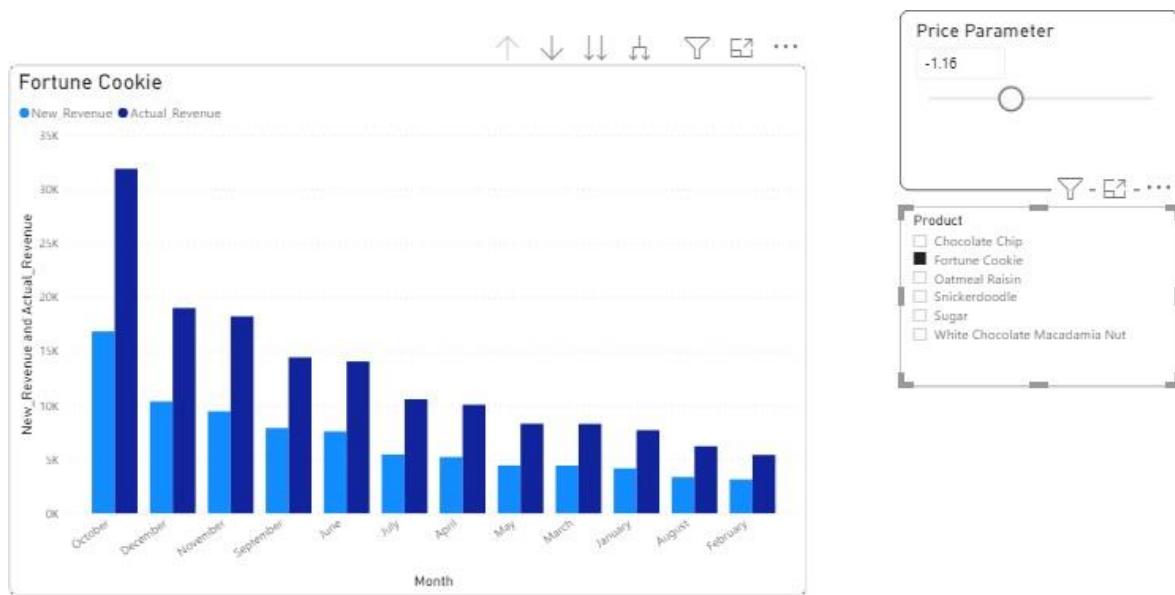
Price Parameter

-1.16



Product

- Oatmeal Raisin
- Fortune Cookie
- Chocolate Chip
- Snickerdoodle
- Sugar
- White Chocolate Macadamia Nut



Thus using What-If analysis we compared the predicted revenue vs. the actual revenue of each cookie product.

Problem Statement: Based on the dataset of Covid-19 data (downloaded from Kaggle) prepare a dashboard in Power Bi

Functionalities added for the dashboard:

- 1) Downloaded two datasets from Kaggle given below:
 - i) Covid Vaccination Details Country-wise
 - ii) Vaccine Manufacture details



country_vaccination
s.csv



country_vaccination
s_by_manufacturer.c

- 2) Filtered the columns in both the datasets and created a dimension table called "Vaccine" and assigned unique ID to it using "Index Column" option in Power BI.

Morgan Stanley

Screenshot of Power BI Data Editor showing two versions of a 'country_vaccinations' query.

Top Screenshot (Initial State):

- Properties:** Name: country_vaccinations
- Applied Steps:** Filtered Rows

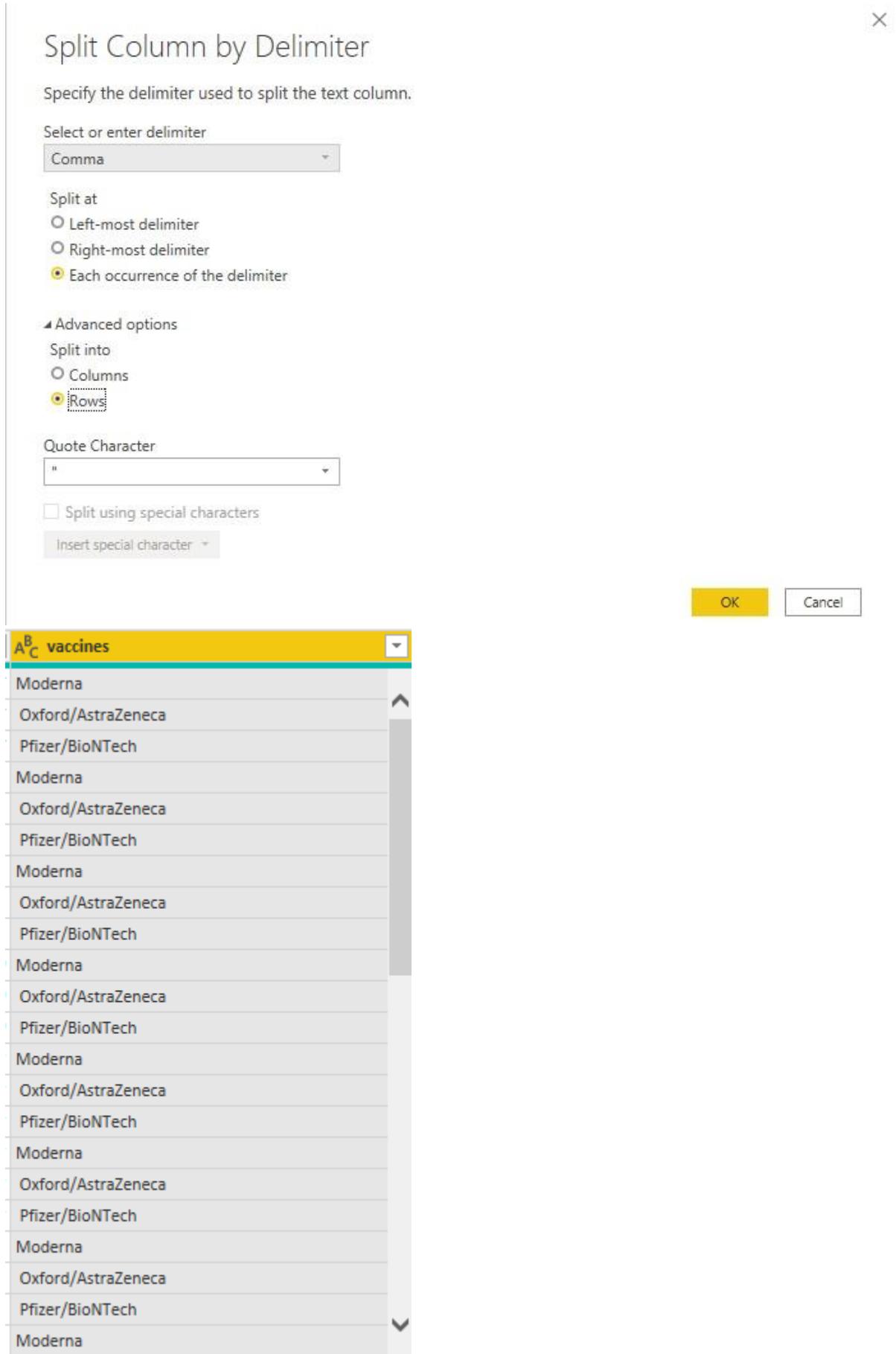
| | A _C country | A _C iso_code | date | i ² ₃ total_vaccinations | i ² ₃ people_vaccinated |
|----|------------------------|-------------------------|------------|--|---|
| 1 | Canada | CAN | 14/12/2020 | 5 | |
| 2 | Canada | CAN | 15/12/2020 | 723 | |
| 3 | Canada | CAN | 16/12/2020 | 3023 | |
| 4 | Canada | CAN | 17/12/2020 | 7202 | |
| 5 | Canada | CAN | 18/12/2020 | 11174 | 11 |
| 6 | Canada | CAN | 19/12/2020 | 11894 | 11 |
| 7 | Canada | CAN | 20/12/2020 | 12738 | 12 |
| 8 | Canada | CAN | 21/12/2020 | 20866 | 26 |
| 9 | Canada | CAN | 22/12/2020 | 26603 | 26 |
| 10 | Canada | CAN | 23/12/2020 | 35089 | 35 |
| 11 | Canada | CAN | 24/12/2020 | 45398 | 45 |
| 12 | Canada | CAN | 25/12/2020 | 45398 | 45 |
| 13 | Canada | CAN | 26/12/2020 | 48417 | 46 |
| 14 | Canada | CAN | 27/12/2020 | 52280 | 52 |
| 15 | Canada | CAN | 28/12/2020 | 59449 | 55 |
| 16 | Canada | CAN | 29/12/2020 | 70463 | 76 |
| 17 | Canada | CAN | 30/12/2020 | 84142 | 84 |
| 18 | Canada | CAN | 31/12/2020 | 97582 | 97 |
| 19 | Canada | CAN | 01/01/2021 | 103790 | 103 |
| 20 | Canada | CAN | 02/01/2021 | 108732 | 108 |
| 21 | Canada | CAN | 03/01/2021 | 114034 | 114 |
| 22 | Canada | CAN | 04/01/2021 | 137735 | 137 |
| 23 | | | | | |

Bottom Screenshot (After Transformation):

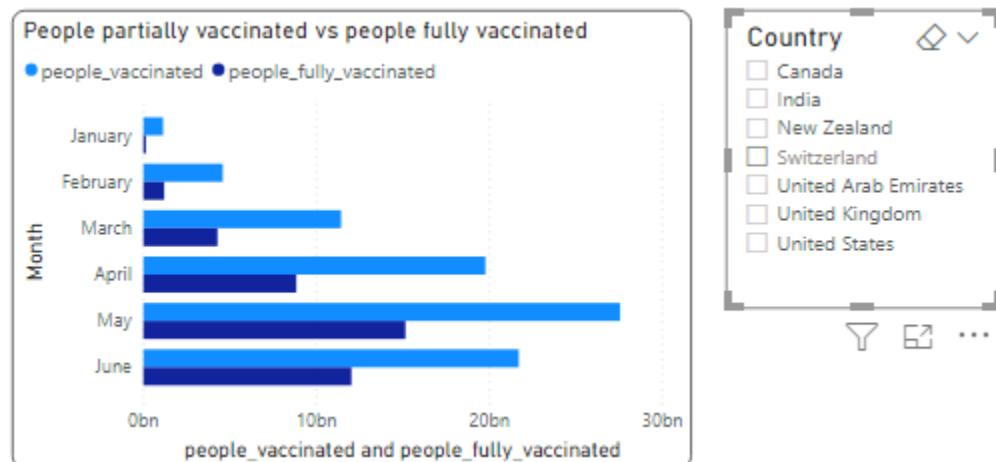
- Properties:** Name: country_vaccinations
- Applied Steps:** Filtered Rows, Removed Columns

| | A _C country | date | i ² ₃ total_vaccinations | i ² ₃ people_vaccinated | i ² ₃ people_fully_vaccinated |
|----|------------------------|------------|--|---|---|
| 1 | Canada | 06/01/2021 | 183360 | 182930 | |
| 2 | Canada | 07/01/2021 | 219446 | 218359 | |
| 3 | Canada | 08/01/2021 | 260654 | 258627 | |
| 4 | Canada | 09/01/2021 | 296241 | 293301 | |
| 5 | Canada | 10/01/2021 | 319603 | 316661 | |
| 6 | Canada | 11/01/2021 | 359049 | 356107 | |
| 7 | Canada | 12/01/2021 | 388493 | 385470 | |
| 8 | Canada | 13/01/2021 | 419209 | 407278 | |
| 9 | Canada | 14/01/2021 | 459492 | 445000 | |
| 10 | Canada | 15/01/2021 | 507687 | 489693 | |
| 11 | Canada | 16/01/2021 | 543291 | 524177 | |
| 12 | Canada | 17/01/2021 | 570742 | 550438 | |
| 13 | Canada | 18/01/2021 | 614656 | 593632 | |
| 14 | Canada | 19/01/2021 | 653653 | 629268 | |
| 15 | Canada | 20/01/2021 | 696242 | 659194 | |
| 16 | Canada | 21/01/2021 | 738864 | 695612 | |
| 17 | Canada | 22/01/2021 | 776606 | 727115 | |
| 18 | Canada | 23/01/2021 | 801238 | 747434 | |
| 19 | Canada | 24/01/2021 | 816451 | 760010 | |
| 20 | Canada | 25/01/2021 | 839949 | 775296 | |

- 3) Once can split one row of a column from multiple options to single option for each row value



Number of vaccinations = `sum(country_vaccinations[total_vaccinations])`

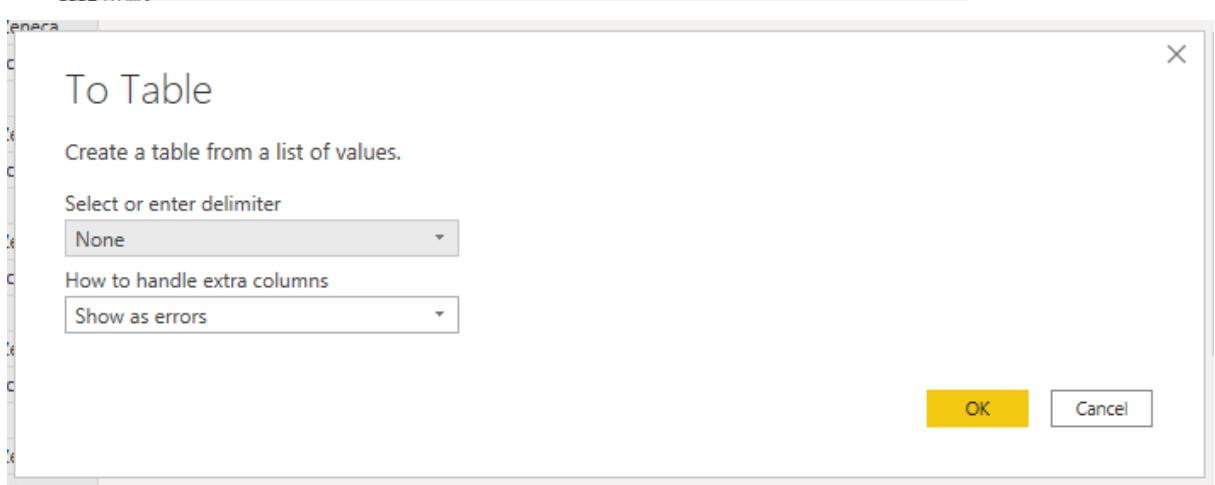


4) For checking which values are duplicates.

| accinated_per_hundred | 1.2 people_fully_vaccinated_per_hundred | 123 daily_vaccinations_per_million | A% vaccines |
|-----------------------|---|------------------------------------|------------------------|
| 18 unique | 346 distinct, 45 unique | 443 distinct, 88 unique | 4 distinct, 0 unique |
| 1 | 0.48 | 0 | 376 Moderna |
| 2 | 0.48 | 0 | 376 Oxford/AstraZeneca |
| 3 | 0.48 | 0 | 376 Pfizer/BioNTech |
| 4 | 0.58 | 0 | 461 Moderna |
| 5 | 0.58 | 0 | 461 Oxford/AstraZeneca |
| 6 | 0.58 | 0 | 461 Pfizer/BioNTech |
| 7 | 0.69 | 0.01 | 594 Moderna |
| 8 | 0.69 | 0.01 | 594 Oxford/AstraZeneca |
| 9 | 0.69 | 0.01 | 594 Pfizer/BioNTech |
| 10 | 0.78 | 0.01 | 710 Moderna |
| 11 | 0.78 | 0.01 | 710 Oxford/AstraZeneca |
| 12 | 0.78 | 0.01 | 710 Pfizer/BioNTech |
| 13 | 0.84 | 0.01 | 778 Moderna |
| 14 | 0.84 | 0.01 | 778 Oxford/AstraZeneca |
| 15 | 0.84 | 0.01 | 778 Pfizer/BioNTech |
| 16 | 0.94 | 0.01 | 838 Moderna |

- 5) Create a new table “Vaccine” which will be used as a new dimension table for the remaining two fact tables. The list is converted to a table and unique ID is assigned.

| | List |
|----|--------------------|
| 1 | Moderna |
| 2 | Oxford/AstraZeneca |
| 3 | Pfizer/BioNTech |
| 4 | Moderna |
| 5 | Oxford/AstraZeneca |
| 6 | Pfizer/BioNTech |
| 7 | Moderna |
| 8 | Oxford/AstraZeneca |
| 9 | Pfizer/BioNTech |
| 10 | Moderna |
| 11 | Oxford/AstraZeneca |
| 12 | Pfizer/BioNTech |
| 13 | Moderna |
| 14 | Oxford/AstraZeneca |
| 15 | Pfizer/BioNTech |
| 16 | Moderna |
| 17 | Oxford/AstraZeneca |
| 18 | Pfizer/BioNTech |
| 19 | Moderna |
| 20 | Oxford/AstraZeneca |
| 21 | Pfizer/BioNTech |
| 22 | Moderna |
| 23 | Oxford/AstraZeneca |



- 6) Merge the Vaccine table with the other two tables.

Queries [3]

- country_vaccinations
- country_vaccinations_by...
- vaccines

| | Vaccine Names |
|---|--------------------|
| 1 | Moderna |
| 2 | Oxford/AstraZeneca |
| 3 | Pfizer/BioNTech |
| 4 | Covaxin |
| 5 | Sinopharm/Beijing |
| 6 | Sinopharm/Wuhan |
| 7 | Sputnik V |
| 8 | Johnson&Johnson |

Queries [3]

- country_vaccinations
- country_vaccinations_by...
- vaccines

| | Vaccine Names |
|---|--------------------|
| 1 | Moderna |
| 2 | Oxford/AstraZeneca |
| 3 | Pfizer/BioNTech |
| 4 | Covaxin |
| 5 | Sinopharm/Beijing |
| 6 | Sinopharm/Wuhan |
| 7 | Sputnik V |
| 8 | Johnson&Johnson |

The screenshot shows the Microsoft Power BI Data Studio interface with a 'Merge' dialog open. At the top, the Power BI ribbon is visible with various tabs like File, Home, Transform, Add Column, View, Tools, and Help. The 'Transform' tab is selected.

Queries [3]

- country_vaccinations**: Contains 4 rows of data for Canada with columns: date, total_vaccinations, people_vaccinated, and people_fully_vaccinated.
- country_vaccinations_by...**
- vaccines**

Properties pane shows the table name is set to 'country_vaccinations'.

Merge Dialog

Select a table and matching columns to create a merged table.

country_vaccinations table preview:

| coated_per_hundred | people_fully_vaccinated_per_hundred | daily_vaccinations_per_million | vaccines |
|--------------------|-------------------------------------|--------------------------------|--------------------|
| 0.48 | 0 | 376 | Moderna |
| 0.48 | 0 | 376 | Oxford/AstraZeneca |
| 0.48 | 0 | 376 | Pfizer/BioNTech |
| 0.58 | 0 | 461 | Moderna |
| 0.58 | 0 | 461 | Covaxin |

vaccines table preview:

| VaccineId | Vaccine Names |
|-----------|--------------------|
| 1 | Moderna |
| 2 | Oxford/AstraZeneca |
| 3 | Pfizer/BioNTech |
| 4 | Covaxin |
| 5 | Sinopharm/Beijing |

Join Kind: Left Outer (all from first, matching from second)

Use fuzzy matching to perform the merge

Fuzzy matching options

Feedback: ✓ The selection matches 2027 of 2027 rows from the first table.

Buttons: OK, Cancel

Merge

Select a table and matching columns to create a merged table.

country_vaccinations_by_manufacturer



| location | date | vaccine | total_vaccinations |
|----------|------------|--------------------|--------------------|
| Austria | 05/01/2021 | Johnson&Johnson | 0 |
| Austria | 05/01/2021 | Moderna | 0 |
| Austria | 05/01/2021 | Oxford/AstraZeneca | 0 |
| Austria | 05/01/2021 | Pfizer/BioNTech | 30885 |
| Austria | 12/01/2021 | Johnson&Johnson | 0 |

| vaccines | |
|-----------|--------------------|
| VaccineId | Vaccine Names |
| 1 | Moderna |
| 2 | Oxford/AstraZeneca |
| 3 | Pfizer/BioNTech |
| 4 | Covaxin |
| 5 | Sinopharm/Beijing |

Join Kind

Left Outer (all from first, matching from second)

Use fuzzy matching to perform the merge

Fuzzy matching options

✓ The selection matches 7246 of 7246 rows from the first table.

OK

Cancel

Queries [3]

| | r2 people_fully_vaccinated_per_hundred | r2 daily_vaccinations_per_million | A ^B vaccines | r2 _C VaccineId |
|----|--|-----------------------------------|-------------------------|---------------------------|
| 1 | 0 | | 376 Moderna | 1 |
| 2 | 0 | | 461 Moderna | 1 |
| 3 | 0.01 | | 594 Moderna | 1 |
| 4 | 0 | | 376 Oxford/AstraZeneca | 2 |
| 5 | 0 | | 461 Oxford/AstraZeneca | 2 |
| 6 | 0.01 | | 594 Oxford/AstraZeneca | 2 |
| 7 | 0 | | 376 Pfizer/BioNTech | 3 |
| 8 | 0 | | 461 Pfizer/BioNTech | 3 |
| 9 | 0.01 | | 594 Pfizer/BioNTech | 3 |
| 10 | 0.01 | | 710 Moderna | 1 |
| 11 | 0.01 | | 710 Oxford/AstraZeneca | 2 |
| 12 | 0.01 | | 710 Pfizer/BioNTech | 3 |
| 13 | 0.01 | | 778 Moderna | 1 |
| 14 | 0.01 | | 778 Oxford/AstraZeneca | 2 |
| 15 | 0.01 | | 778 Pfizer/BioNTech | 3 |
| 16 | 0.01 | | 838 Moderna | 1 |
| 17 | 0.01 | | 838 Oxford/AstraZeneca | 2 |
| 18 | 0.01 | | 838 Pfizer/BioNTech | 3 |
| 19 | 0.01 | | 883 Moderna | 1 |
| 20 | 0.01 | | 883 Oxford/AstraZeneca | 2 |
| 21 | 0.01 | | 883 Pfizer/BioNTech | 3 |
| 22 | 0.03 | | 893 Moderna | 1 |
| 23 | | | | |

13 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows

Query Settings

PROPERTIES

Name: country_vaccinations

All Properties

APPLIED STEPS

- Filtered Rows
- Filtered Rows1
- Filtered Rows2
- Filtered Rows3
- Filtered Rows4
- Filtered Rows5
- Filtered Rows6
- Filtered Rows7
- Removed Columns
- Split Column by Delimiter
- Changed Type1
- Filtered Rows8
- Trimmed Text
- Filtered Rows9
- Merged Queries
- Expanded vaccines.1

PREVIEW DOWNLOADED AT 23:05

Morgan Stanley

Queries [3]

| | location | date | A ^B vaccine | 1 ² 3 total_vaccinations | 1 ² 3 Vaccineld |
|----|----------|------------|------------------------|-------------------------------------|----------------------------|
| 1 | ria | 05/01/2021 | Johnson&Johnson | 0 | 8 |
| 2 | ria | 12/01/2021 | Johnson&Johnson | 0 | 8 |
| 3 | ria | 19/01/2021 | Johnson&Johnson | 0 | 8 |
| 4 | ria | 05/01/2021 | Moderna | 0 | 1 |
| 5 | ria | 12/01/2021 | Moderna | 88 | 1 |
| 6 | ria | 05/01/2021 | Oxford/AstraZeneca | 0 | 2 |
| 7 | ria | 12/01/2021 | Oxford/AstraZeneca | 0 | 2 |
| 8 | ria | 05/01/2021 | Pfizer/BioNTech | 30885 | 3 |
| 9 | ria | 12/01/2021 | Pfizer/BioNTech | 115203 | 3 |
| 10 | ria | 19/01/2021 | Moderna | 298 | 1 |
| 11 | ria | 19/01/2021 | Oxford/AstraZeneca | 0 | 2 |
| 12 | ria | 19/01/2021 | Pfizer/BioNTech | 212305 | 3 |
| 13 | ria | 26/01/2021 | Johnson&Johnson | 0 | 8 |
| 14 | ria | 26/01/2021 | Moderna | 4151 | 1 |
| 15 | ria | 26/01/2021 | Oxford/AstraZeneca | 0 | 2 |
| 16 | ria | 26/01/2021 | Pfizer/BioNTech | 256675 | 3 |
| 17 | ria | 02/02/2021 | Johnson&Johnson | 0 | 8 |
| 18 | ria | 02/02/2021 | Moderna | 5830 | 1 |
| 19 | ria | 02/02/2021 | Oxford/AstraZeneca | 1 | 2 |
| 20 | ria | 02/02/2021 | Pfizer/BioNTech | 351951 | 3 |
| 21 | ria | 09/02/2021 | Johnson&Johnson | 0 | 8 |
| 22 | ria | 09/02/2021 | Moderna | 9891 | 1 |
| 23 | | | | | |

5 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows.

File Home Transform Add Column View Tools Help

Close & Apply New Source Recent Sources Enter Data Data source settings Manage Parameters Refresh Preview Advanced Query

Close New Query Data Sources Parameters Queries

Queries [3]

| | A ^B country | date |
|----|------------------------|------|
| 1 | Canada | |
| 2 | Canada | |
| 3 | Canada | |
| 4 | Canada | |
| 5 | Canada | |
| 6 | Canada | |
| 7 | Canada | |
| 8 | Canada | |
| 9 | Canada | |
| 10 | Canada | |
| 11 | Canada | |
| 12 | Canada | |
| 13 | Canada | |
| 14 | Canada | |
| 15 | Canada | |
| 16 | Canada | |
| 17 | Canada | |
| 18 | Canada | |
| 19 | Canada | |
| 20 | Canada | |
| 21 | Canada | |
| 22 | Canada | |
| 23 | | |

Query Settings X

PROPERTIES Name country_vaccinations_by_manufacturer All Properties

APPLIED STEPS Source Promoted Headers Changed Type Filtered Rows Trimmed Text Filtered Rows1 Merged Queries Expanded vaccines

PREVIEW DOWNLOADED AT 23:07 X

Choose Columns

Choose the columns to keep

Search Columns A Z

(Select All Columns)

country

date

total_vaccinations

people_vaccinated

people_fully_vaccinated

daily_vaccinations_raw

daily_vaccinations

total_vaccinations_per_hundred

people_vaccinated_per_hundred

people_fully_vaccinated_per_hundred

daily_vaccinations_per_million

vaccines

Vaccineld

OK Cancel

The screenshot shows the Power BI Data Editor interface. On the left, the 'Queries [3]' pane lists three queries: 'country_vaccinations', 'country_vaccinations_by...', and 'vaccines'. The 'country_vaccinations' query is currently selected, displaying a table with columns 'location' and 'date', and rows showing repeated entries for 'Austria'. A 'Choose Columns' dialog is open over the table, titled 'Choose Columns'. It asks 'Choose the columns to keep' and contains a 'Search Columns' input field and a list of columns with checkboxes. The checked columns are 'location', 'date', 'total_vaccinations', and 'VaccineId'. Below the dialog is a data model diagram showing relationships between 'country_vaccinations', 'vaccines', and 'country_vaccinations_by...'. The 'vaccines' table has columns 'Vaccine Names' and 'VaccineId'. The 'country_vaccinations' table has columns 'country', 'daily_vaccinations', 'daily_vaccinations_per_million', 'daily_vaccinations_raw', 'date', 'people_fully_vaccinated', 'people_fully_vaccinated_per_hundred', 'people_vaccinated', 'people_vaccinated_per_hundred', 'total_vaccinations', and 'total_vaccinations_per_hundred'. The 'country_vaccinations_by...' table has columns 'date', 'location', 'total_vaccinations', and 'VaccineId'.

7) Using advanced DAX queries create new measures

The screenshot shows the Power BI Data Model ribbon with two DAX measure definitions listed:

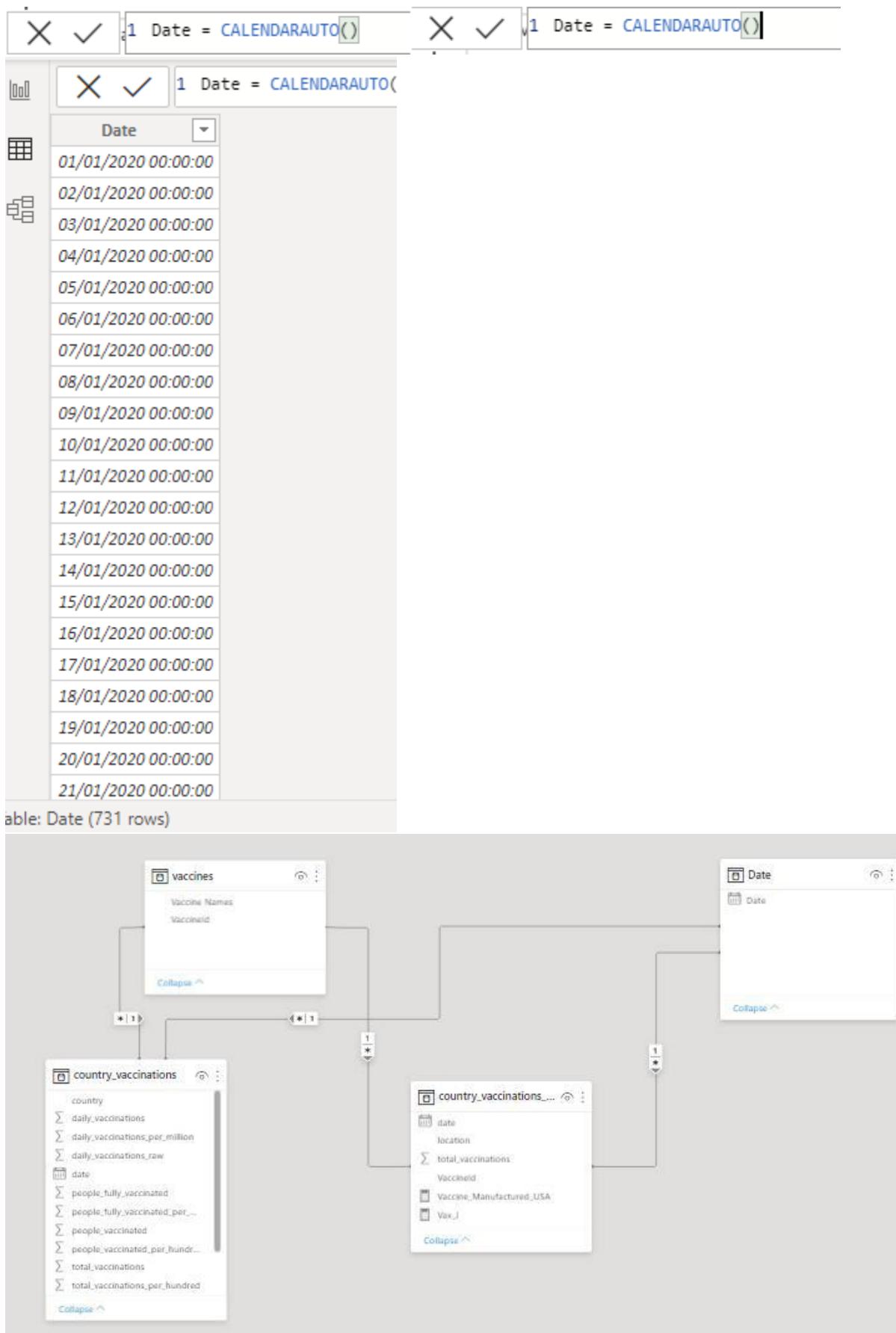
- 1 Vaccine_Manufactured_USA = `SUMX(FILTER(country_vaccinations_by_manufacturer, country_vaccinations_by_manufacturer[location] = "United States"), country_vaccinations_by_manufacturer[total_vaccinations])`
- 1 Vaccine_Given_USA = `SUMX(FILTER(country_vaccinations, country_vaccinations[country] = "United States"), country_vaccinations[total_vaccinations])`

```
X ✓ | 1 Vaccine_Given_USA = SUMX(FILTER(country_vaccinations, country_vaccinations[country] = "United States"),  
country_vaccinations[people_fully_vaccinated])  
⋮  
X ✓ | 1 Vax_J = SUMX(FILTER(country_vaccinations_by_manufacturer, country_vaccinations_by_manufacturer[VaccineId] = 8)  
, country_vaccinations_by_manufacturer[total_vaccinations])  
⋮ | vaccine
```

- 8) Created custom visuals using the new created measures



- 9) Created a Date table using CALENDARAUTO () function that lists the date for the day-to-day vaccinations.



```

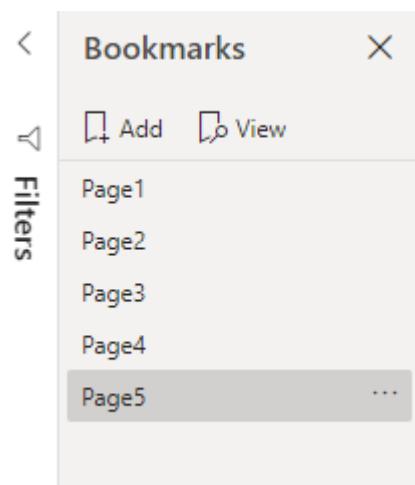
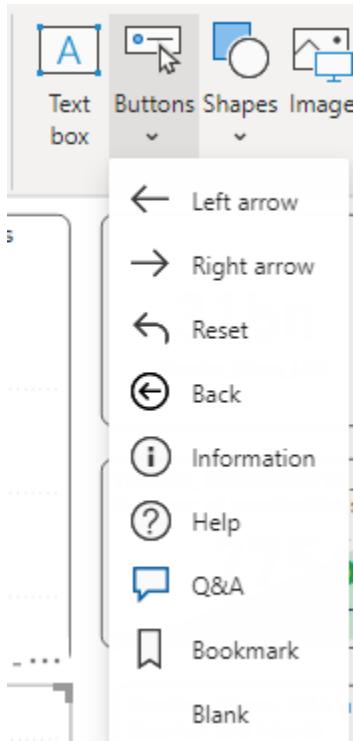
X ✓ 1 given vs manufactured = DIVIDE(country_vaccinations[Number of vaccinations],  
country_vaccinations_by_manufacturer[Vaccine_Manufactured_Total])  

X ✓ 1 Measure = CALCULATE(SUM(country_vaccinations[people_fully_vaccinated]), ALLEXCEPT(country_vaccinations,country_vaccinations[country],  
country_vaccinations[date]))  

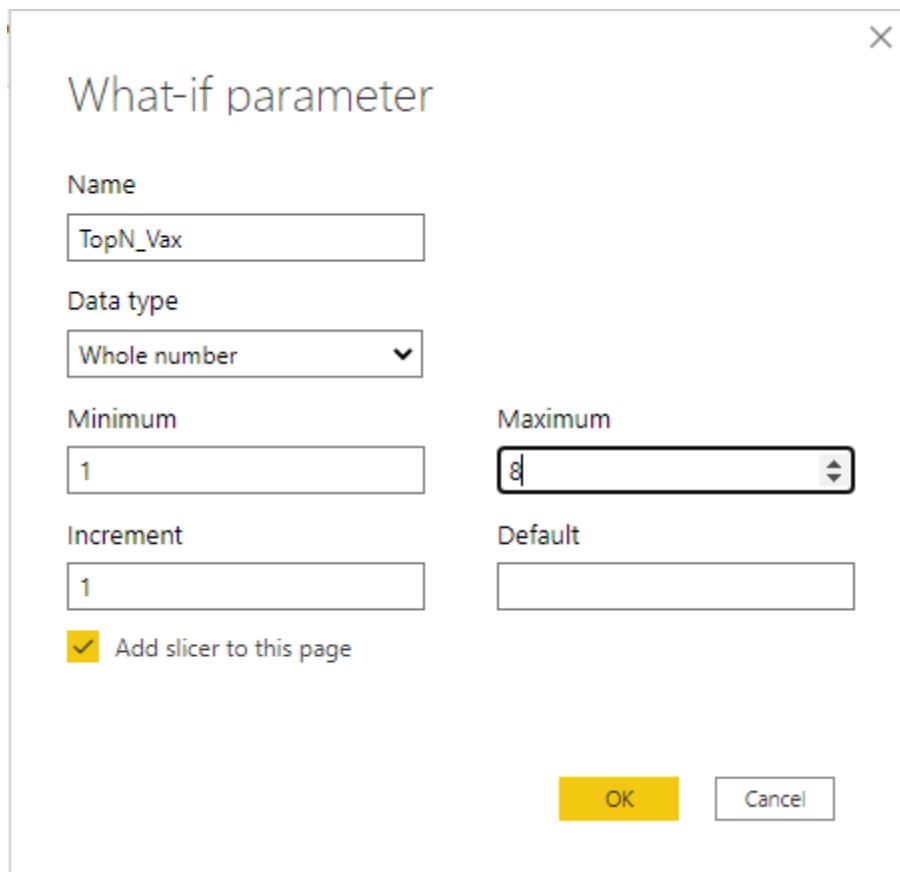
X ✓ 1 Measure_Final = MAX(ALLEXCEPT(country_vaccinations, country_vaccinations[country]), [Measure1])

```

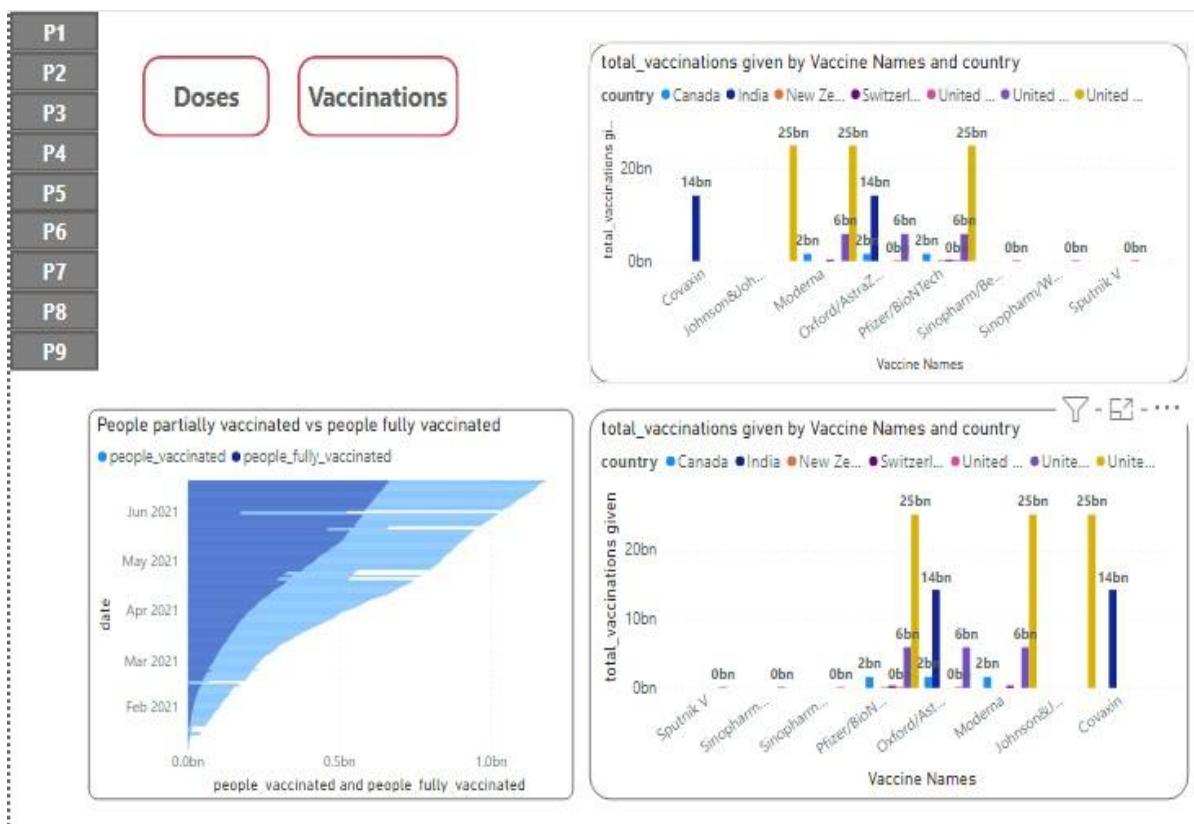
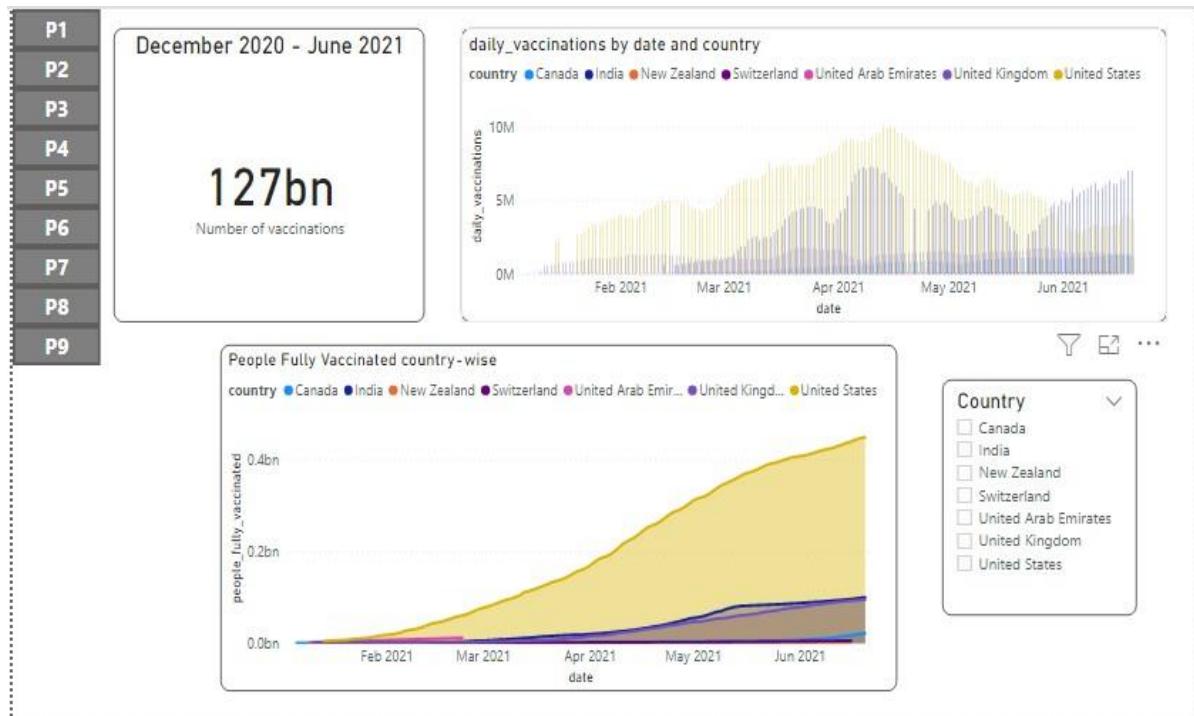
10) Created custom bookmarks and buttons for navigation between the pages.

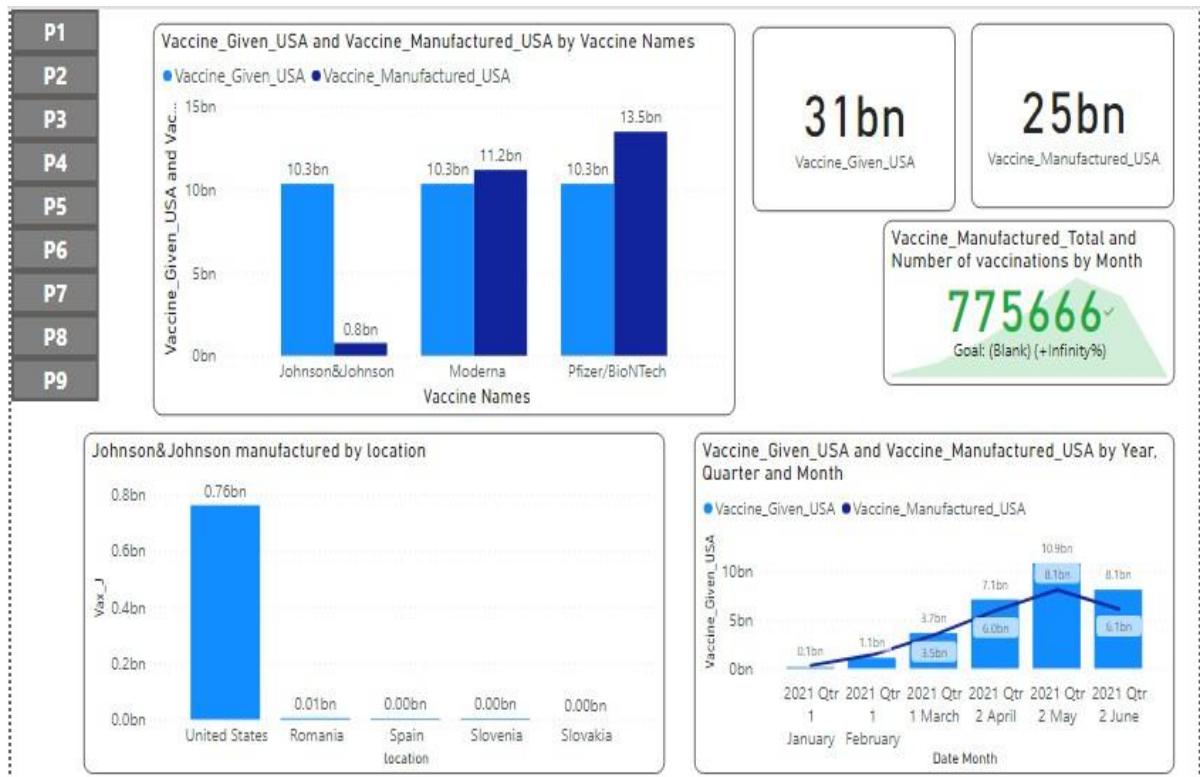
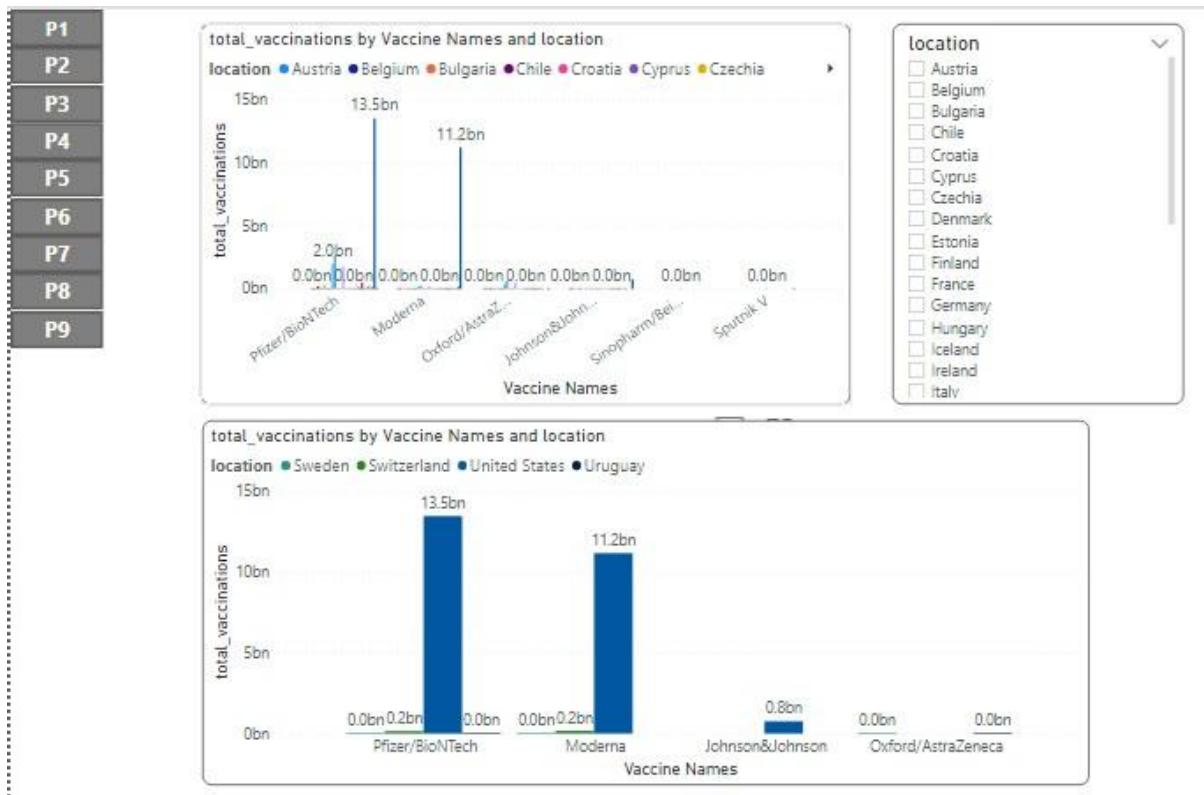


11) Using RANKX measure

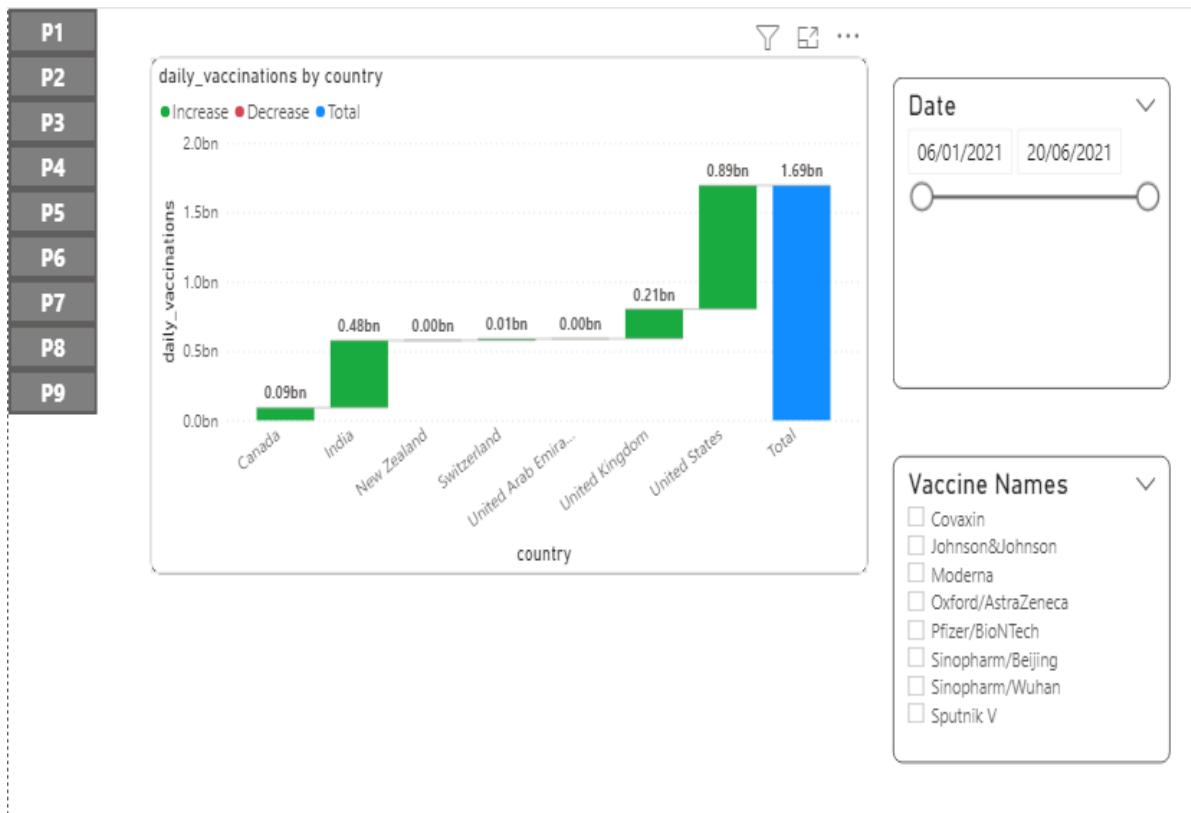
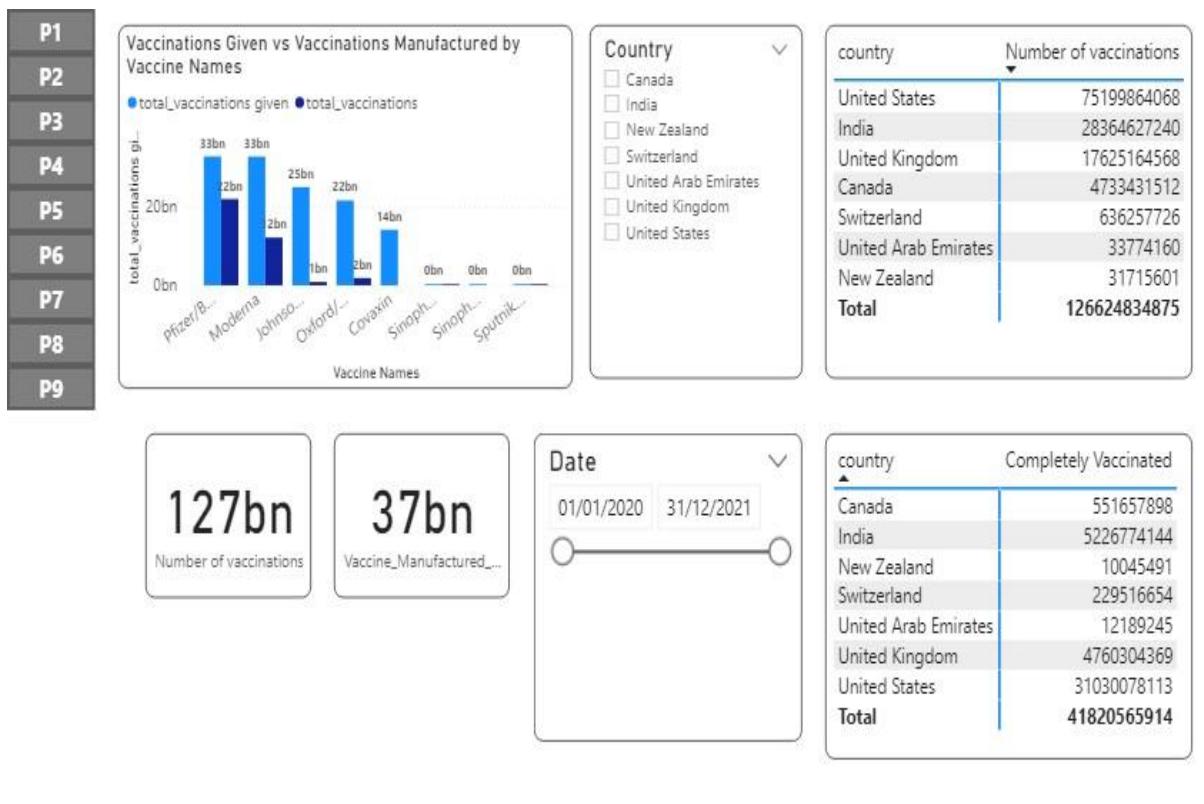


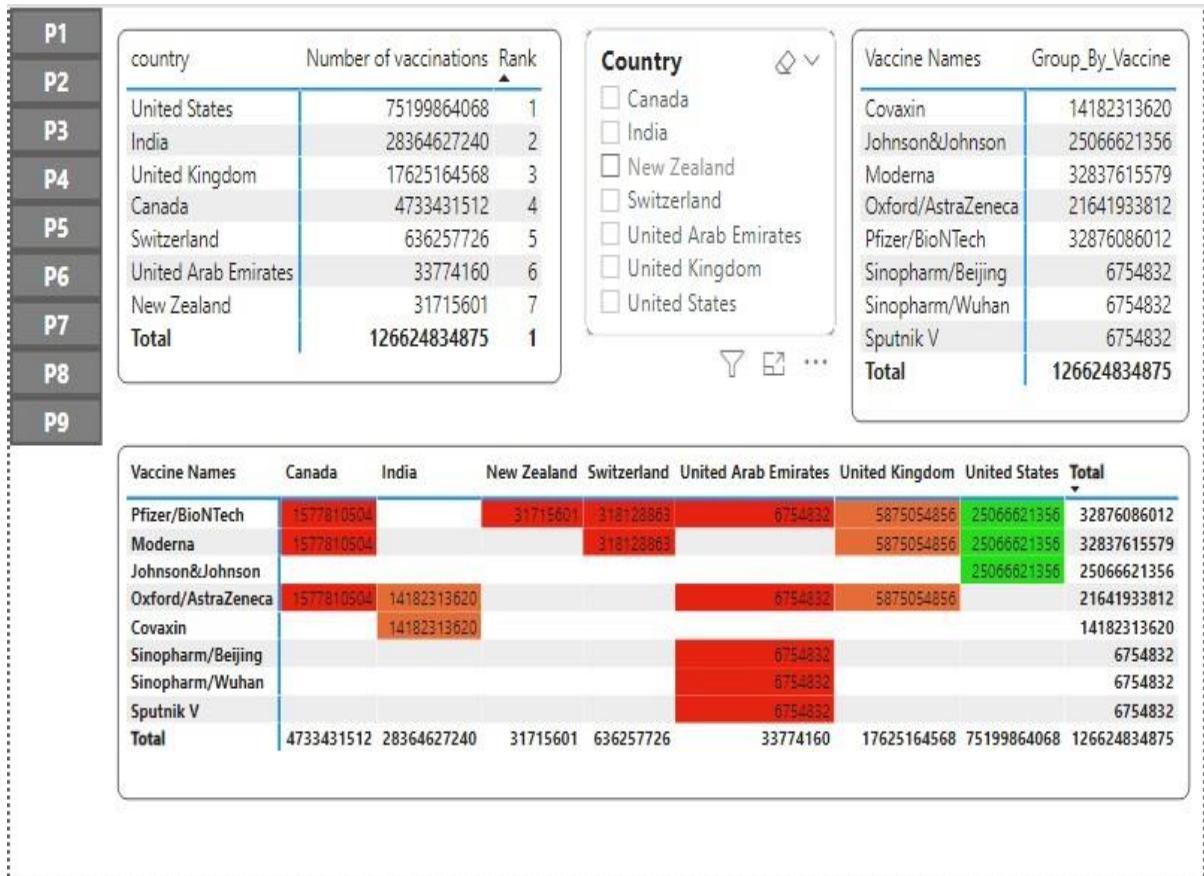
```
Ranking by Vaccine :=  
IF (  
    ISINSCOPE ( 'vaccines'[Vaccine Names] ),  
    VAR VaccinesToRank = [TopN_Vax Value]  
    VAR TotalVaccination = [Number of vaccinations]  
    RETURN  
        IF (  
            TotalVaccination > 0,  
            VAR VisibleVaccines =  
                CALCULATETABLE (  
                    VALUES ( 'country_vaccinations' ),  
                    ALLSELECTED ( 'vaccines'[Vaccine Names] )  
                )  
            VAR Ranking =  
                RANKX (  
                    VisibleVaccines,  
                    [Number of vaccinations],  
                    TotalVaccination  
                )  
            RETURN  
                IF (  
                    Ranking > 0 && Ranking <= VaccinesToRank,  
                    Ranking  
                )  
        )  
    )
```





Morgan Stanley





Background color - Number of vaccinations

Format by Rules Apply to Values only

Based on field Number of vaccinations

Rules Reverse color order + New rule

If value is greater than or equal to 0 Percent and is less than 15 Percent then █

If value is greater than or equal to 15 Percent and is less than 75 Percent then █

If value is greater than or equal to 75 Percent and is less than or equal to 100 Percent then █

P1
P2
P3
P4
P5
P6
P7
P8
P9

Top N Vaccines

| country | Number of vaccinations | Ranking by Vaccine |
|----------------------|------------------------|--------------------|
| Canada | 4733431512 | |
| India | 28364627240 | |
| New Zealand | 31715601 | |
| Switzerland | 636257726 | |
| United Arab Emirates | 33774160 | |
| United Kingdom | 17625164568 | |
| United States | 75199864068 | |
| Total | 126624834875 | |

Date

total_vaccinations given and people_fully_vaccinated by country

Bing

Country

- Canada
- India
- New Zealand
- Switzerland
- United Arab Emirates
- United Kingdom
- United States

P1
P2
P3
P4
P5
P6
P7
P8
P9

| country | Number of vaccinations | Partially Vaccinated | % Partially Vaccinated | Measure_Final | % Completely Vaccinated |
|----------------------|------------------------|----------------------|------------------------|---------------|-------------------------|
| United States | 75199864068 | 45723851439 | 51.29% | 31030078113 | 34.81% |
| United Kingdom | 17625164568 | 12864860199 | 60.22% | 4760304369 | 22.28% |
| United Arab Emirates | 33774160 | 21584915 | 36.24% | 12189245 | 20.46% |
| Switzerland | 636257726 | 406741072 | 35.91% | 229516654 | 20.27% |
| New Zealand | 31715601 | 21670110 | 26.06% | 10045491 | 12.08% |
| India | 28364627240 | 23137853096 | 47.83% | 5226774144 | 10.80% |
| Canada | 4733431512 | 4181773614 | 45.32% | 551657898 | 5.98% |
| Total | 126624834875 | 86358334445 | 50.98% | 41820565914 | 24.69% |

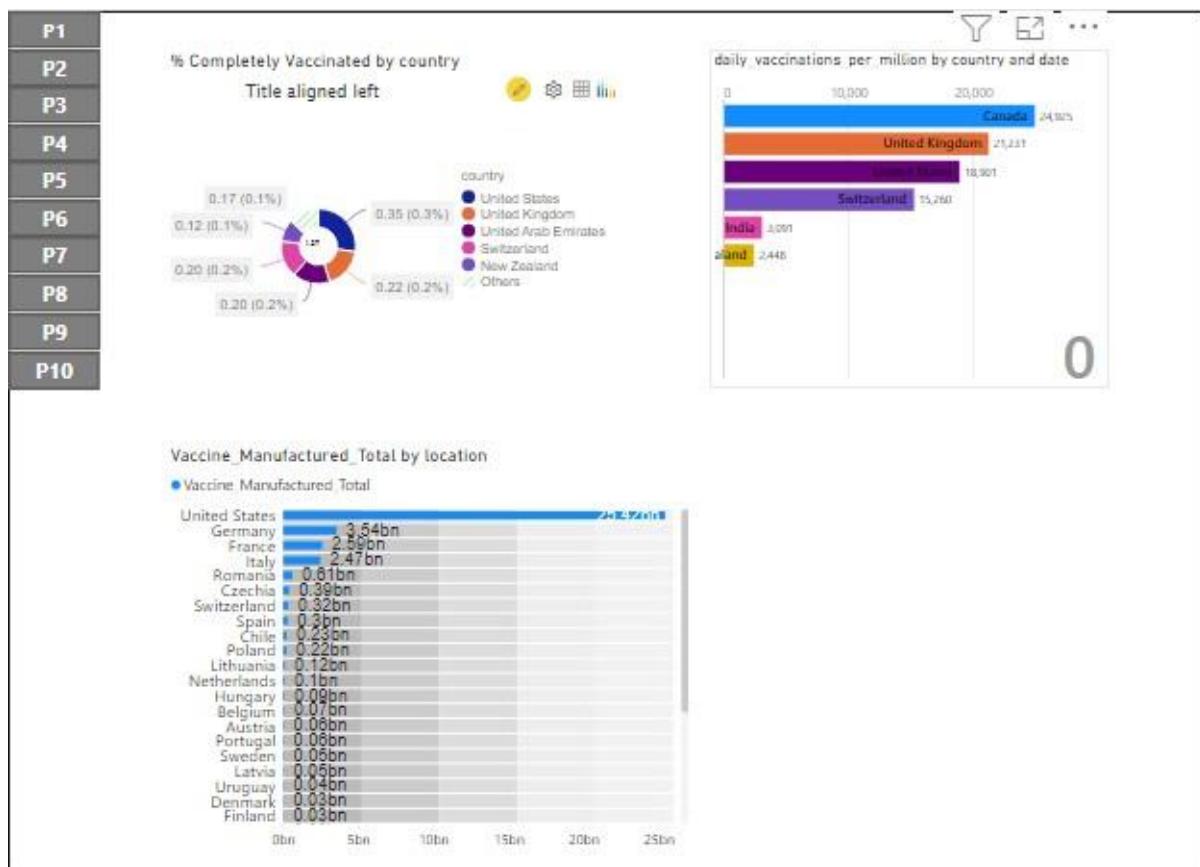
42bn
Completely Vaccinated

86bn
Partially Vaccinated

% of Completely Vaccinated by country

| Country | % Completely Vaccinated | (Measure_Final) |
|----------------------|-------------------------|-----------------|
| United States | 34.81% | (27.48%) |
| United Kingdom | 22.28% | (17.59%) |
| United Arab Emirates | 20.46% | (16.15%) |
| Switzerland | 20.27% | (16%) |
| New Zealand | 12.08% | (9.53%) |
| India | 10.80% | (8.53%) |
| Canada | 5.98% | (4.72%) |

Date



Custom Visuals in Power BI

1) Bullet Chart

Bullet Chart

Microsoft Corporation

Power BI visuals

PBI Certified

3.6 (36 AppSource ratings)

Pricing Free Get it now

A bar chart with extra visual elements to provide additional context. Useful for tracking goals. Bullet chart serves as a replacement for dashboard gauges and meters. Bullet charts were developed to overcome the fundamental issues of gauges and meters. The Bullet chart consists of 5 primary components:

- Text label: Your chart caption which defines what your chart is about and the unit of measurement.
- Quantitative Scale: Measures the value of your metric on a linear axis.

- The Featured Measure: The bar that displays the primary performance measure (eg: Revenue YTD).
- Comparative Measure: The measure against which you want to compare your featured measure (example: Target revenue).
- Qualitative Scale: The background fill that encodes qualitative ranges like bad, satisfactory, and good.

2) Animated bar chart race



Animated Bar Chart Race

Wishyoulization

Power BI visuals

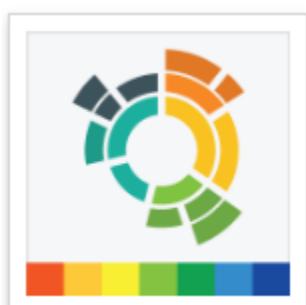
4.2 (42 AppSource ratings)

Pricing Free

Get it now

The animated bar chart race helps you visualize the change in trends over time, these types of charts are very popular on social media as they provide a holistic data story/insight in a concise and easy to understand chart. This visual has two modes, a standalone mode with auto play for animation or use as a ranked bar chart visual that can be integrated on reports with other filters to loop through the selections. The chart also supports configurations for colour, showing and hiding different graphical markings and labels, and animation related controls. For examples and tips on usage please check the sample report.

3) Advanced Pie and Donut:



Advanced Pie and Donut - xViz

Visual BI Solutions

Power BI visuals

PBI Certified

4.8 (4 AppSource ratings)

The Advanced Pie and Donut for Power BI enables you to track the progress of a task and for displaying the KPIs. It provides different Chart Types such as Pie, Donut, Semi Donut, Variable Pie to represent the data in a single visual. The key functionalities

include Top N Support, Conditional Formatting, and many more. It also works on a single measure value.

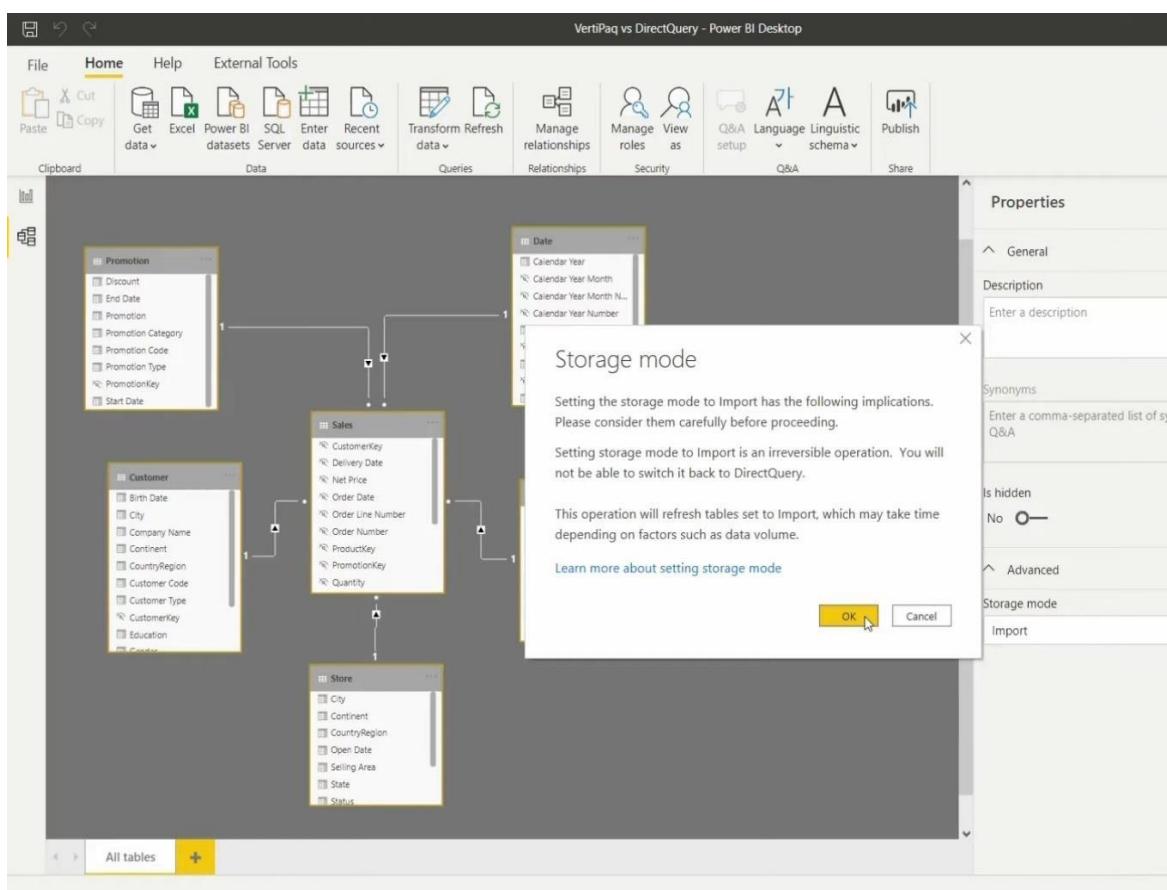
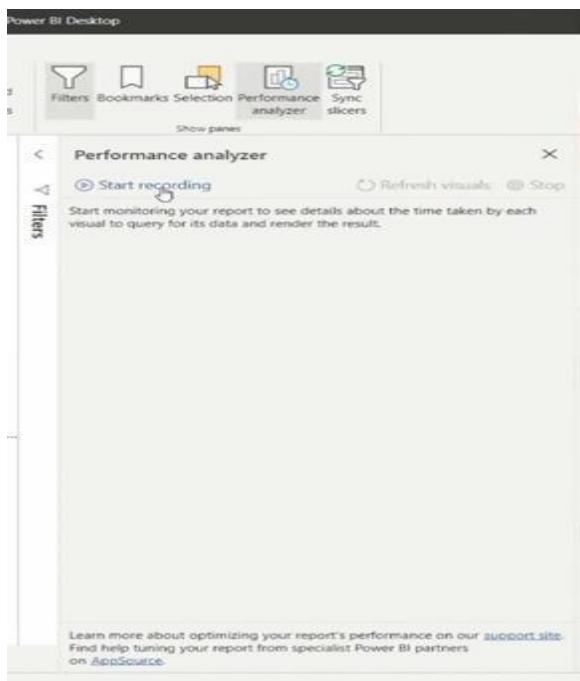
Power Query in Power BI

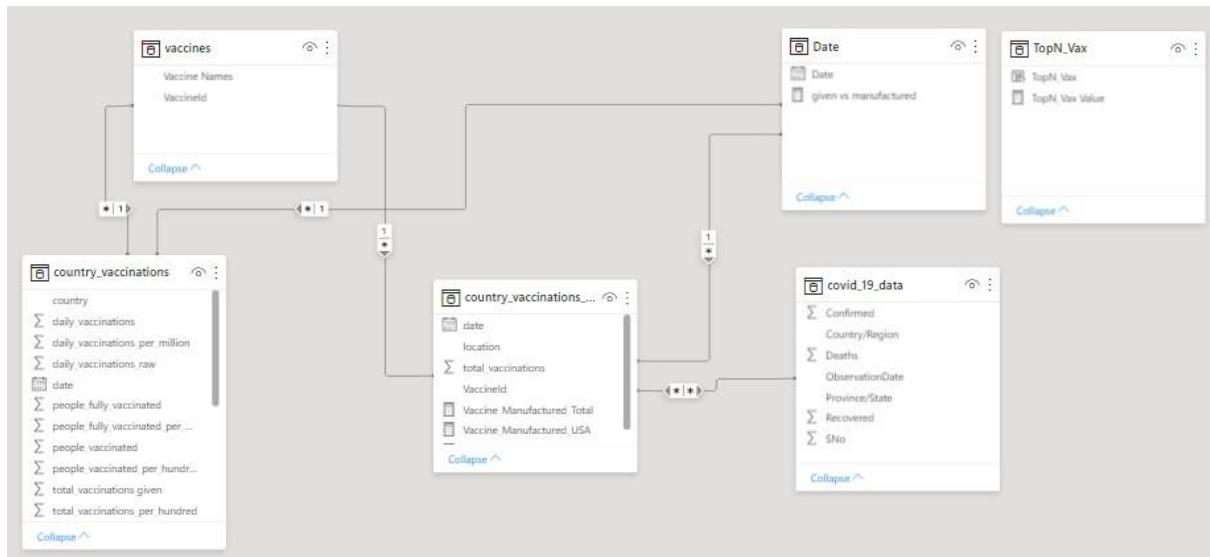
Power BI Desktop also comes with Power Query Editor. Use Power Query Editor to connect to one or many data sources, shape and transform the data to meet your needs, then load that model into Power BI Desktop.

- 1) Data Import: When one says to import data, also referred to as cached data, which is bringing our data from data source into Power BI. Working with import and cached data is the fastest method. All the data sources are available in an import perspective and thus no restrictions on the data source. Default option is import mode.
- 2) Without Power BI premium, your dataset size is limited to 1 GB. Without Power BI premium your dataset size can be up to 10 GB or more.
- 3) In import mode one copies the data into Power BI, but it's different with direct query. In direct query we connect to the data source and we grab the schema of the data, so the table structure (column names) is in the data model. Data stays itself in the data source. Direct query is essential and the ideal method for working with massive amounts of data. Not all data sources support direct query. When one does one faces performance issues. With direct query one faces performance issues. It is recommended to avoid using the Time Intelligence function in combination with Direct query.
- 4) Live connection connects to analysis services or power bi datasets. Composite models help one mix and match all connection types. For live connection direct query mode is required.



Morgan Stanley

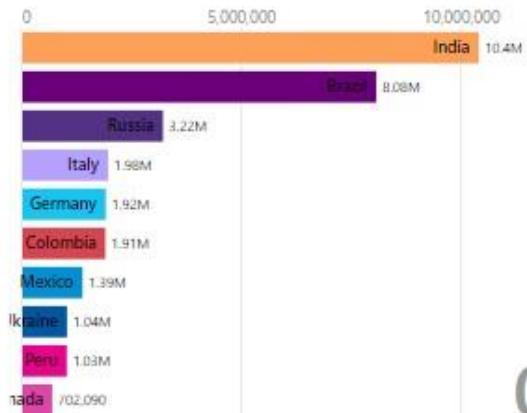




| location | total_vaccinations | Confirmed | Deaths |
|---------------|--------------------|--------------------|------------------|
| Germany | 3544264916 | 517436073 | 13150115 |
| Hungary | 93350725 | | |
| Iceland | 16799840 | | |
| Ireland | 29697749 | | |
| Italy | 2471052705 | 627754296 | 24852224 |
| Latvia | 49100322 | | |
| Liechtenstein | 182424 | | |
| Lithuania | 119441614 | | |
| Luxembourg | 3662096 | | |
| Malta | 4893176 | | |
| Netherlands | 102515669 | 230400838 | 3616057 |
| Poland | 210467258 | | |
| Total | 36934721321 | 18314899029 | 447832514 |



Recovered by Country/Region and ObservationDate



0

Add a Smart Narrative

One can also add a smart narrative to a report. The smart narrative visualization uses artificial intelligence to provide a text summary of your visuals and reports. It provides relevant insights that you can customize.

Options

GLOBAL

- Data Load
- Power Query Editor
- DirectQuery
- R scripting
- Python scripting
- Security
- Privacy
- Regional Settings
- Updates
- Usage Data
- Diagnostics
- Preview features**
- Auto recovery
- Report settings

CURRENT FILE

- Data Load
- Regional Settings
- Privacy
- Auto recovery

Preview features

The following features are available for you to try in this release. Preview features might change or be removed in future releases.

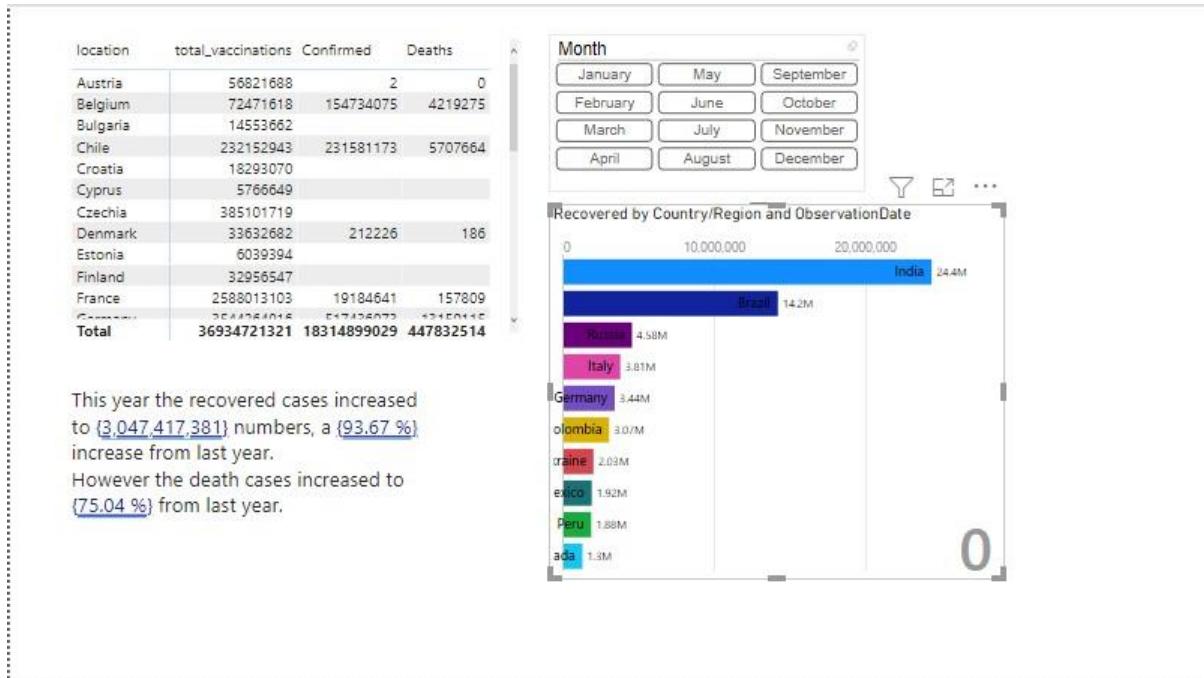
- Shape map visual [Learn more](#)
- Spanish language support for Q&A [Learn more](#)
- Json Table Inference [Learn more](#)
- New web table inference [Learn more](#)
- Import text using examples [Learn more](#)
- Excel table inference [Learn more](#)
- Q&A for live connected Analysis Services databases [Learn more](#)
- Information protection [Learn more](#)
- Azure map visual [Learn more](#)
- Data point rectangle select [Learn more](#)
- DirectQuery for Power BI datasets and Analysis Services [Learn more](#) | [Share feedback](#)
- Smart narrative visual [Learn more](#)
- Dynamic M Query Parameters [Learn more](#)
- Anomaly detection [Learn more](#)
- New field list [Learn more](#)
- Small multiples [Learn more](#) | [Share feedback](#)

OK **Cancel**

```

Sum_Confirmed_Cases_2020 = CALCULATE(SUM(covid_19_data[Confirmed]), FILTER(covid_19_data, CONTAINSSTRING(covid_19_data[ObservationDate], "2020")))
Sum_Confirmed_Cases_2021 = CALCULATE(SUM(covid_19_data[Confirmed]), FILTER(covid_19_data, CONTAINSSTRING(covid_19_data[ObservationDate], "2021")))
Sum_Recovered_Cases_2020 = CALCULATE(SUM(covid_19_data[Recovered]), FILTER(covid_19_data, CONTAINSSTRING(covid_19_data[ObservationDate], "2020")))
Sum_Recovered_Cases_2021 = CALCULATE(SUM(covid_19_data[Recovered]), FILTER(covid_19_data, CONTAINSSTRING(covid_19_data[ObservationDate], "2021")))
Sum_Death_Cases_2020 = CALCULATE(SUM(covid_19_data[Deaths]), FILTER(covid_19_data, CONTAINSSTRING(covid_19_data[ObservationDate], "2020")))
Sum_Death_Cases_2021 = CALCULATE(SUM(covid_19_data[Deaths]), FILTER(covid_19_data, CONTAINSSTRING(covid_19_data[ObservationDate], "2021")))

```



DAX uses functions to work on data that is stored in tables. Power Query (M) is used to query data sources, clean, and load data. Then you use DAX to analyse the data in Power Pivot.