IBM Journey to Cloud and AI

Analytics Modernization Workshop

Featuring: Cloud Pak for Data (CPD)

Lab Workbook

Lab workbook & CPD environment by: **Burt Vialpando**, Executive Analytics Architect

Adapted from original work by: **Vikram Khatri**, Executive Architect, Cloud Pak for Data (CPD)

September 18, 2019

Acknowledgements

- **Duane Almeter** and **Eric Watson** for their leadership and vision on this project
- **Daniel Kikuchi** for the design of the Collect labs, the CPD install and other technical support
- **Ed Duhe, Rich Russo** and **Joshua Laing** for providing the underlying development platforms in our Dallas and Raleigh TECs
- **Anjali Shah** for her help on updating the notebook in the Analyze Part 2 lab
- **Jeff Tuck** for his help in updating his original work on the predictive modeling microservice that enriches the Stock Trader "After" application
- **Frank Ketelaars** and **Adrian Houselander** for providing input on the workshop flow and overall design
- **Ashwin Dev**, **Prashant Patel, Kanda Zhang, Luiz Erico Machado de Almeida, Nicholous Markey, Louis Mau** and **Sahil Shah** for CPD development team support
- **John Lucas** for final product testing and workbook publishing

For work on the original workshop, additional thanks goes to:

- **John Van Buren** for the Organize lab design
- **Kent Rubin** Analyze Part 1 lab design
- **Anjali Shah**, **Rui Fan** and **Ben Chard** Analyze Part 2 lab design
- **Sriram Srinivasan**, Chief Architect of the Cloud Pak for Data (CPD)and his help
- **Craig Maddux, Dean Compher**, **Dominic Farrar**, **Gary Brunell**, **Karen Groski**, **Kyle Talish**, **Neal Finkelstein**, **Patrick Pitre**, **Paul Betts**, **Ryan Kather**, **Sang Suh**, **Dale Mumper** and **David Solomon** for marketplace leadership and active participation in testing the original workshop, as well as proctoring and contributions in delivering the workshops

Special thanks to:

Beth Friday, VP, North America Technical sales - our technical executive sponsor

Hemanth Manda, Director of Platform Offerings, and **Sampada Basarkar**, Director of Development, for providing help from their teams

--

Final thanks goes to my wife Sheniqua and my son Malcolm for putting up with my taking up many evenings and more than a few weekend time-outs to complete this project.

Contents

[This page left intentionally blank]

Lab 01 Introduction and Setup

1.1 IBM Journey to Cloud and AI: Analytics Modernization Workshop

This workshop provides hands-on experience with Cloud Pak for Data (CPD)that will show you how to modernize your microservices applications by enriching them with Machine Learning (ML) and Artificial Intelligence (AI).

The Journey to AI requires a strong information architecture that supports self-service capabilities and balances the needs of both the agility required by lines of business as well as the "Enterprise class" delivery required by IT. This journey can move significantly faster and with more efficiency when you use a single integrated platform like Cloud Pak for Data (CPD). It is the world's leading platform that allows you to **Collect**, **Organize** and **Analyze** data, and then **Deploy** the results into your applications to **Infuse** them with AI.

1.2 IBM Cloud Private (ICP)

IBM Cloud Private (ICP) is a private cloud computing platform run solely for one organization. With it you can develop and manage containerized applications using the container orchestrator Kubernetes, a private image registry, a management console, and a monitoring framework.

ICP can be managed internally or by a third party, and can be hosted on premises behind your company's firewall, or externally on a third-party provider's cloud. Private cloud offers the benefits of a public cloud including rapid deployment and scalability, ease of use, and elasticity. Additionally, a private cloud provides greater control, increased performance, predictable costs, tighter security, and flexible management options.

IBM Cloud Private can be customized to meet the unique needs and security requirements of your organization.

1.3 Cloud Pak for Data (CPD)

Cloud Pak for Data (recently renamed from IBM Cloud Private for Data) is an integrated end-to-end analytics platform designed to help make data more accessible and trusted, as well as providing access to many analytical tools to help your organization gain insights from your data.

Cloud Pak for Data (CPD) provides the data platform that accelerates the journey up the "AI Ladder." With it, you can quickly build, train, and deploy machine learning (ML) to create applications with Artificial intelligence (AI). CPD provides inventory and cataloging of your data sources, self-service shopping for data, and data integration and refinement capabilities. Thus, high quality, trusted data can be more easily prepared, assembled and used with this powerful integrated platform.

In this lab workshop image, CPD is installed on the foundation of ICP. However, CPD can be installed on other private cloud platforms such as OpenShift.

1.4 Audience for this IBM workshop

This IBM workshop is aimed at the line-of-business professionals who are tasked to gain new insights from all available data – regardless of its type and origin. The following personas who will be represented in the various labs will greatly benefit from this workshop:

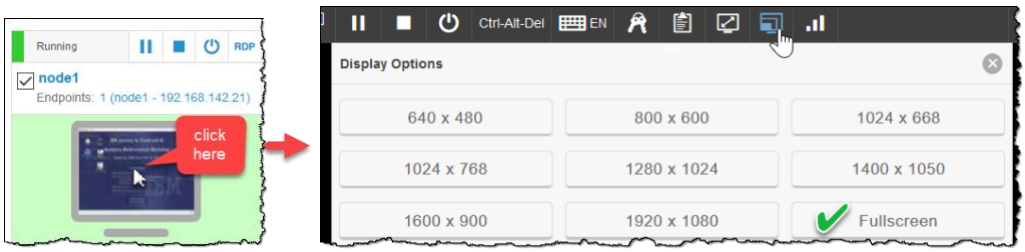
Persona (Role)	Capabilities
 Administrator	Administrators set up and maintain the CPD environment itself. `The exercises in this first lab represent some typical CPD Administrator activities.`
 Data Engineer	Data Engineers build and optimize the systems to allow data scientists and business analysts to perform their work. The Data Engineer ensures that any data is properly received, transformed, stored, and made accessible to other users.
 Data Steward	Data Stewards brings integration and governance to the data.
 Business Analyst	Business Analysts deliver value by taking data, using it to answer questions, and communicating the results to help make better business decisions.
 Data Scientist	Data Scientists brings expertise in statistics and the process of building ML/AI models to make predictions and answer key business questions.
 Developer	Developers create and maintain the end-user applications that utilize the output from all the other personas on the CPD platform.

1.5 Lab workshop environment

We are using a small CPD cloud cluster for this workshop. This software environment was built with the following key software components:

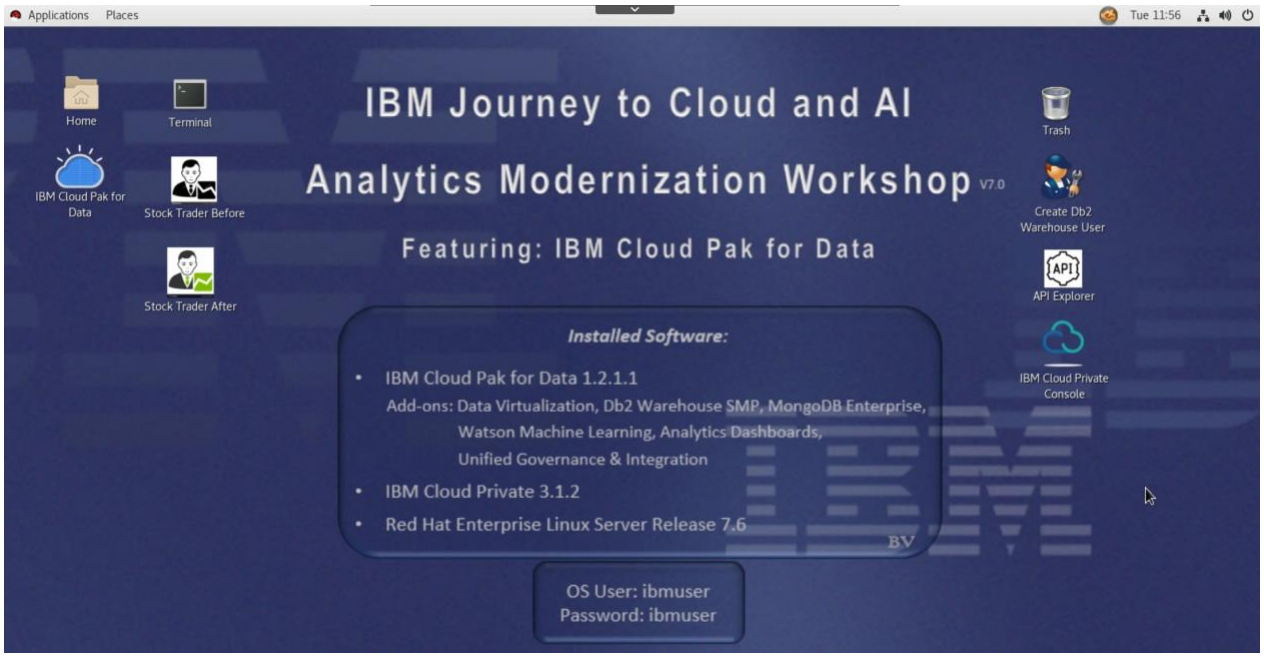
- `IBM Cloud Private 3.1.2` as the foundational cloud-native technology platform of Kubernetes, Docker and Helm, as well as other open-source tools.
- `Cloud Pak for Data 1.2.1.1` as the microservice-built, integrated data and analytics platform, installed on top of ICP, with various add-ons installed and enabled.

Note: Make sure to only click on the `node1` VM and then set to `Fullscreen`



1.6 Let's get started!

__1. On your Cloud VM, the lab desktop looks like this:

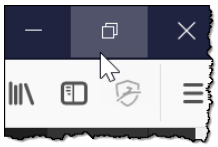


Admin	If a screensaver function locks the screen, hit `[Enter]` to get to the log in screen. Log back in with: User `ibmuser`, Password `ibmuser`

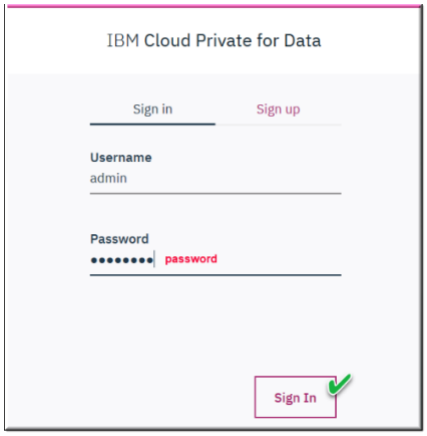
__2. Make sure to run in "Fullscreen" mode to make the most of your computer screen's real estate.

Double-click the icon `Cloud Pak for Data` on your desktop to open the browser that will start the CPD web console.

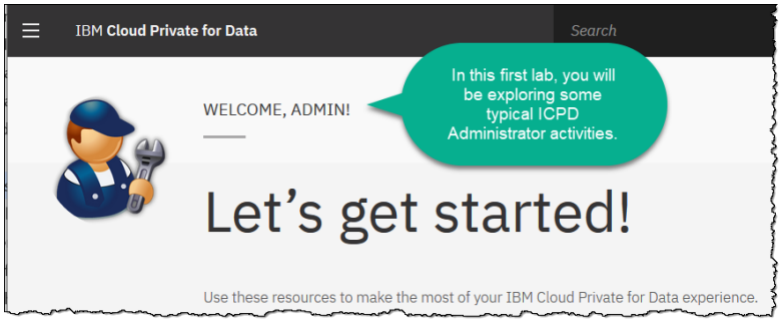
__3. After launching, maximize the browser window to make the most of your browser desktop real estate.



__4. The CPD web console GUI displays as shown. Use `admin` and `password` for the *Username* and *Password* and click `Sign In`.



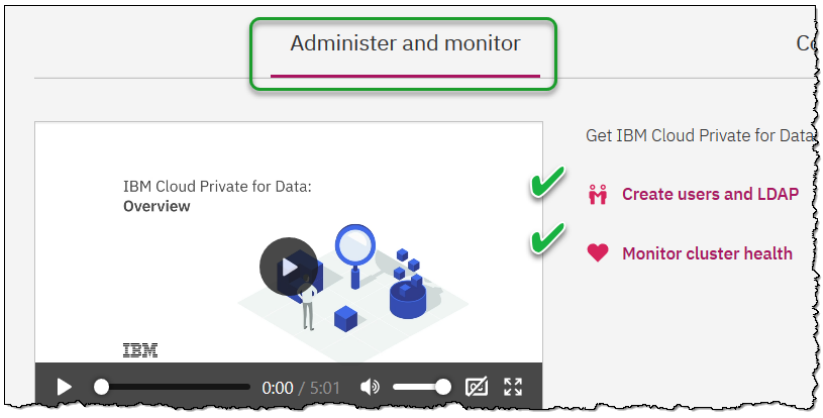
__5. You should now be at the page: `Let's get started!`



__6. Spend a few minutes reviewing some of the links in the page `Let's get started!` (**Note: you don't have to watch any of the videos.**)

a) The **Administrator and monitor** categories provide administration tools to organize teams, define roles, provide access, and perform monitoring of the CPD cluster health.

Review the content in the links checked below - don't change, edit, or create anything.

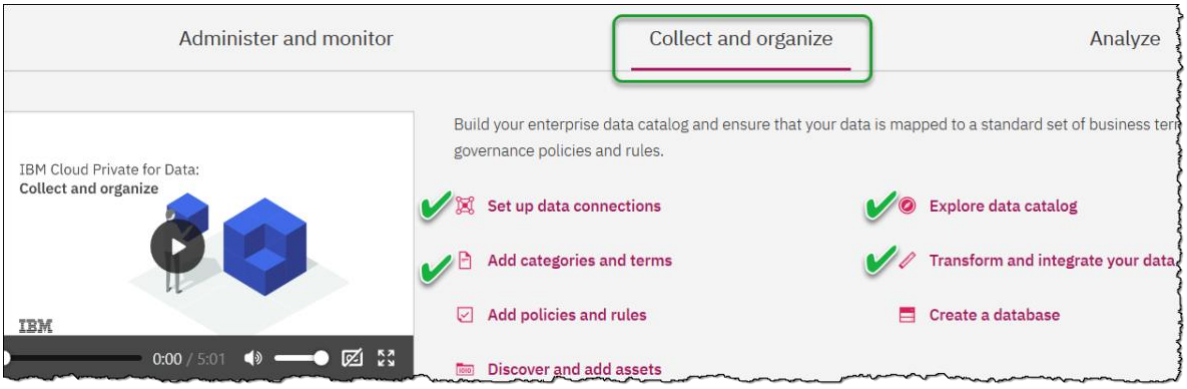


When you are done reviewing any CPD web console page in this exercise, use the back arrow key to return to the `Let's get started!` page.

b) The **Collect & organize** categories provide the ability to collect data from many data sources, no matter where they reside, e.g. IBM Db2®, Oracle, Teradata, Hadoop, flat files, etc. It does not matter if data is stored in a private or public cloud, or on-premises.

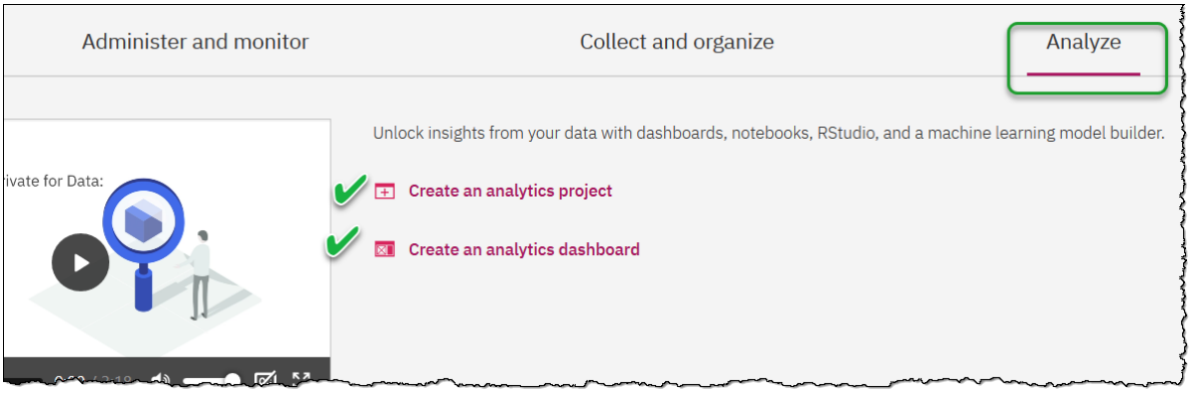
It also provides a way to organize that data using embedded machine learning to assign business terms to newly discovered data, and then enforce governance policies and create a catalog of the data to enable easily prepare that data for analysis.

Review the content in the links checked below - don't change, edit, or create anything.



c) The **Analyze** category allows you to build predictive and prescriptive models using open source programming languages (typically for the data scientist) or using graphical user interface tooling (typically for the business analyst). With the click of a button, deploy these models into production and publish an Open REST API that can be consumed by any application running on IBM Cloud Private or any other platform. Build dashboards and interact with your data to gain insight and business understanding.

Review the content in the links checked below - don't change, edit, or create anything.



Admin	In this workshop, we will demonstrate the `Collect, Organize` and `Analyze` capabilities to create a machine learning model that can you can `Deploy` for consumption by a microservice application.

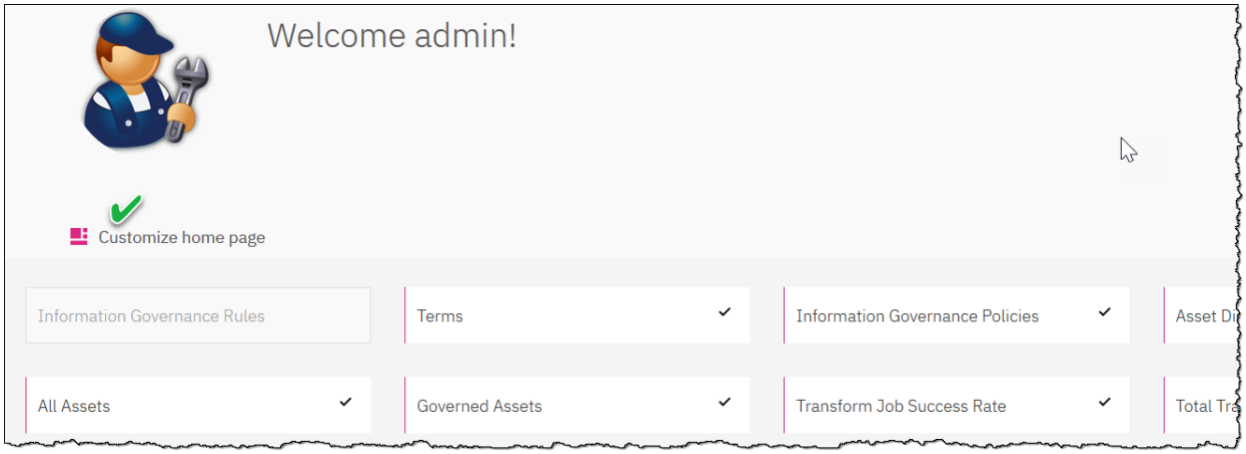
1.7 Explore the home page

__7. From the `Let's get started` page, click on `Go to your home page`.

__8. Click on `Customize home page`

Notice the home page shows the inventory of the integrated data platform for Information Governance Rules, Terms and Policies, Asset Distribution, All Assets, Governed Assets, Transform Job Success Rate, Total Transform Job Executions. The intent of this page is to provide individual customizations on daily tasks for each CPD user.

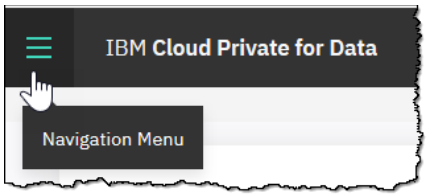
Scroll through the page to see inventory and asset types. Since we are starting fresh, there is not a lot to see now, but we can visit the home page later to review the list of assets after we have created new ones.



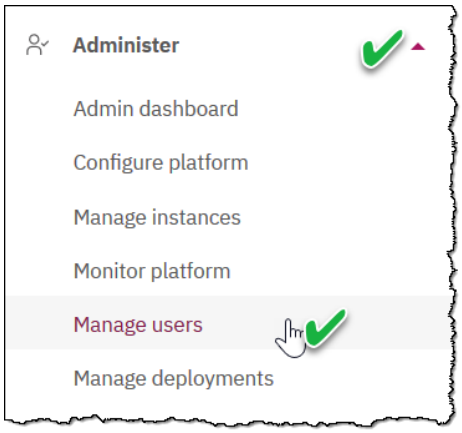
1.8 User management: Persona-based roles and teams

This section explores user authorizations for the various stages of the data analytics pipeline.

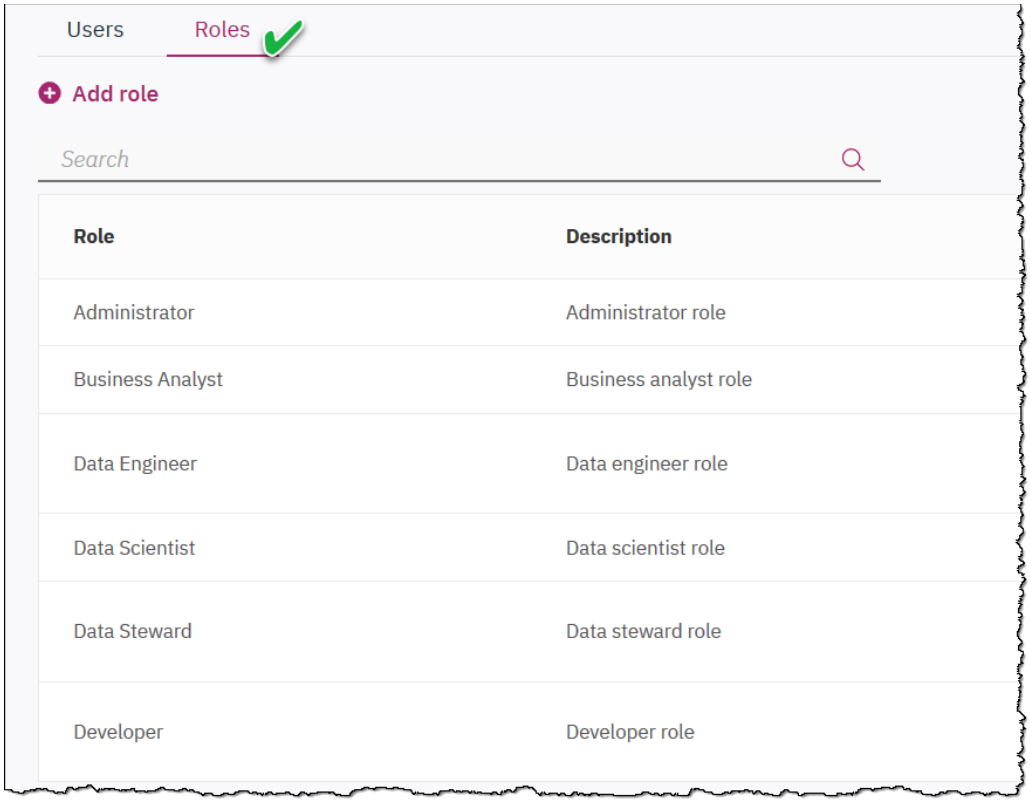
__9. Click on the hamburger icon on the top left part of the screen to access the `Navigation Menu`:



__10. Click on the `Administer` twistie to display the drop-down menu, and then click on `Manage users`

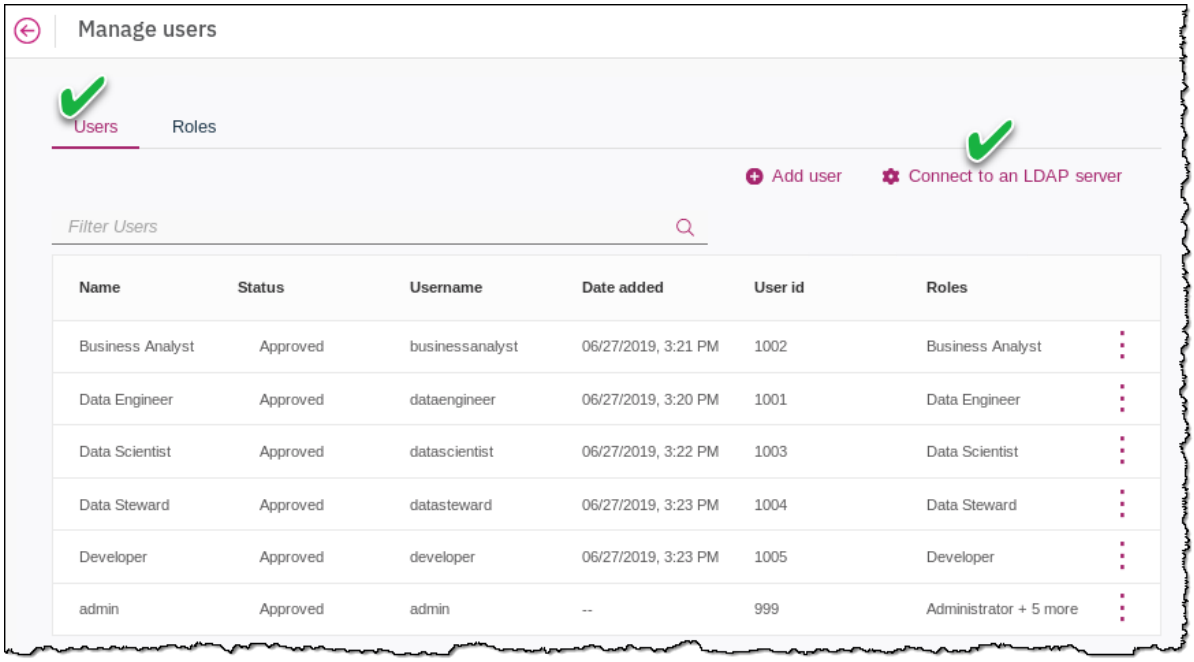


__11. Click on the `Roles` section to review the various personas that can be represented by any given user. A user can be granted more than one role if needed.

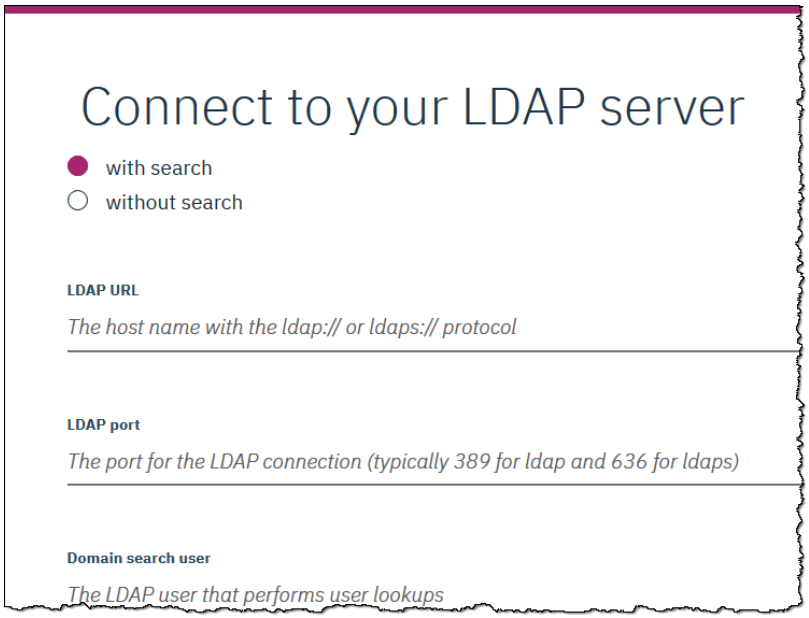


Admin	These existing roles can be customized, or new ones created, to suit your company's needs. CPD is very much "persona driven" in that each user can play their particular part in your organization's journey to AI. Each user (acting as one or more personas) can hand off and/or share their work with other users/personas, for a totally collaborative environment.

__12. Click back to the `Users` section and then click on `Connect to an LDAP Server`.



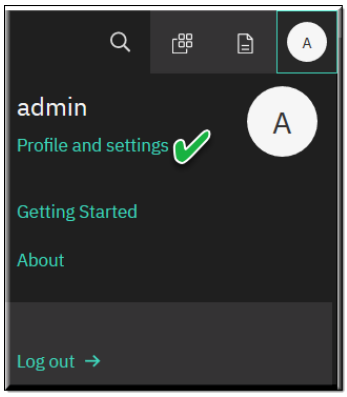
Notice that you can integrate with your LDAP server by creating a connection to it using “with search” or “without search” approaches.



1.9 Profile settings

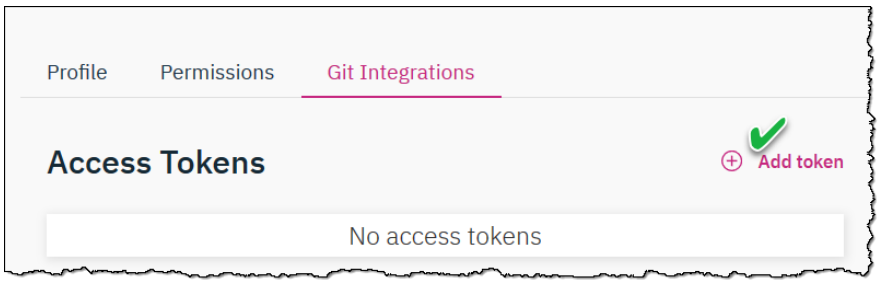
__13. Click the top right circle of your screen. Currently, it has the letter “A” on it, which is the first letter of “admin.” If you were to log in with a user name like “ibmuser” this circle would say “I”.

Choose `Profile and settings`

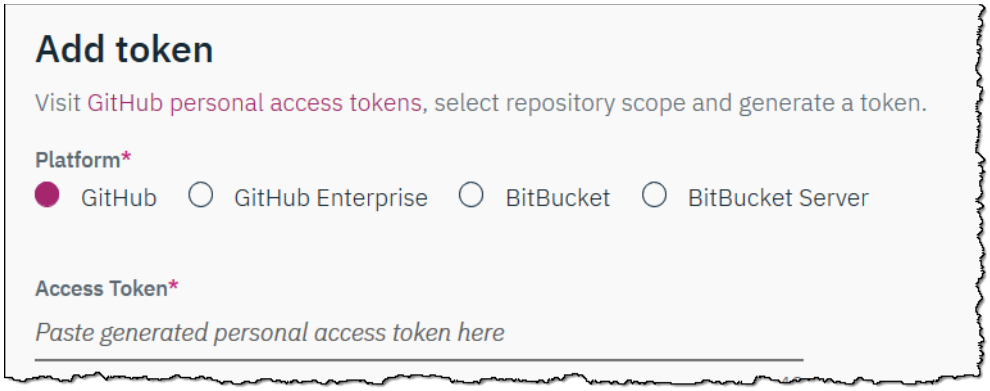


__14. Then review the things you can change in your `Profile`, then review `Permissions`, then go to `Git Integrations` (Note: in Permissions you have all permissions because you are the Admin user. A non admin user would NOT have all these privileges.)

__15. In `Git Integrations` ⇨ `Add token`.

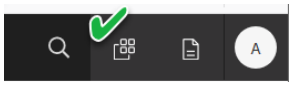


__16. Notice that you can manage your repository with `GitHub, GitHub Enterprise, BitBucket` and `BitBucket Server`.



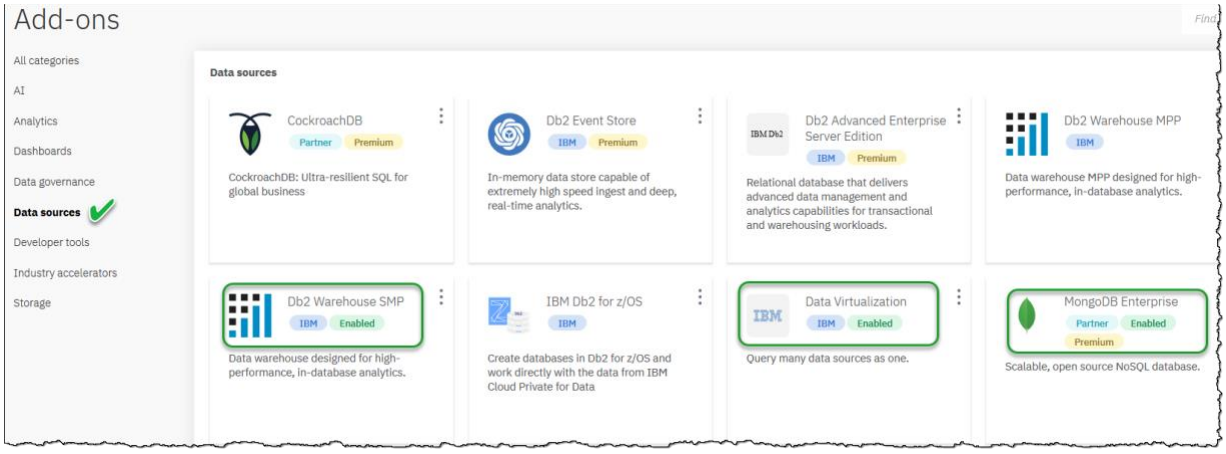
This capability allows you to integrate CPD projects with your current CICD (Continuous Integration and Continuous Delivery) pipeline to automate delivery of the artifacts you create in the CPD platform. You can use capabilities from the underlying IBM Cloud Private platform to build cloud native microservice applications which are tied to the ML / AI model development and delivery pipeline.

__17. Click the `Add-on` icon (four little squares over one bigger square) on the top right corner of your screen.



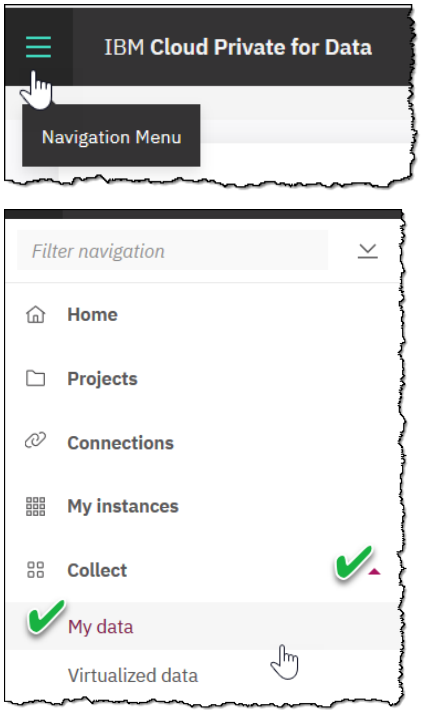
__18. This will bring up all available add-on products for CPD.

Choose `Data sources` from the menu (or scroll down to find this section) and notice that `Db2 Warehouse SMP, Data Virtualization` and `MongoDB Enterprise` have been "enabled," which means they will be available to be used in this workshop.



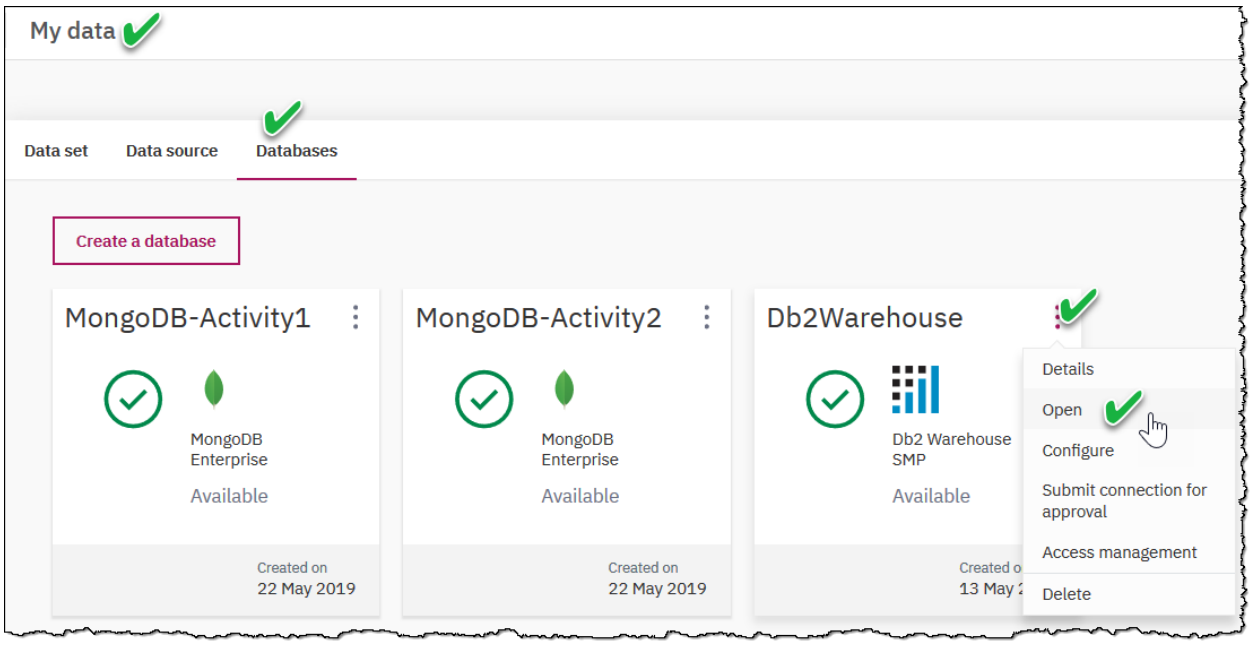
Admin	There are currently many other add-ons available for your organization's use at the click of a button, and IBM continues to add more with each release of Cloud Pak for Data.

__19. To review the Db2 Warehouse that has been installed, click on the `Navigation Menu` hamburger icon ⇨ `Collect` ⇨ `My data`

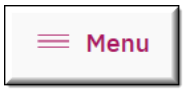


__20. Choose the `Databases` section to see `Db2WarehouseSMP`.

Click the ellipses (the three stacked dots) then `Open` to launch the *Db2 Data Server Manager* console.



__21. From here, many options are available to you for using this database through the hamburger menu icon. Choose it to do a quick review the capabilities of this Db2 Warehouse GUI.

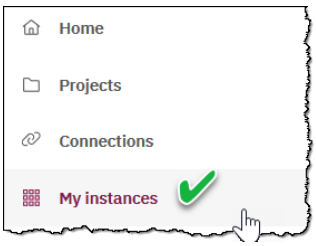


We will be using this database add-on in later exercises as a different persona, so there is no need to explore further now.

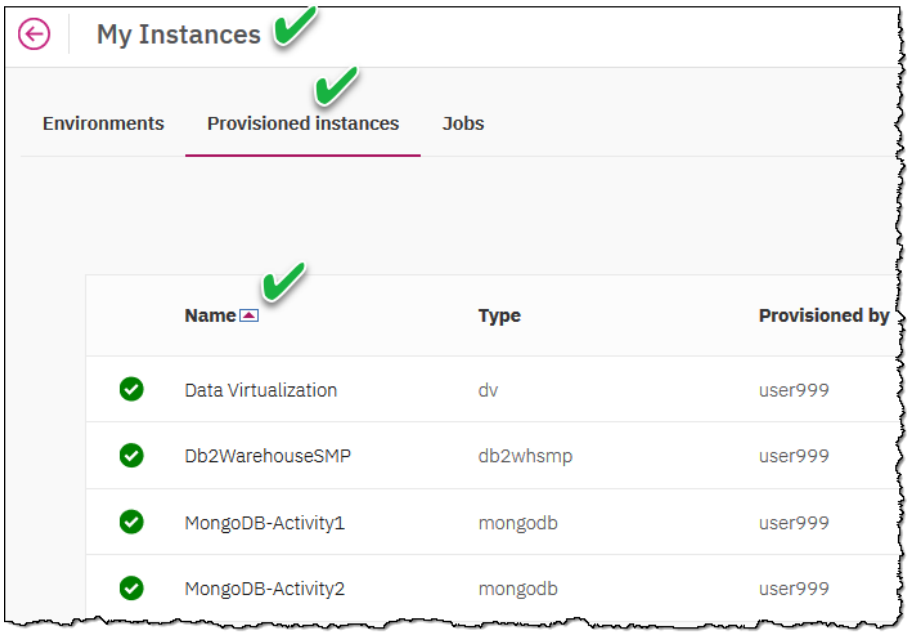
 Admin	Applications and databases available through add-ons are native to, and integrated on, the CPD platform. Besides using its own products, IBM partners with MongoDB, portworx, Datameer, Lightbend, Senzing Prolifics and others, to build a rich ecosystem around the CPD platform. Details can be found at https://www.ibm.com/products/cloud-private-for-data/partners

1.10 Instances

__22. Click on the hamburger icon to access the `Navigation Menu` ⇨ `My Instances`



__23. Click tab `Provisioned Instances` and then click the twistie to sort the instances of Data Virtualization, Db2Warhouse, and MongoDB databases that were provisioned for this workshop:



1.11 Lab conclusion

Cloud Pak for Data is most useful in the following business use-case scenarios:

1. **Manage All Your Data**: Use discovery to automate the process of assigning business terms to technical assets. Use enterprise catalog to secure, govern and control access to your data regardless where it resides. Use a combination of virtualization and transformation to prepare your data for analysis.
2. **Build Your Ladder to AI**: Build a strong information architecture to help you realize the value of leveraging machine learning and AI.
3. **Modernize Your Data & Analytics Workloads**: Modernize your data platform and provide data scientists and application developers the ability to quickly add AI to your applications. Since ICP is the foundation of CPD, you can use ICP to build cloud-native, microservice-based applications and use CPD to enrich them with machine learning and artificial intelligence.
4. **Compliance Readiness** – Our strong governance capabilities allow you to provide regulatory compliance.

Make Cloud Pak for Data your platform for data and analytics. Why? Because IBM understands data and provides an integrated, end-to-end data platform that enables enterprises to:

- Collect relevant data and make it simpler and more accessible
- Use federation, virtualization and/or transformation to combine and refine data sets
- Organize data so it can be trusted
- Analyze insights on demand
- Infuse machine learning in your applications

All of the above will be demonstrated in the following workshop labs.

** End of Lab 01 – Introduction and Setup

Lab 02 Executive Demo

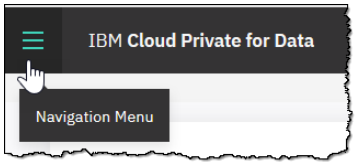
The key purpose of this lab is to see what an application called Stock Trader could look like before and after it has been infused with a machine learning (ML) model created from the CPD platform.

Before you begin to use CPD to create something, this lab shows you what can be accomplished with it.

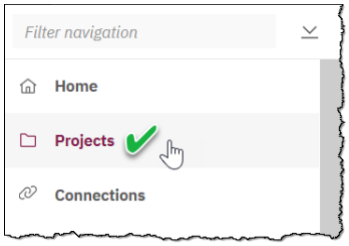
2.1 Get started by importing a project

To get started, we will work with a "project" that has been created for this workshop which was exported to a zip file. As you will see in more detail later, a project is a useful collaborative CPD construct that allows a team to work together on many assets with appropriate access controls.

__1. In the CPD console, click on the hamburger icon to access the `Navigation Menu`:



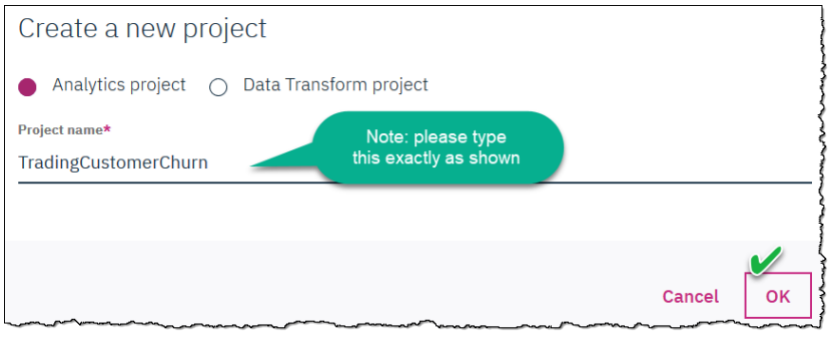
Then choose `Projects`



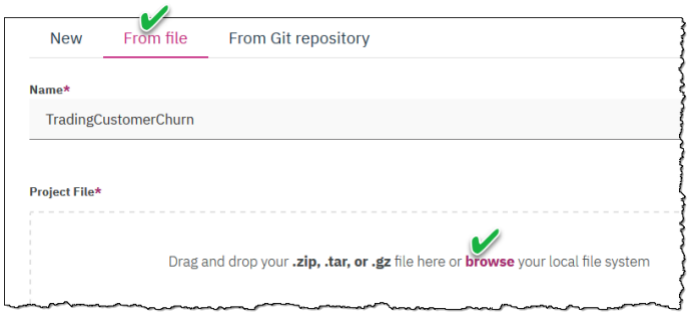
__2. Click `New project`.

__3. Give the project the name `TradingCustomerChurn`.

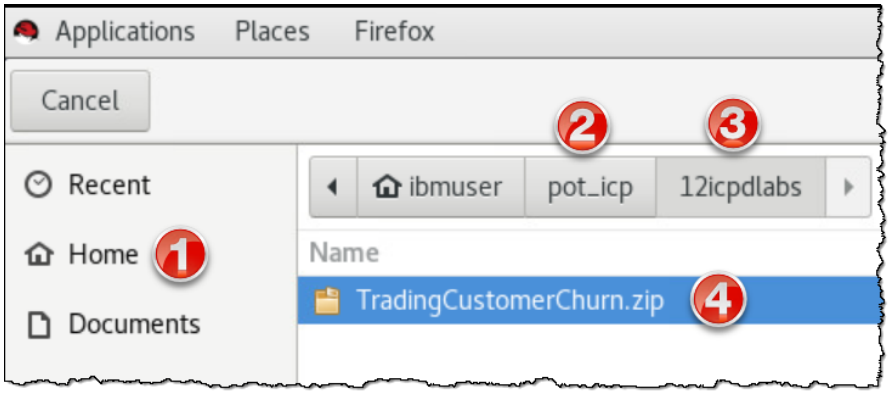
Note: case is important for this exercise, please enter the project name exactly as shown and then click `OK`.



__4. Click the middle tab `From file` and then click the link `browse`.



__5. On the left pane, double-click `Home` (which is the home for *ibmuser*) then choose directory `pot_icp` then directory `12icpdlabs`. Finally, double click file `TradingCustomerChurn.zip`

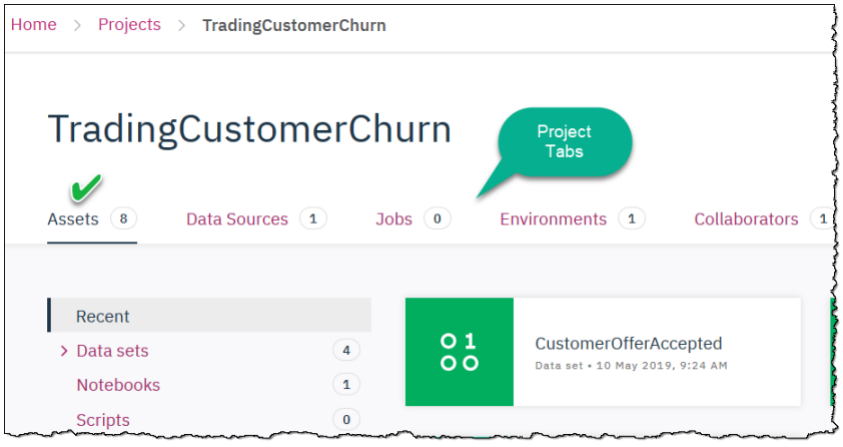


__6. Click `Create` (towards the bottom right) to create the new project by importing the zip file.

Note: If there is a problem with importing the zip file. Delete and retry the import.

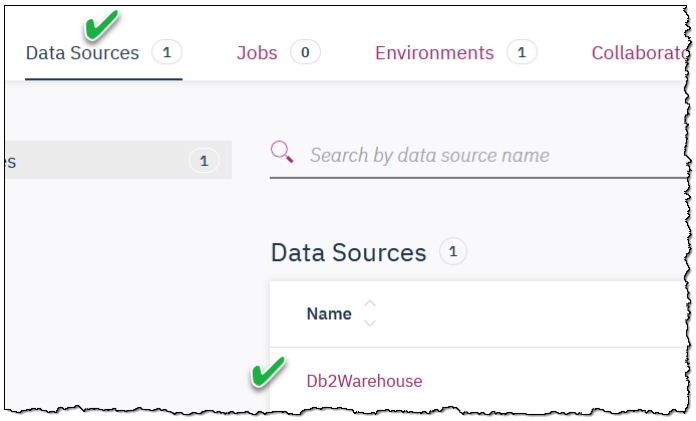
The project creation process launches multiple resources in the CPD cluster, which may take a minute or two – be patient while it completes.

__7. When it is finished, you will be presented with a project navigation screen that is at first positioned at the `Assets` project tab. Notice there are other tabs like `Data Sources`, `Jobs`, etc.



2.2 Connect to a Data Source

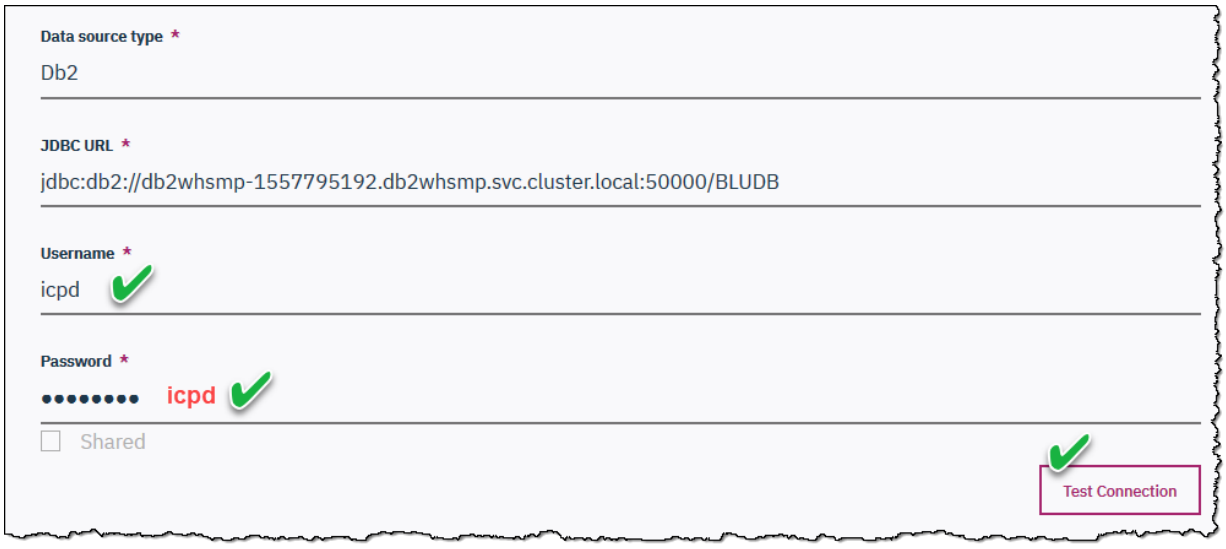
__8. Next, click the tab called `Data Sources` and then `Db2Warehouse`.



	You might recall that in this lab environment we installed Db2 Warehouse as an add-on feature, indicating that it has tight integration with CPD and is cloud-native. This is the case for all add-ons that you choose to deploy in CPD.

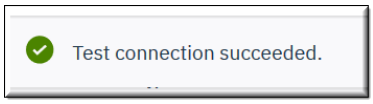
__9. Scroll down and specify *Username:* `icpd` and *Password*: `icpd`

Click `Test Connection` to make sure your credentials work.



__10. You should get `Test connection succeeded`.

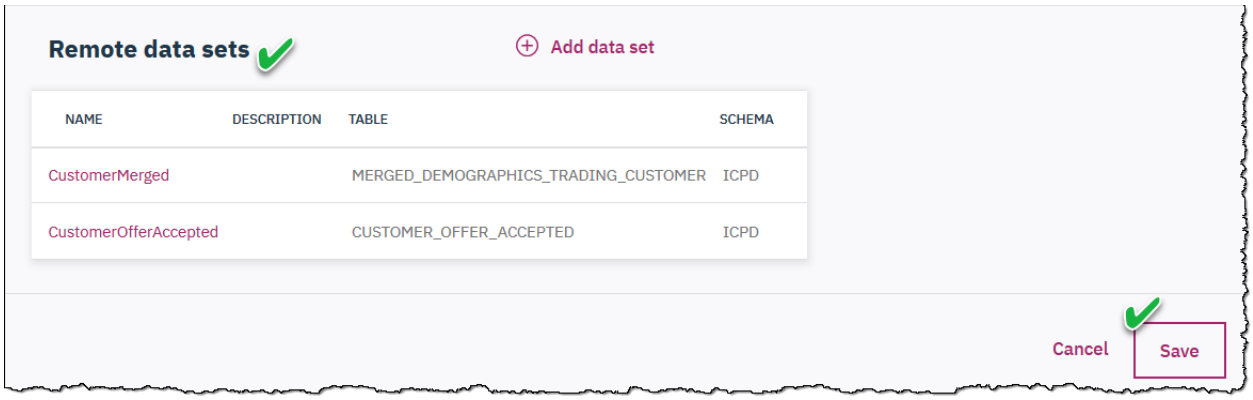
If not, retype the username/password again until you get this message… and make sure your case is correct!



	If the `Test connection` fails with a hostname issue, see Appendix B – Lab fixes – Lab 02 “Test connection fails”

__11. Scroll down to review the two `Remote data sets` already in the project. In this case, these particular “data sets” are actually tables in the database

Before you exit this screen, save the Username/Password changes you did earlier in this screen by clicking `Save`.



	If you have trouble rendering the dashboard later in this lab, that is because you did not save the credentials to this data source and did not test the credentials properly.

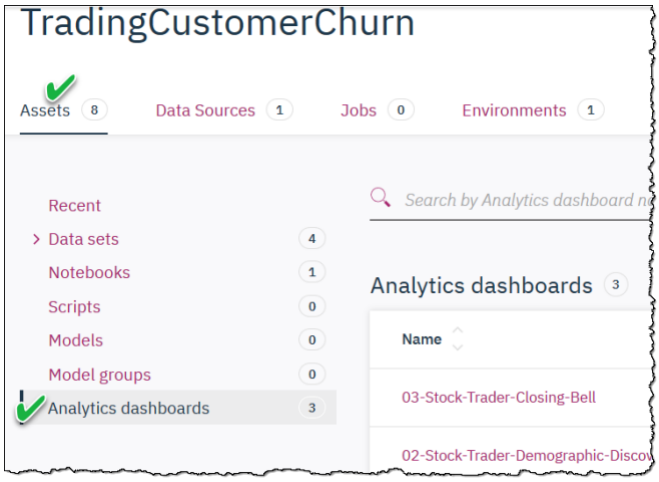
2.3 Stock Trade Opening Bell Analysis dashboard

This is the business use case scenario: Boatswain Trading is aware that revenues are declining. They have used CPD to create various assets to help them with this problem.

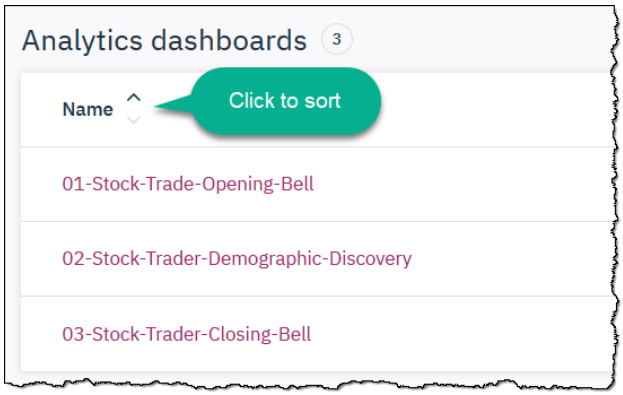
To begin, we will discover the trend of that revenue decline by reviewing some simple visualizations. In this scenario, the business analyst has requested aggregated stock transaction data. It contains data such as the historical total number of individual customer visits to their website as well as the number of trades per customer for the past year.

__12. Let's take a look at an example dashboard that the business analyst has already created. (Note: in a later lab, you will build a dashboard from scratch.)

In the project, click the `Assets` tab, then the `Analytics dashboards` option:



__13. Notice we have three pre-built dashboards. Click on the up/down arrows next to `Name` to sort them.

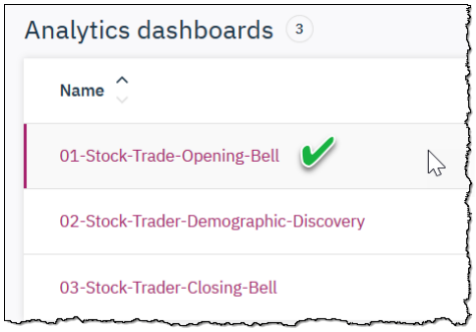


__14. The first dashboard (`01…`) shows a visualization of the 'Stock Trade Before' analysis.

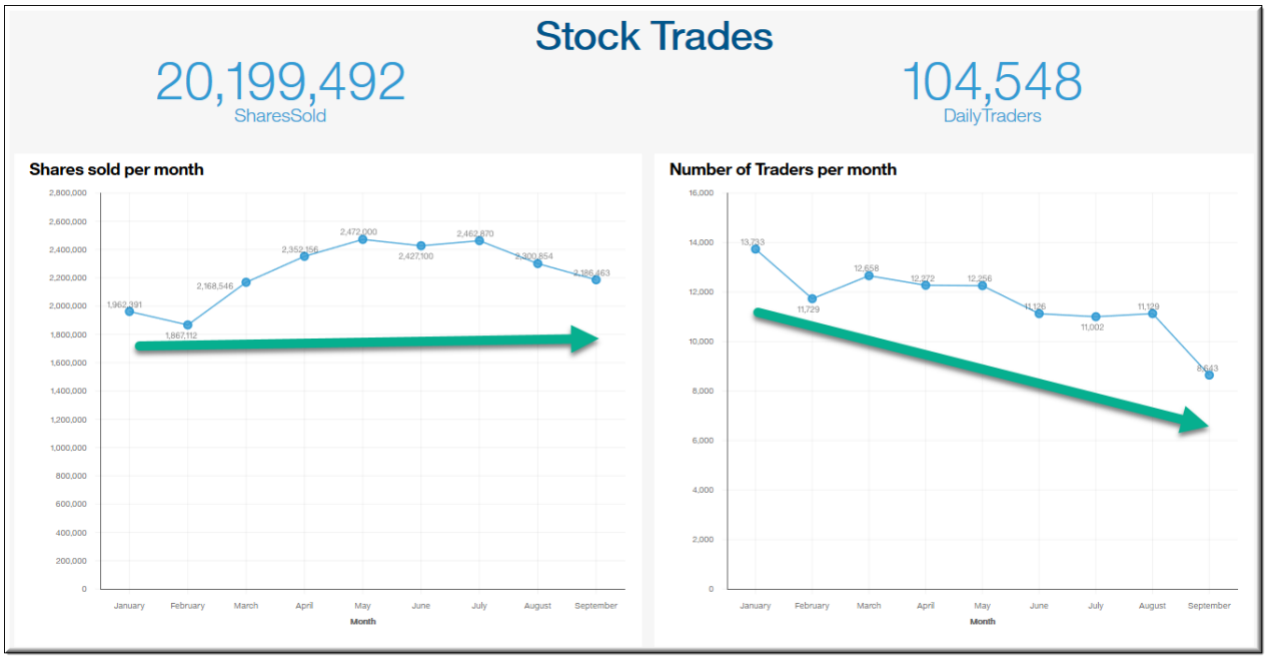
The second dashboard (`02…`) shows the results of the analysis of the customer demographics data that was collected from various sources (such as Db2 Warehouse and MongoDB) after curation of the data. This information is used to evaluate which measures (or factors, or variables) have the greatest impact on the customer churn.

The third dashboard (`03…`) shows the visualization of the Stock Trader application after implementation of the ML model and the impact that it had on the business.

__15. To view the first dashboard, click on it: `01-Stock-Trade-Opening-Bell`.



__16. This dashboard shows the consolidated report of the business in shares sold and traders. You can see that the number of shares sold per month is relatively flat, and worse, daily traders are declining.



Hint: to adjust the zoon on your browser to see more or less of any dashboard, try using [Ctrl][Mouse-scroll-wheel]

	In CPD, the data for this report is stored at a central place and is governed and managed with team collaboration between business analysts, data engineers, data scientists and data stewards.

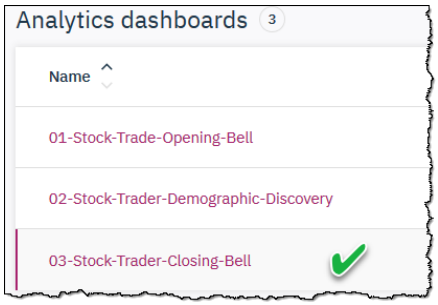
Note: See *Appendix A* for details on building this dashboard from scratch.

2.4 Stock Trader Analysis Closing Bell analysis

After completing the review of how the business was looking in the beginning, let's review the same information after Boatswain Trading infused a Machine Learning model into their Stock Trader application to improve its business.

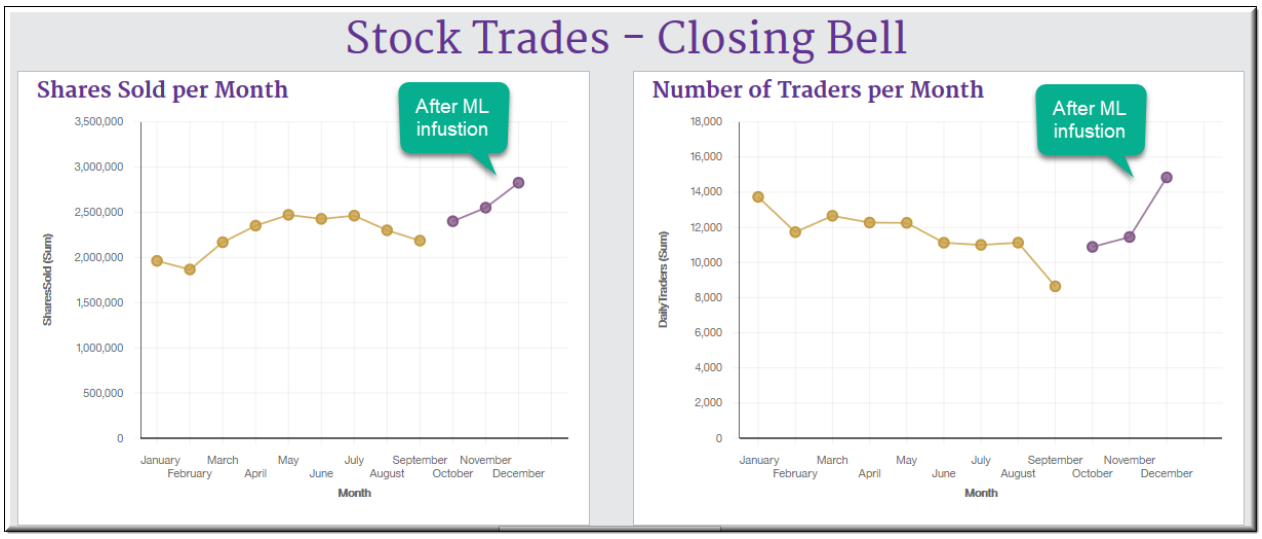
__17. Click on the browser back arrow key to return to your `Analytics dashboards` options.

__18. Now choose dashboard: `03-Stock-Trader-Closing-Bell`



	If the dashboard does not render, see Appendix B – Lab Fixes - Lab 02 – "Dashboard doesn't render."

__19. Notice the visualization trend of `Shares Sold per Month` and `Number of Traders per Month` match with the Opening Bell dashboard we just reviewed… at least, up until September.



The data from October on represents the results of Boatswain Trading's microservice application that was infused with a machine learning model. This model was built and scored in CPD using machine learning algorithms that allowed the application to offer personalized promotions based on their history and risk of churn (leaving Boatswain Trading for another company.)

The results show that the following months after implementation of this new model, they had a large increase in both the number of shares sold per month, as well as the number of stock traders per month.

Boatswain's Trading business is now increasing rather than decreasing – this means the initiative was a resounding success!

__20. To be fair, this success did come with a cost: the free trades and consultations.

Scroll down in the dashboard to see the lower set of visualizations that give us insight into what that cost turned out to be. Note the total number of retention offers accepted and the total cost of those offers, which are summarized and broken down by the offer accepted “type.”

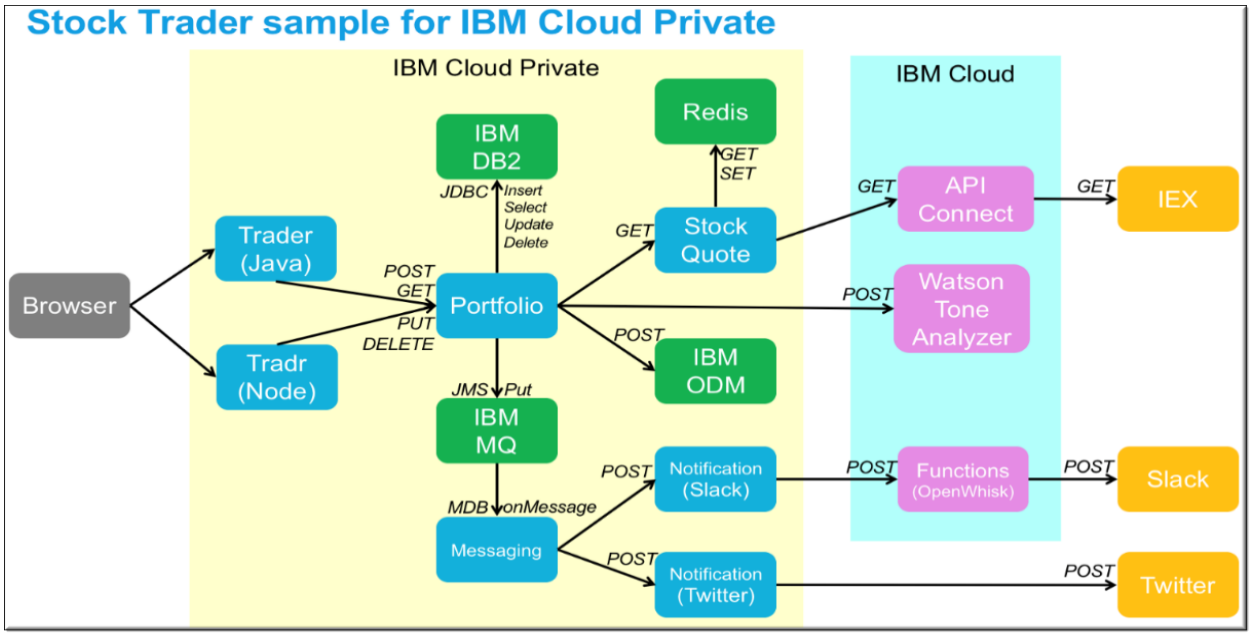
Given the increased number of “shares sold per month” as well as “traders per month” Boatswain executives can calculate that the $15,600 USD “opportunity cost” of those free trades and free consultations is a very good trade-off for increasing their revenue overall at a much higher rate than the cost. Besides that, free trades do not really cost Boatswain Trading in actual dollars, but it can be written off as opportunity cost on the books.

2.5 Run the Stock Trader "Before" application

Since 2015, software development and distribution methods and deployment have been changing rapidly. The Stock Trader application that we show here was built on a microservices-based architecture that allows loosely coupled services to work together and allows components to be upgraded incrementally, as opposed to traditionally with monolithic application development.

The sample Stock Trader application architecture is displayed below showing interaction between various microservices. The infrastructure services Db2, MQ, Redis and ODM are shown in green boxes, whereas microservices are shown in blue. IBM Cloud APIs are shown in pink and messaging endpoints are in orange.

The main advantage of microservice-based application is the ability to modify, deploy, update and scale services dynamically and individually. In a modern microservice mesh architecture, it is possible to run multiple versions of the same service in production and control the traffic dynamically based on rules that can be defined outside of the service itself, eliminating the need to need to modify any of the microservice code.



The Stock Trader application shown here is simply a demo application that does not address all features and functions of a commercial application, but does give insight into the concept of application modernization. The IBM Cloud Private platform provides tools to help you migrate your monolithic applications to a microservices-based architecture, free of cost.

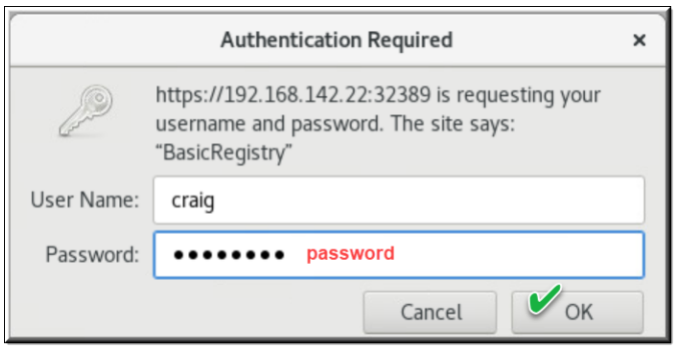
If your journey on this path has not yet begun, share this application with your development teams. It will provide your organization with the framework you need to start building your own microservice-based applications. Find it at: https://github.com/IBMStockTrader

	Another advantage of a microservices-based application is that it allows polyglot services to be written in the language of choice by the developers. This allows services and applications to work together easily. Developers are no longer forced to change their preferred language or learn a new language due to monolithic legacy applications.

Let's launch this application.

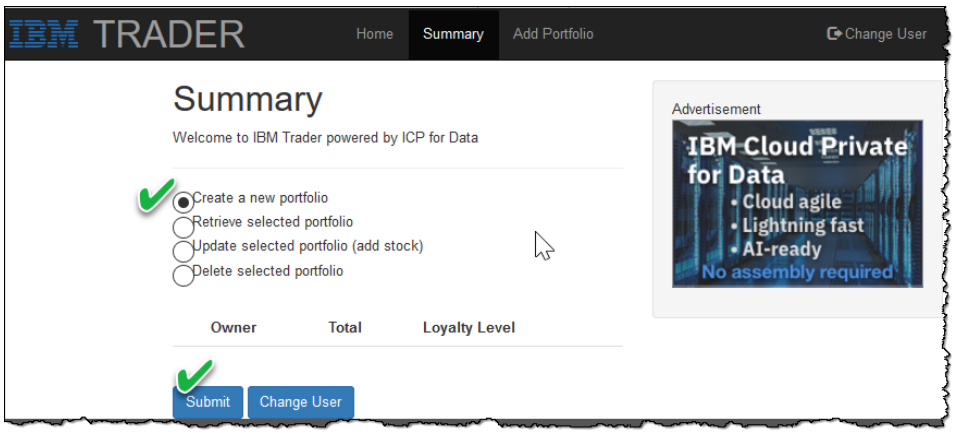
__21. On your workshop desktop, find and double-click on the icon: `Stock Trader Before`

__22. A login popup text box displays – enter User Name: `craig` and Password: `password`.

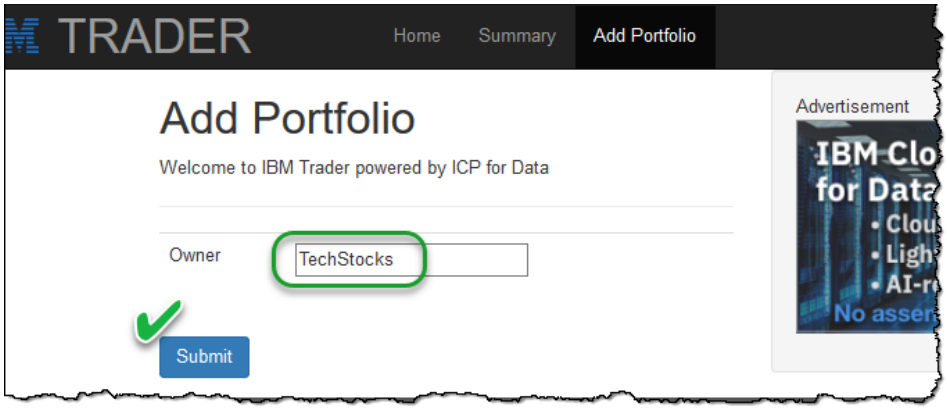


NOTE: The first time the Stock Trader application is launched it may take a little longer to start than subsequent launches.

__23. Select `Create a new portfolio` and click `Submit`.

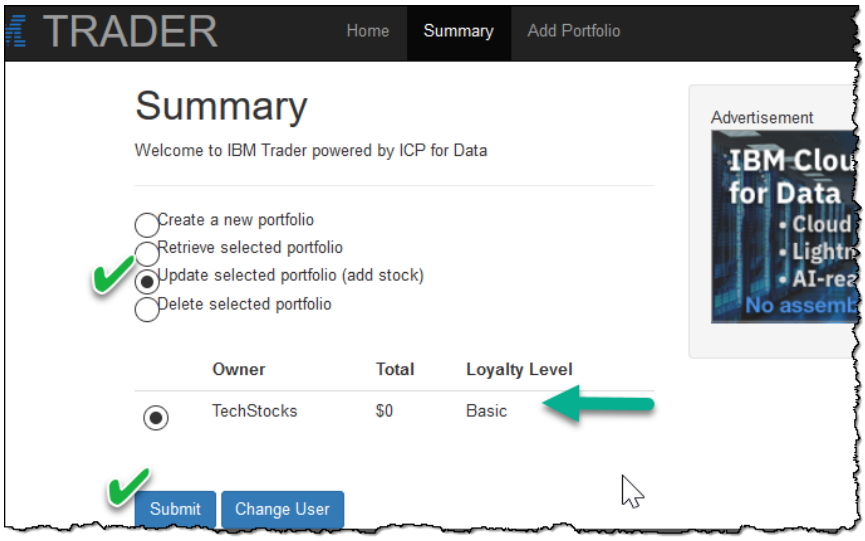


__24. Type `TechStocks` as name of the portfolio and click `Submit`.

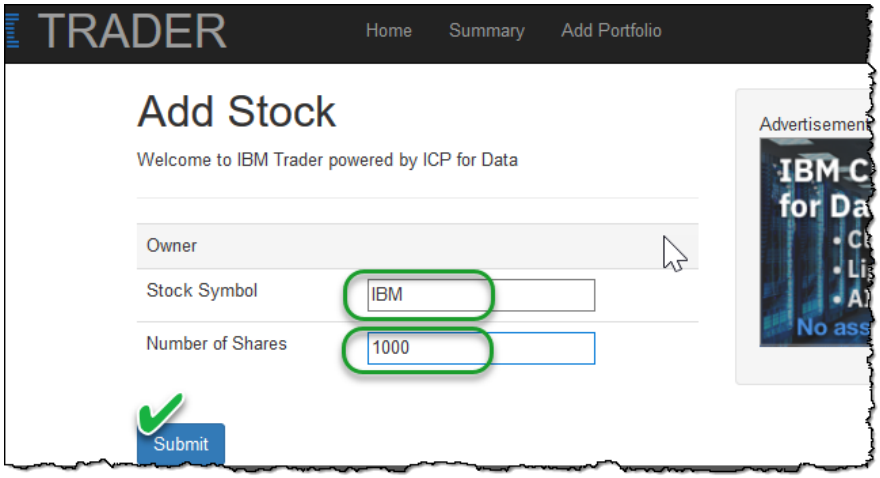


__25. Notice Loyalty Level is Basic.

Select `Update selected portfolio` and click `Submit`.

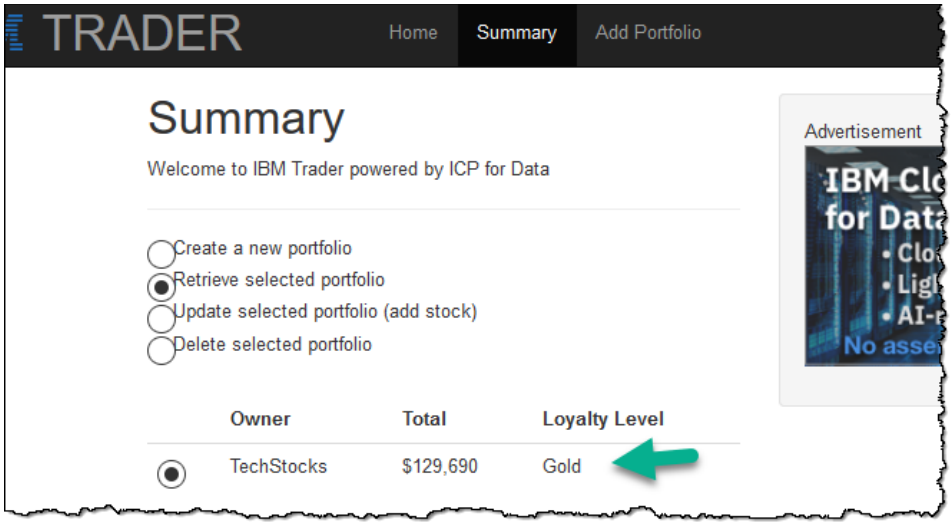


__26. Type `IBM` and `1000` and click `Submit` to buy 1000 stocks of IBM.



	If internet access is available – the stock quotes are taken from quandl.com using the IBM API Connect through IBM Cloud. The cached values from Quandl are stored in a Redis database.

__27. Notice that the loyalty level changed to Gold.



__28. Final note: as part of this application, there are “Ingress” rules defined such that the application’s `notification-service` can trigger a notification to either `Slack` or `Twitter`.

In this case, the `notification-service` sent our stock purchase transaction to Twitter (if the internet is available to this service) - you can check it out by going here: `https://twitter.com/ibmstocktrader`.

__29. Close the `Stock Trader Before` browser window to finish the review of this application.

	The Boatswain Trading goal is to enhance their Stock Trader application with analytics modernization to increase revenue and profits. They did so and named their new application: `Stock Trader After` This modern Stock Trader application was built on the modern IBM Cloud Private platform uses agile technologies with a modern CICD pipeline to implement changes rapidly.

2.6 Run Stock Trader "After" application

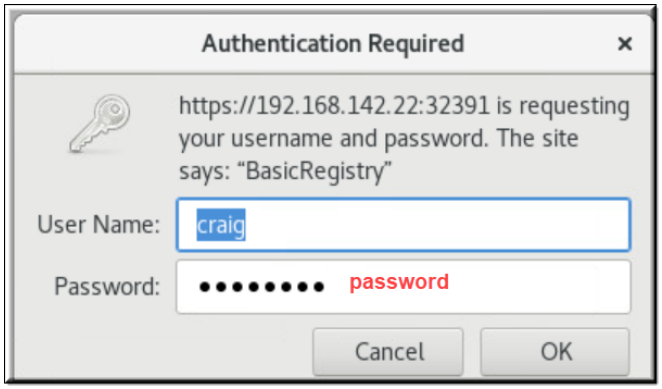
Companies that are able to keep pace with the technological breakthroughs will have a competitive advantage over companies that do not. (Think Netflix vs. Blockbuster)

Modernization of your application is the essence of ICP, and CPD goes one step further – it allows you to use your organization's data to modernize its analytics capabilities to infuse ML / AI into those applications.

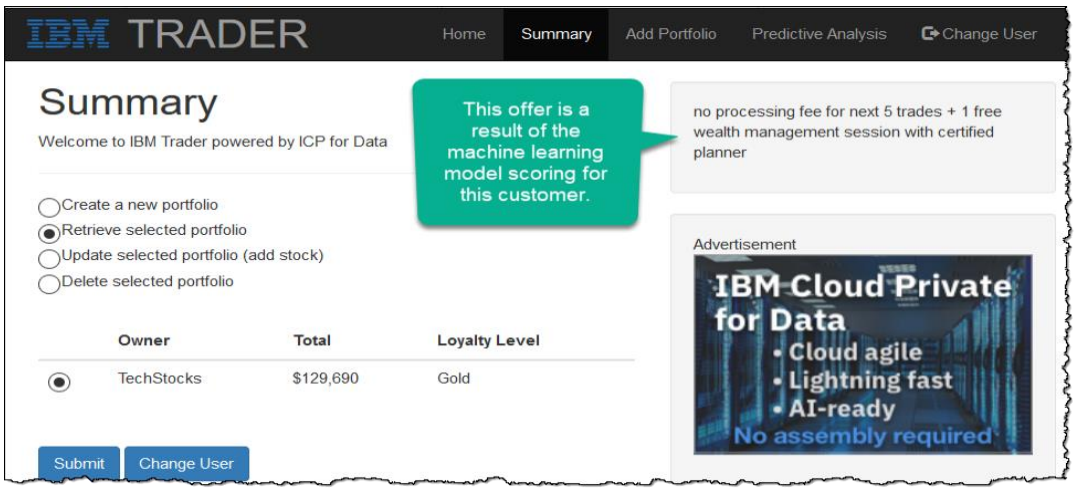
__30. So, let's look at the finished product of the newly modernized, ML-infused application `Stock Trader After`.

Find and double-click desktop icon: `Stock Trader After.`

__31. Log in using Username: `craig` and Password: `password` ⇨ `OK`



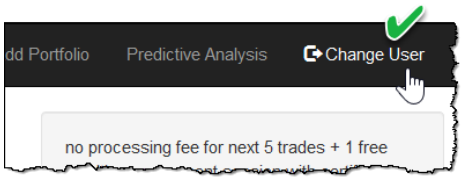
__32. Notice the offer given for this user. (Note: Your offer may vary)



Boatswain Trading used a predictive model that looks for the score of the user `craig` and returns a number of parameters. Our new predictive analysis microservice uses a single parameter and transforms it into an offer to the customer based on the "separation risk" that the model predicts.

How this works will become more evident as you work through this workshop.

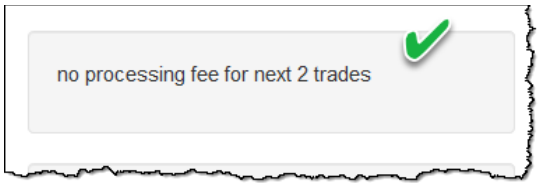
__33. Click `Change User`.



__34. Specify credentials: `dan` and `password`.

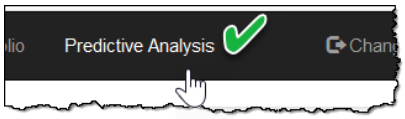
__35. Notice the personalized award offer.

You can try other users like jeff, anjali, vikram (all have the same password)



__36. Let's dive deeper behind the scenes of machine learning model prediction.

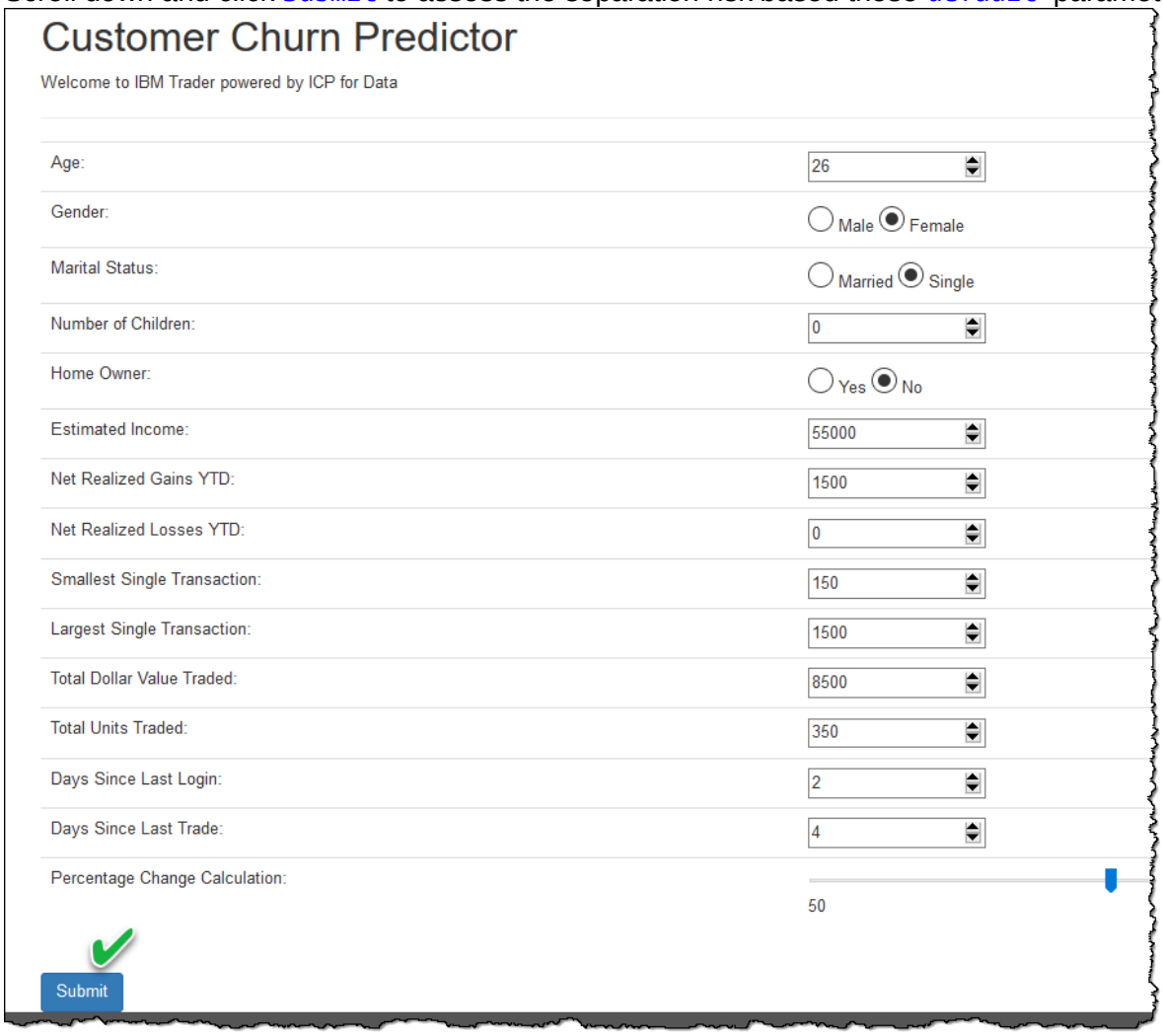
Click `Predictive Analysis`.



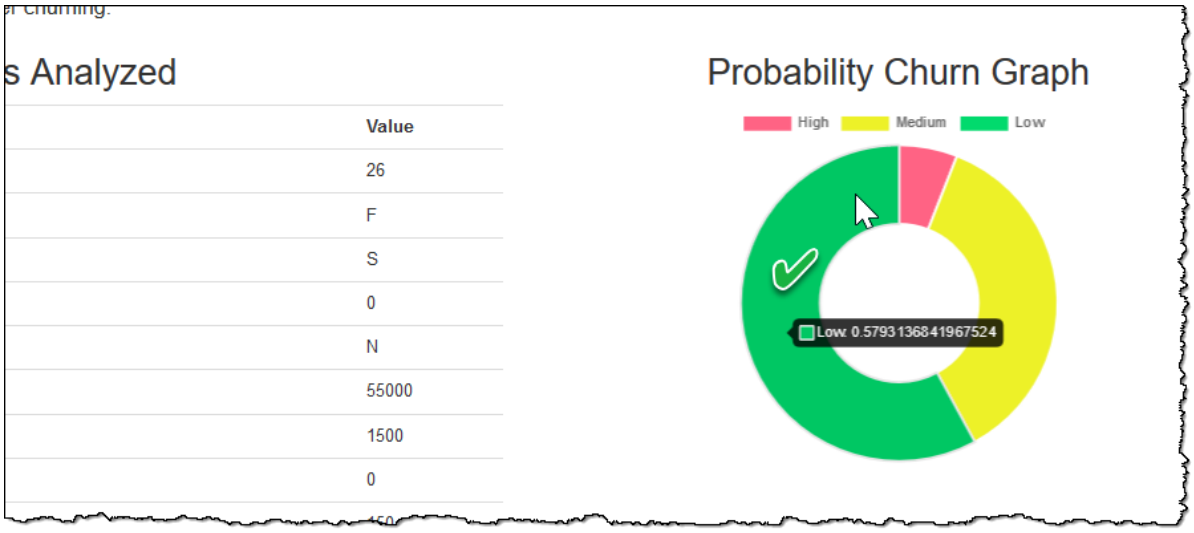
__37. This screen is built to directly interact with the machine learning model you will create in the workshop.

You can change the variables to see how those specific sets of variables impact predictions, but in your first test just use the defaults.

Scroll down and click `Submit` to assess the separation risk based these `default` parameters.

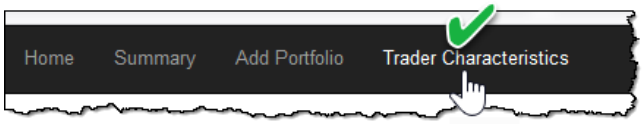


__38. This is the result using the default values of separation risk predictor. In this case, there is a *Low* to moderate risk of this customer leaving. This customer has a profitable portfolio.



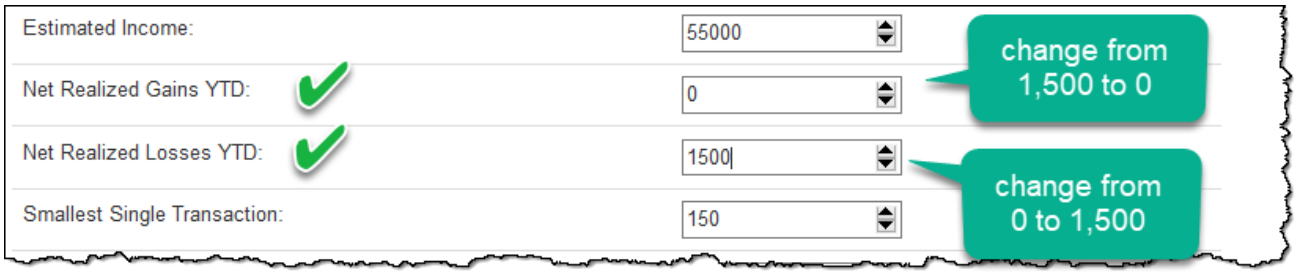
__39. Let's change some parameters.

Click `Trader Characteristics`.



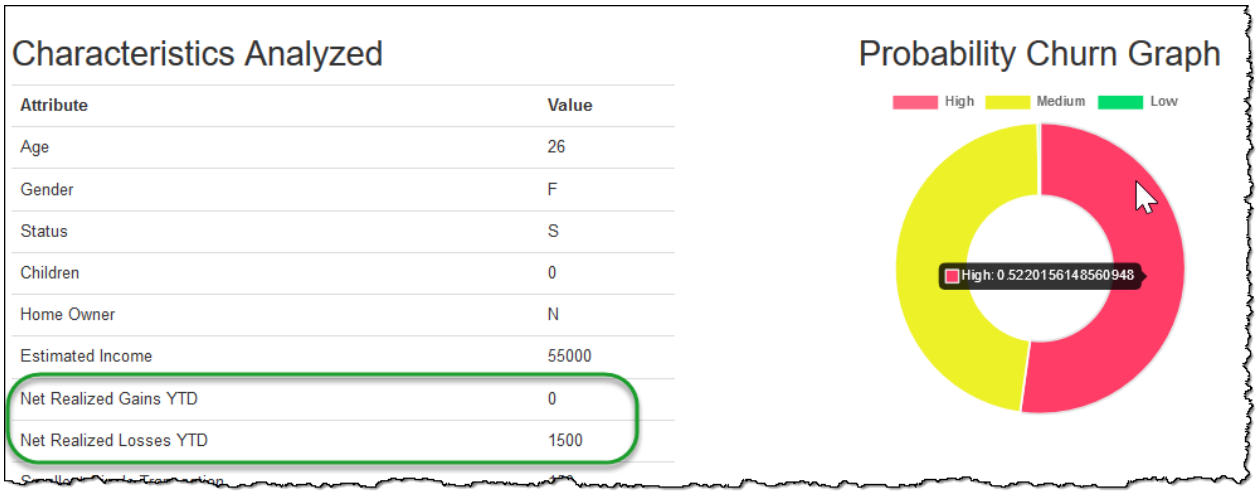
__40. Change `Net Realized Gains YTD` from `1500` to `0` and change `Net Realized Losses YTD` to `1500` from `0`. This represents a portfolio that is not profitable.

Scroll to the bottom and click `Submit`.



__41. Notice the probability of separation risk of this customer. The risk has changes from `Low` to `Medium` given the fact that portfolio of this customer is no longer profitable.

In fact, the details show that this customer is on the cusp of being `High` probability of churn.



2.7 Lab conclusion

Regardless of your problem classification domain, CPD provides the tools needed to access, clean, shape and govern your data in preparation for ML model development, release and continuous improvement framework.

From automatic ML model generation (no coding) to comprehensive tool sets to develop complex models (with coding), CPD provides the tools necessary to produce your specific ML model classification.

Once ML models are deployed, they are easily consumed by applications as RESTful-compliant services. The deployed models are easily scalable to meet usage demands as they are packaged as independent scalable docker containers.

As probabilities are consumed from one or more models, application logic can be applied to allow applications to make intelligent business decisions based on a set of predefined business rules and workflows.

You will next dive deeper into the **Collect**, **Organize**, **Analyze** and **Deploy** capabilities of CPD as an integrated data platform.

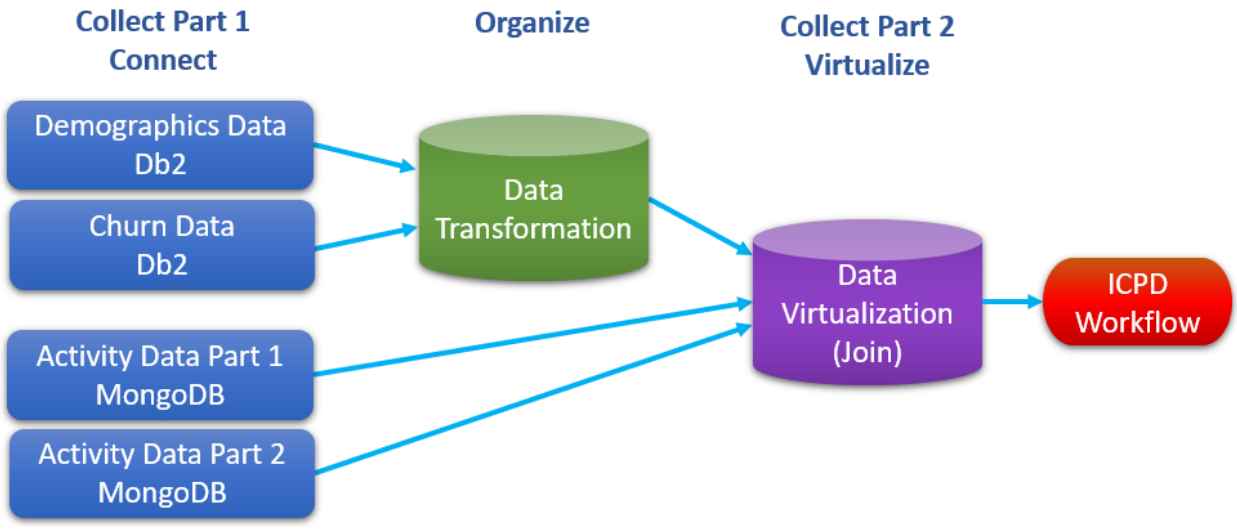
**** End of Lab 02: Executive Demo**

Lab 03 Collect Part 1 - Connect

3.1 Lab overview

The Collect capability of CPD means accessing your organization's data regardless of where it resides, whether that be a in-cluster data source (e.g. the Db2 Warehouse), a Federated data source in a remote location, or a native connection to a remote data source. You can even use the Db2 Event Store add-on feature to provide streaming access that is suited for Internet of Things processing. Additionally, Data Virtualization and Data Transformation are available real-time to streamline the access, performance, and formatting of the data for use in later steps of the CPD analytics workflow.

In this lab you will explore creating Connections and data sets for the Collect process. In later labs you will explore Data Transformation and Virtualization.



3.2 Persona represented in this lab

The Data Engineer persona is the likely role to perform the various Collect tasks shown in this lab.

Persona (Role)	Capabilities
 Data Engineer	Data Engineers build and optimize the systems to allow data scientists and business analysts to perform their work. The Data Engineer ensures that any data is properly received, transformed, stored, and made accessible to other users.

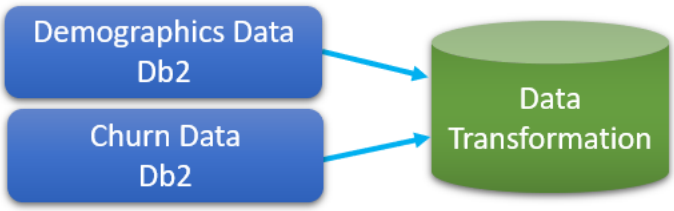
For the sake of simplicity, you will remain logged in the CPD console throughout this workshop as the user Admin, which has been granted all persona roles. This was done so that you will not be required to log off and log on again as different users to represent the varying personas as you make your way through each lab in this workshop.

In your organization however, it is likely that once you have a mature CPD environment set up, separation of duties will be defined by persona where different users will be assigned one or more personas to do their particular tasks.

3.3 Db2 data overview – Transforming for analytics

The Db2 data in our scenario (Demographics and Churn) have two key factors that make an appropriate scenario for `Data Transformation`:

- The data only changes once a month (it is relatively static). Thus, copying it and/or changing it for downstream processing in our CPD analytics workflow is OK because we are not required to have the absolutely latest data to get the results we need.
- The data has to be changed before it can be processed in our CPD analytics workflow.

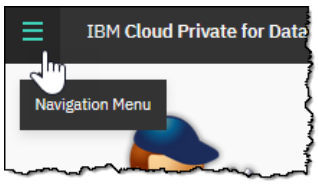


Although you will be doing the actual `Data Transformation` in the next lab, you will prepare for it in this one by doing some `Collect` steps.

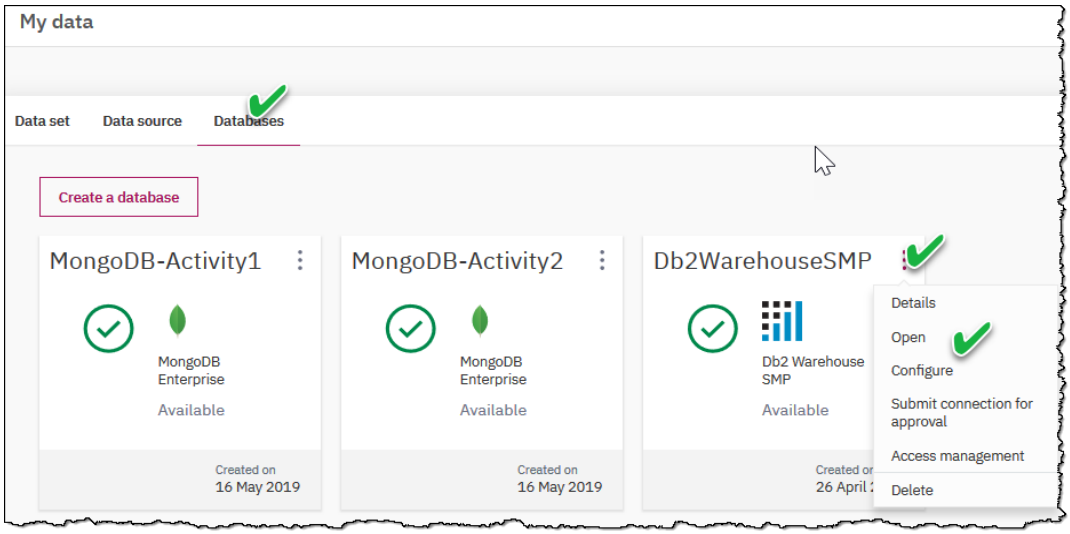
3.3.1 Review the Db2 data

__42. To review this data, you will return to the *Db2 Data Server Manager* (DSM) console that you encountered in our first lab.

Start at the `Navigation Menu`

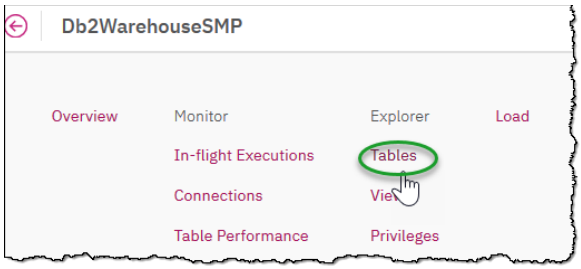


Click `Collect` ➪ `My Data` ➪ `Databases` ➪ `Db2WarehouseSMP` ➪ `ellipsis` ⋮ ➪ `Open`

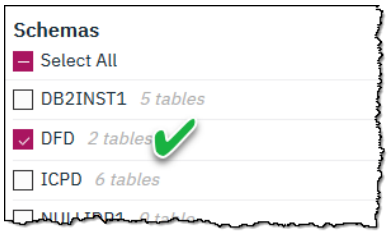


__43. At the top right of the Db2 *DSM* console, click on the hamburger `Menu` icon.

Then choose the option `Tables`

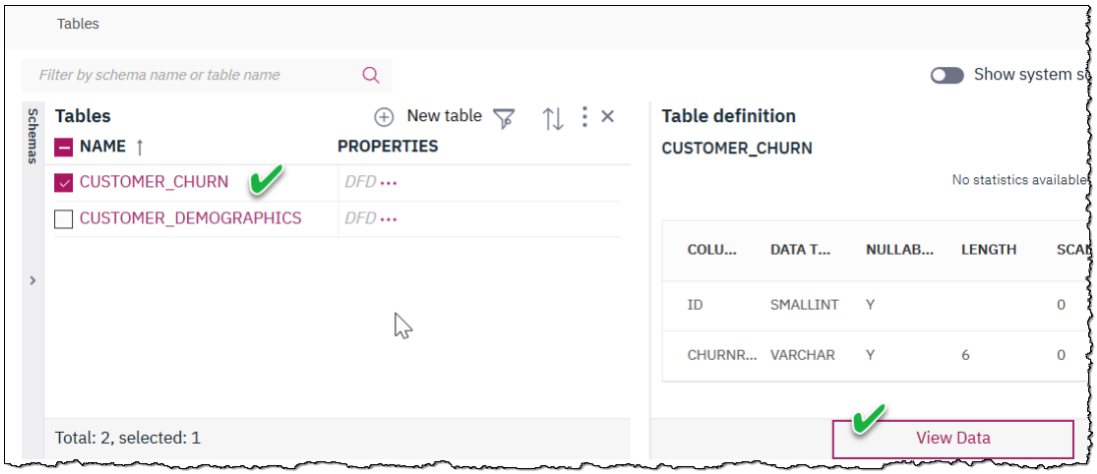


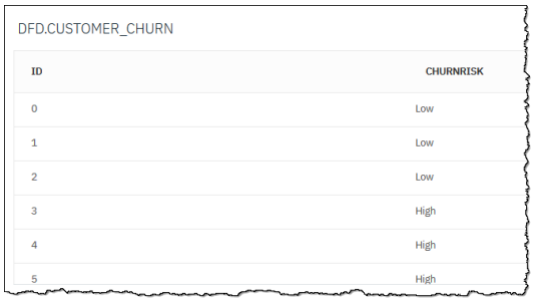
__44. Select schema `DFD`



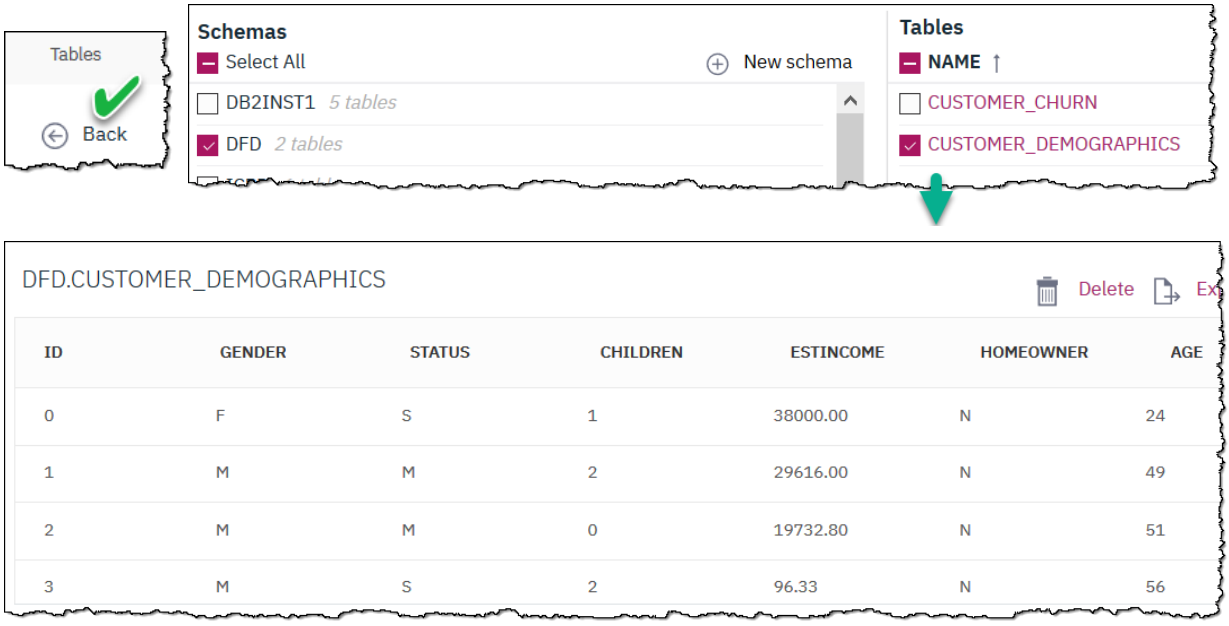
__45. Select table `CUSTOMER_CHURN` and click on it to bring up the table definitions view.

Then click on `View Data`





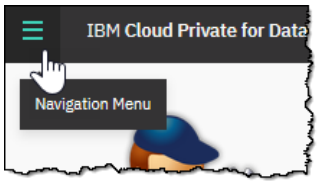
__46. Click on the `Back` icon to view the data for table `CUSTOMER_DEMOGRAPHICS`



3.3.2 Review the Db2 connection

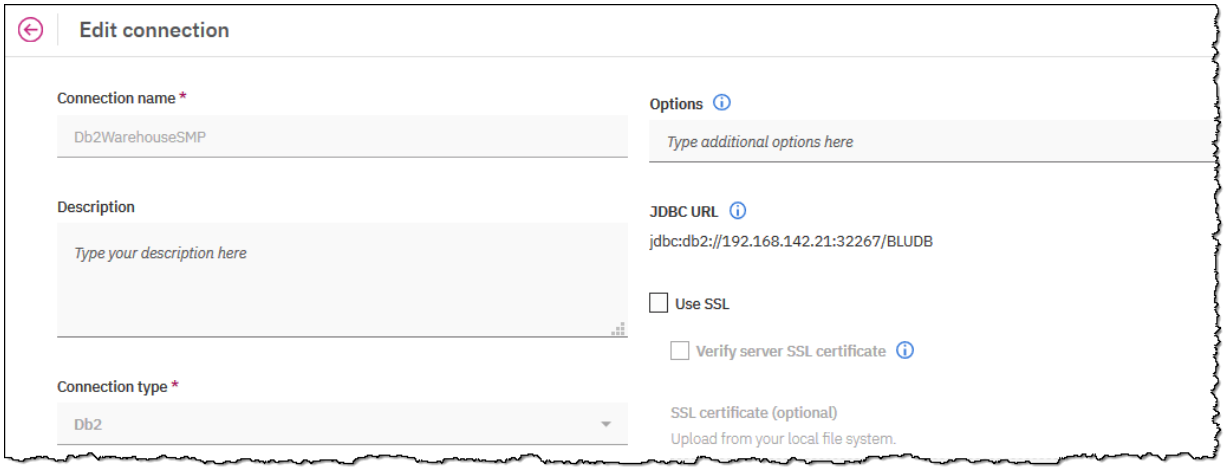
__47. Review the connection to the Db2 Warehouse that we updated in our first lab.

Start at the `Navigation Menu`



__48. Click `Connections` ⇨ `Db2WarehouseSMP.`

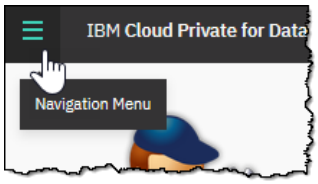
Click on the connection to review it.



3.3.3 Add the Db2 datasets

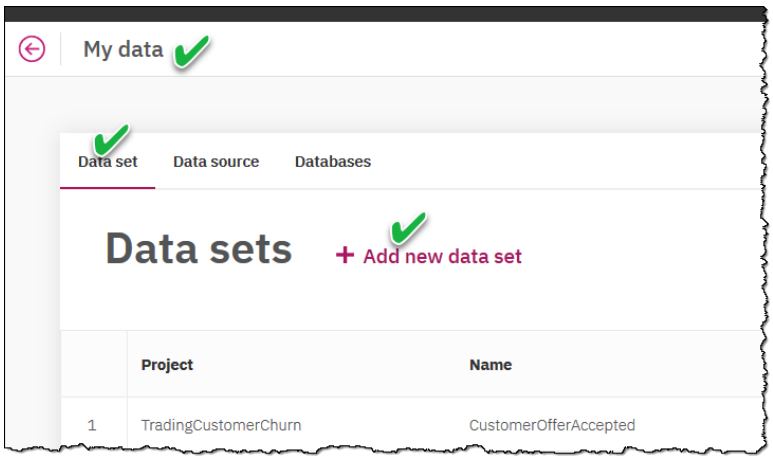
__49. Review the connection to the Db2 Warehouse that we updated in our first lab.

Start at the `Navigation Menu`

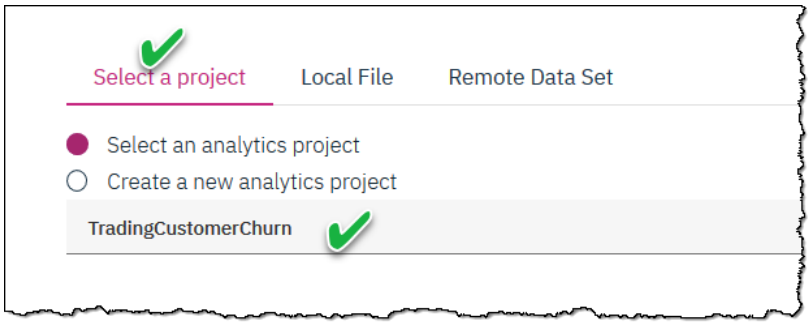


__50. Click `Collect` ⇨ `My data`

From the `Data set` tab, click `+ Add new data set`

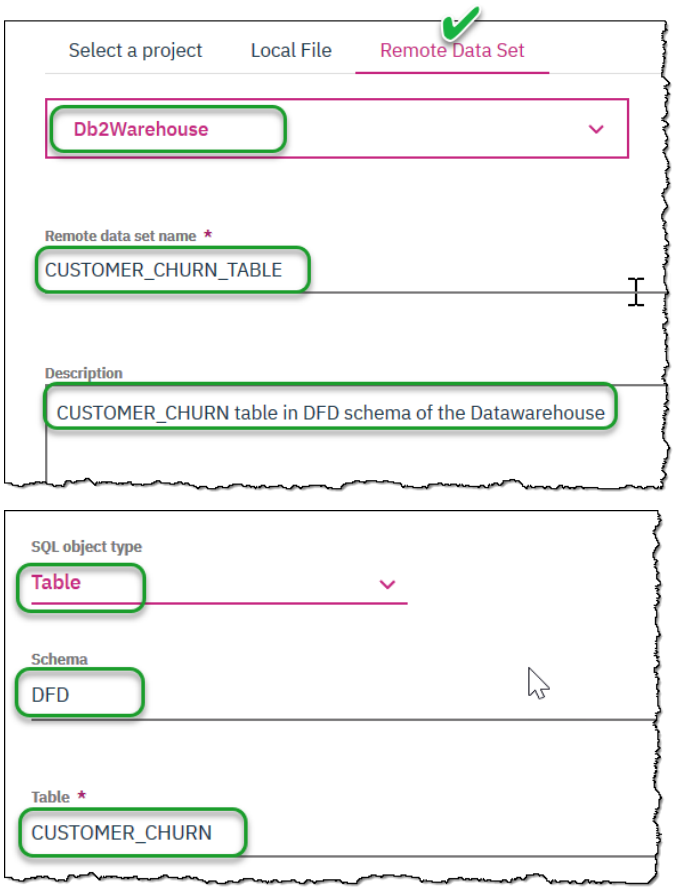


__51. From `Select a project`, choose `TradingCustomerChurn`



__52. Click the tab `Remote Data Set` and fill in like this:

Select a source:	`Db2Warehouse`
Remote data set name:	`CUSTOMER_CHURN_TABLE`
Description:	`CUSTOMER_CHURN table in DFD schema of the Datawarehouse`
SQL object type:	`Table`
Schema:	`DFD`
Table:	`CUSTOMER_CHURN`



__53. Click `Save`

__54. The dataset is added, now add another.

Click `+ Add new data set`

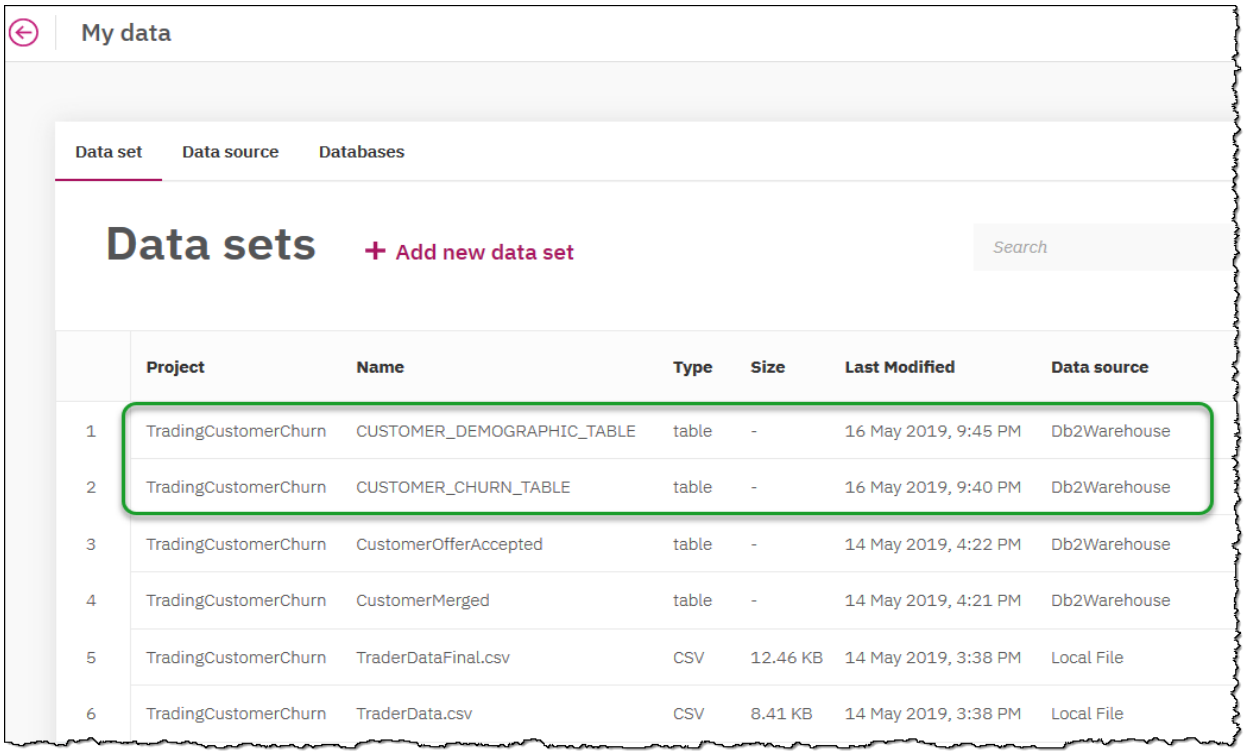
__55. Click the tab `Remote Data Set` and fill in like this:

Note: Please name the table exactly as shown before saving.

Select a source:	`Db2Warehouse`
Remote data set name:	`CUSTOMER_DEMOGRAPHICS_TABLE`
Description:	`CUSTOMER_DEMOGRAPHICS table in DFD schema`
SQL object type:	`Table`
Schema:	`DFD`
Table:	`CUSTOMER_DEMOGRAPHICS`

__56. Click `Save`

__57. The `My data` screen now shows your new data sets.



Now that we have these data sets defined, we could use them anywhere in our CPD downstream processing.

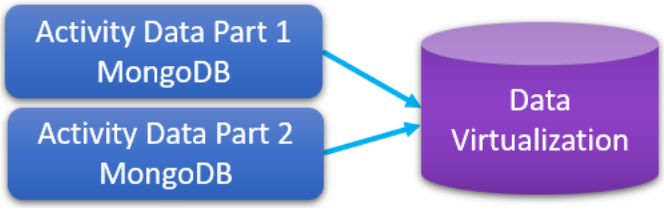
Data Engineer

The `Data Engineer` frequently works with connections and data sets to many different types of data sources.

3.4 MongoDB data overview – Virtualizing for analytics

The Mongo data in our scenario (Activity) has two key factors that makes for an appropriate scenario for `Data Virtualization`:

- The data is constantly changing (it is dynamic)
- Our analytics processing requires the absolute latest data



We have chosen semi-structured data in this scenario to demonstrate the power of CPD `Data Virtualization` – it even works with JSON documents. But we could just have easily chosen a structured database source for it as well, like Db2 or Oracle, SQL Server, and so on.

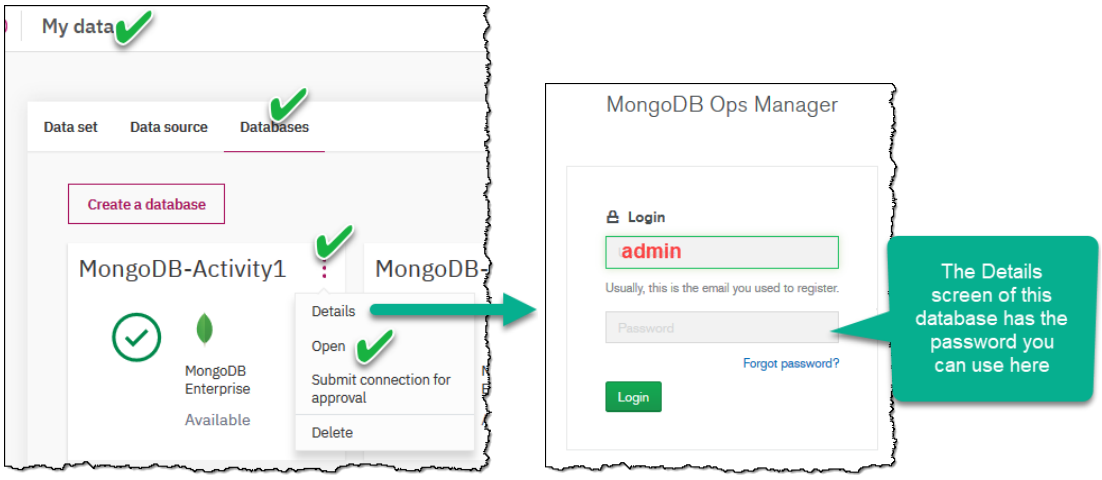
Semi-structured (and even unstructured) data is commonly used in many systems of engagement applications, so this was another reason we chose this as an example data source type. Our scenario presumes that this data comes from a mobile application that will be constantly changing the data, and we require the latest for our analytics workflow.

3.4.1 Review the MongoDB data

The Mongo databases were provisioned after the MongoDB Enterprise add-on was installed, which are also located in `My data`.

__58. Start at the `Navigation Menu`

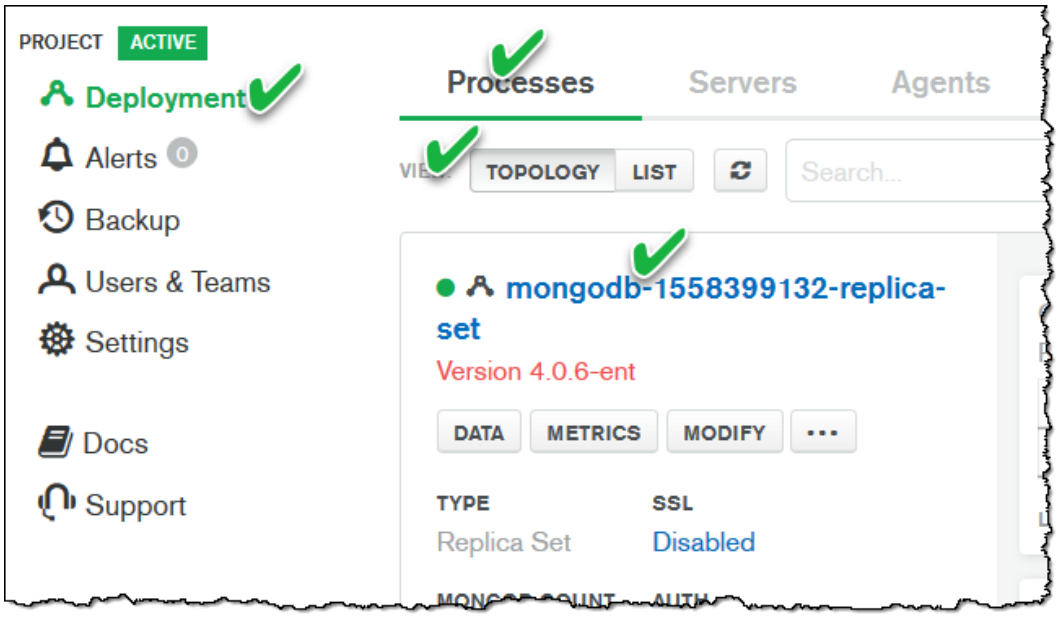
Click `Collect` ⇨ `My data` ⇨ `Databases` ⇨ `MongoDB-Activity1` ⇨ `ellipsis` ⋮ ⇨ `Open`



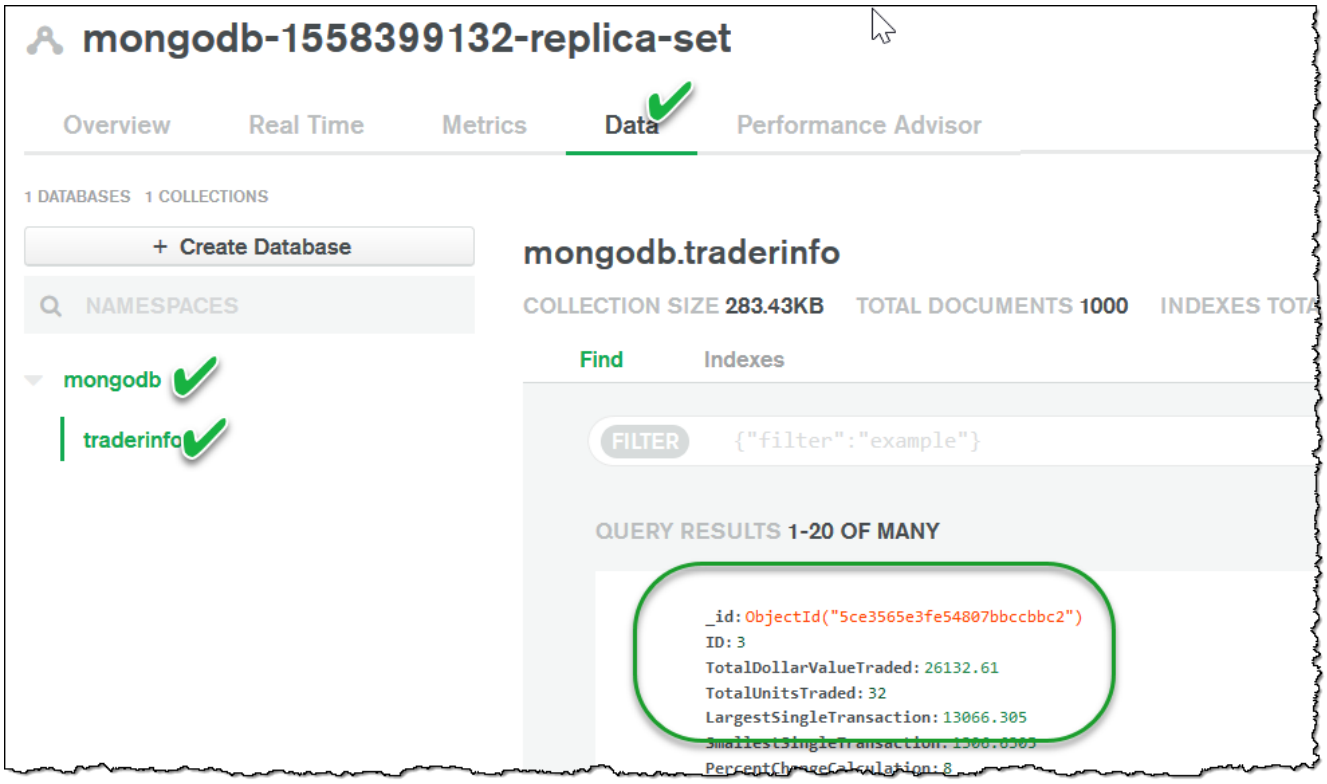
Data Engineer	The very first time you log into this *MongoDB Ops Manager* console it will ask you for user and password credentials. We have saved these for you in a browser setting. However, should you encounter this screen, you can find the password by going back and reviewing the `Details` option, shown in the screenshot above. Simply copy and paste the password from the `Details` screen to perform this console login. By the way, the user is always `admin`.

__59. In the *MongoDB Ops Manager* console, click `Deployment` on the top left of the screen.

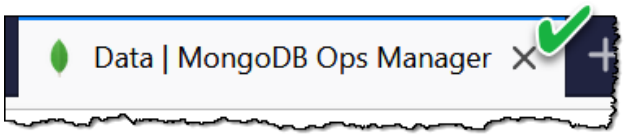
Then in the `Processes` tab, under `TOPOLOGY` section, click on the `Replica Set` link



__60. Click on the `Data` tab to review the JSON documents in database `mongodb`, collection `traderinfo`



__61. Close the *MongoDB Ops Manager* tab when finished

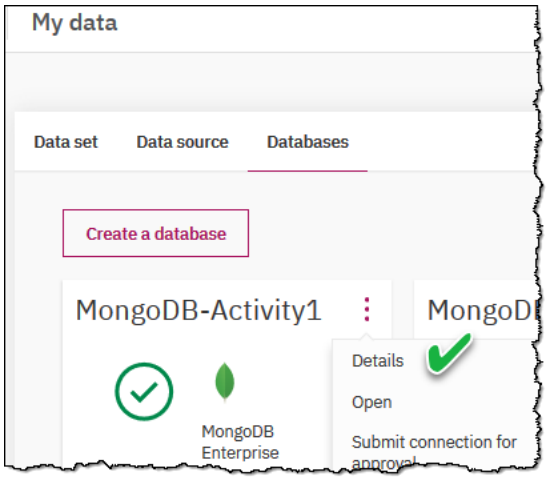


__62. Do the same and review the data in the `MongdoDB-Activity2` database

3.4.2 Create the MongoDB connections

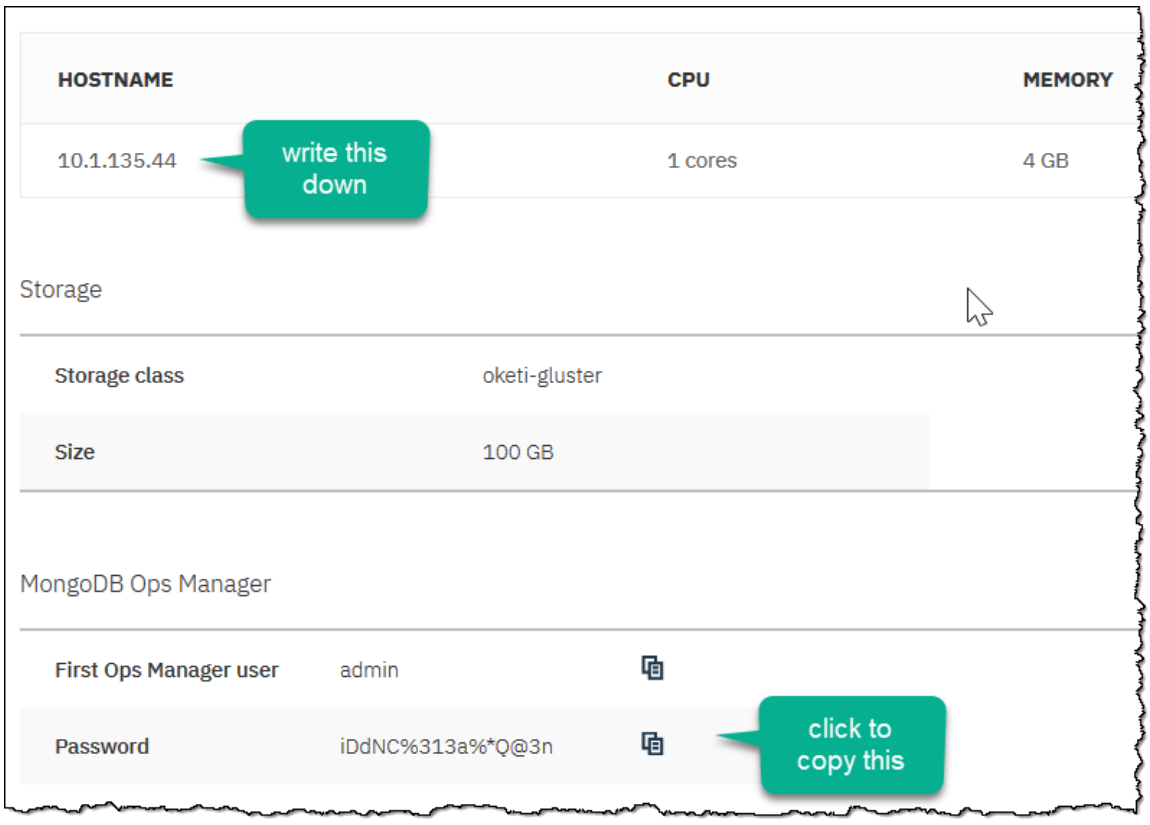
Now let's create connections for the MongoDB databases to be used in our CPD workflow. Before creating a connection, get the information you need for it from the Details screen for that database.

__63. In `My data`, click on the `Details` of the database `MongoDB-Activity1`



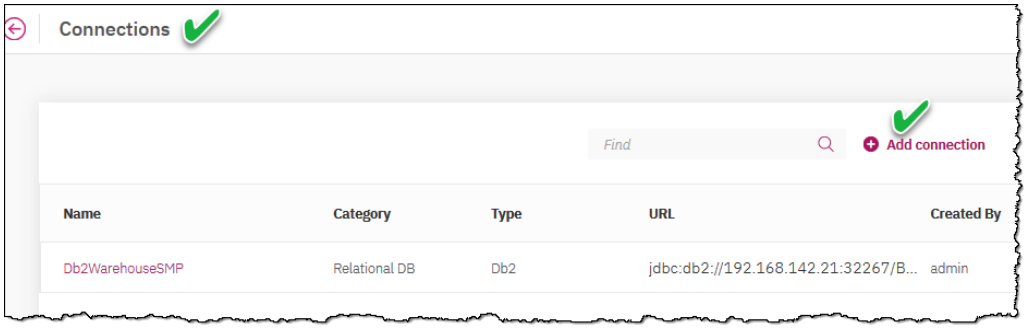
Make note of the host name - write it down (note: the port is always `27017`)

Click copy next to the password (note: the user is always `admin`)



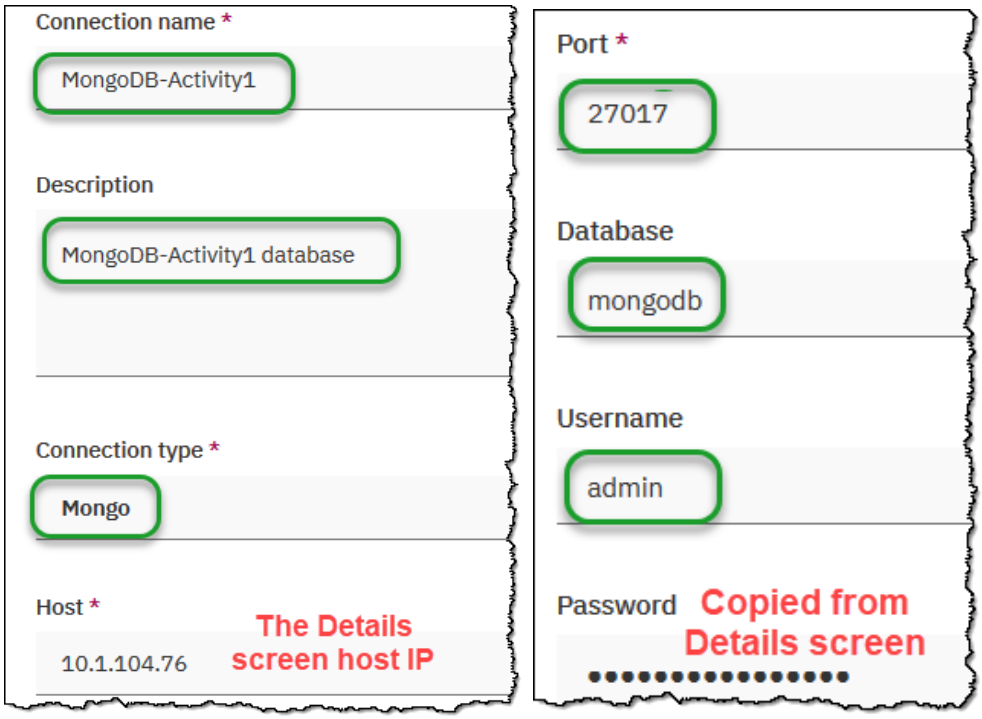
__64. Next start at the `Navigation Menu`

Click `Connections` ⇨ `+ Add connection`

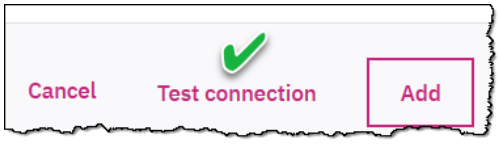


__65. Fill in the details of this screen like this:

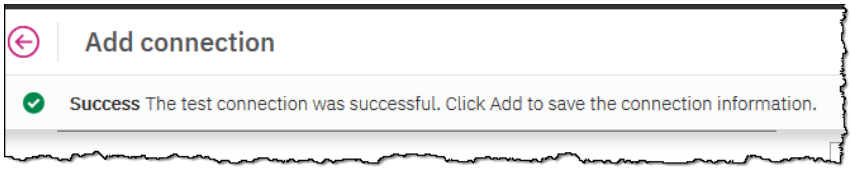
Connection name: `MongoDB-Activity1`
Description: `MongoDB-Activity1 database`
Connection type: `Mongo`
Host: Host IP you wrote down from the Details screen of the database
Port: `27017`
Database: `mongodb`
Username: `admin`
Password: Copied from the Details screen of the database



__66. From bottom of this connection edit screen, `Test` the connection:



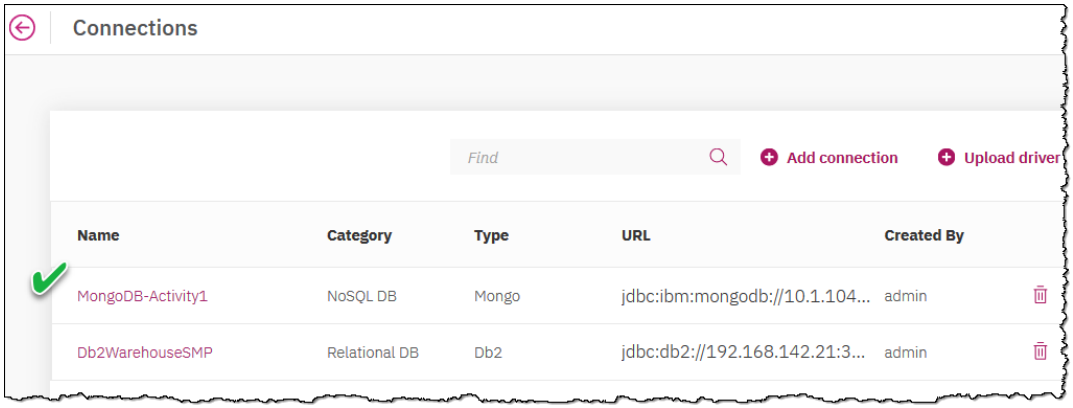
__67. The top of the screen should show a successful test.



Data Engineer	If your test is not successful, check all the credentials again, especially Host and Password.

__68. After a successful connection test, click `Add`

__69. Your `Connections` screen now shows a new connection with the Type: Mongo



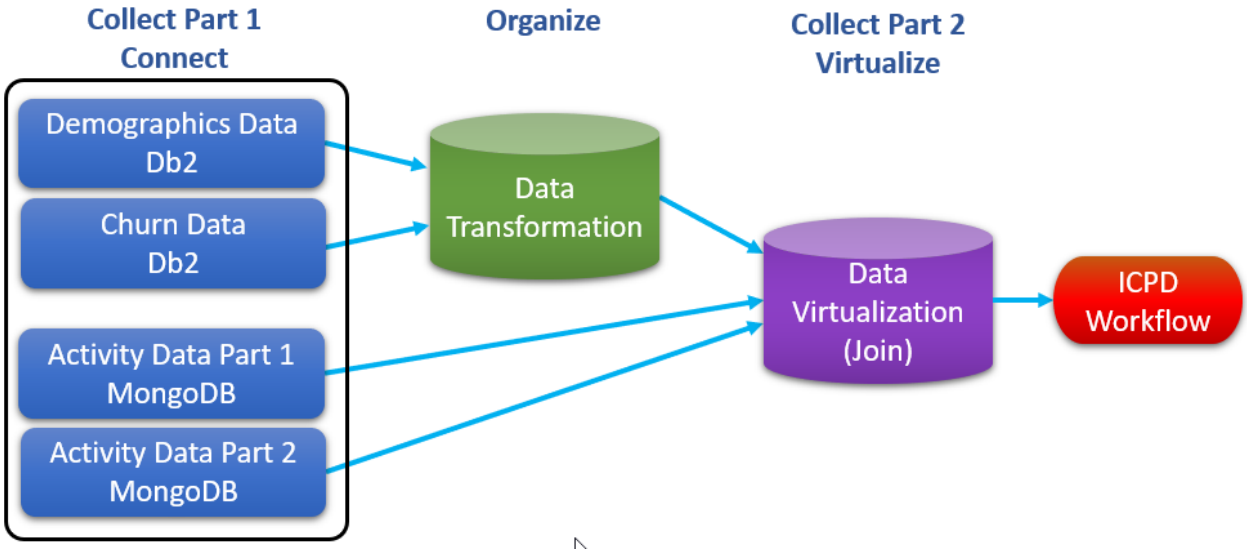
__70. Now create a connection for database `MongoDB-Activity2` making sure to get the correct `Host` and `Password` from the `Details` page for that database. (Hint: `My Data` ⇨ `Databases`)

__71. When you are done, click the twistie in `Connections` to sort them.

Make sure you created two different MongoDB connections and that you did not accidently create two connections for the same MongoDB database. The host/port combination should be different for each.

3.5 Lab conclusion

In this Collect Part 1 lab, you created connections for the Db2 and MongoDB data. This prepares you for the Organize and Collect Part 2 Virtualize steps in our workshop.



In the `Organize` lab coming up, we will be transforming the Db2 data sets into one.

After that, we will finish up the `Collect` processing by virtualizing the results from both the Db2 Transformation output and the MongoDB data together.

**** End of Lab 03: Collect Part 1 - Connect**

Lab 04 Organize

4.1 Lab overview

Many organizations find it difficult to understand their own data because it originates from many sources, is dispersed across many silos, and is controlled by different teams.

This `Organize` lab will show you how to uncover the hidden data in your organization's data and how to build a lineage that is otherwise difficult to establish. Cloud Pak for Data helps your organization move from the manual processes required to establish relationships between data, to an automated one aided by the platform's built-in machine learning capabilities.

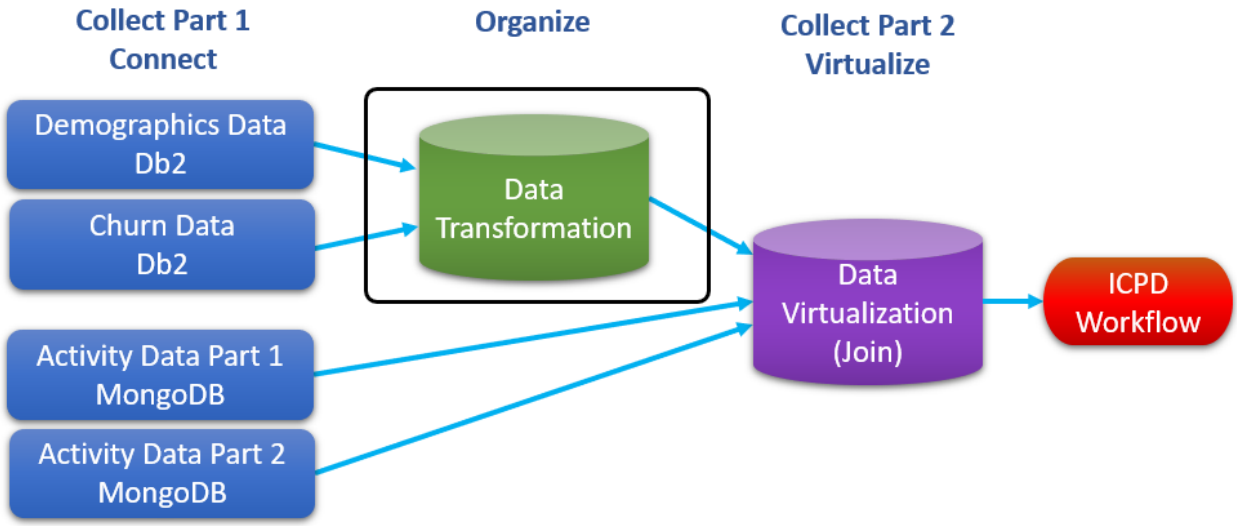
4.2 Persona represented in this lab

The `Data Steward` persona is the likely one to perform most of the `Organize` tasks shown in this lab.

Persona (Role)	Capabilities
 Data Steward	Data Stewards brings integration and governance to the data.

Note that this persona often works closely with the `Data Engineer` persona because they both work with the data to prepare it for the analytics processing by other personas.

For example, in this lab one of things the `Data Steward` will do is to create a Transformation job with the Db2 data. In turn, the `Data Engineer` will then use that file to create a final virtualized table of all the data sources joined together.



Before we start transforming data, let's first explore the other crucial aspects of the CPD `Organize` capabilities, starting with creating a Business Glossary.

4.3 Create a business glossary

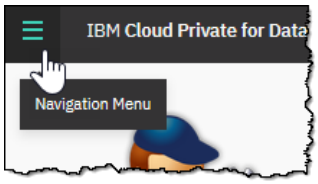
A business glossary consists of categories and terms.

Categories provide the logical structure for the glossary so that you can browse and understand the relationships among terms and categories in the glossary. Categories can be organized in a hierarchy based on their meaning and relationships to one another.

A term is a word or phrase that describes a characteristic of the enterprise. Terms are the fundamental building blocks of the glossary. Each term has a parent category, but it can also be referenced by other categories. When you create a term, you need to provide a meaningful name. Terms can be assigned to other terms, and to other asset types as well.

Let's explore building a business glossary in CPD.

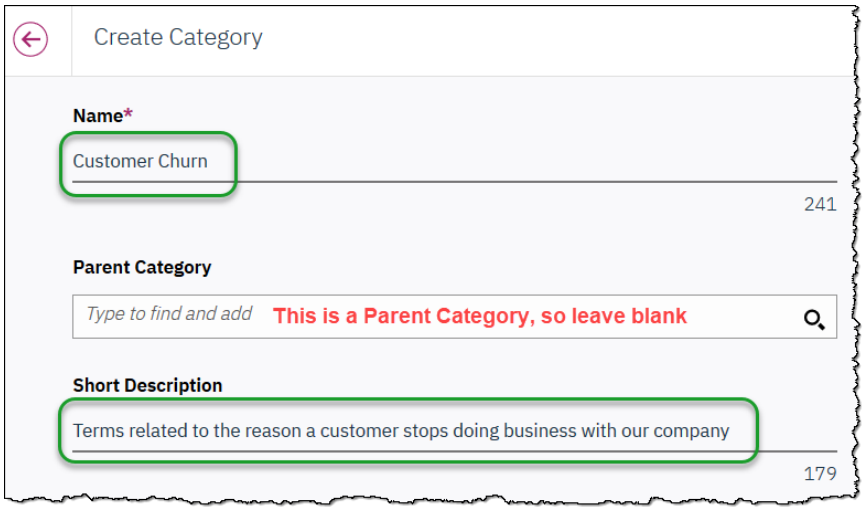
__72. Start at the Navigation Menu



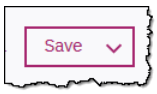
__73. Click Organize ⇨ Business Glossary

__74. Under the tab Categories ⇨ Create Category.

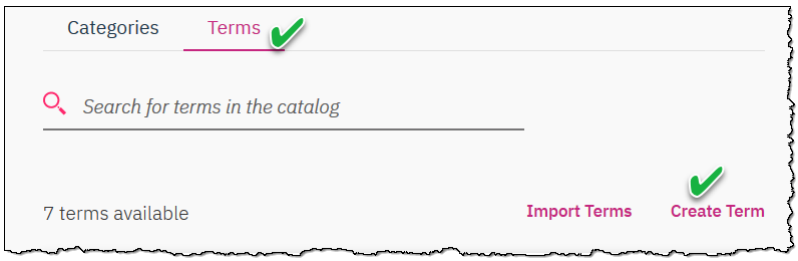
__75. Type *Name:* `Customer Churn` and *Short Description*: `Terms related to the reason a customer stops doing business with our company`



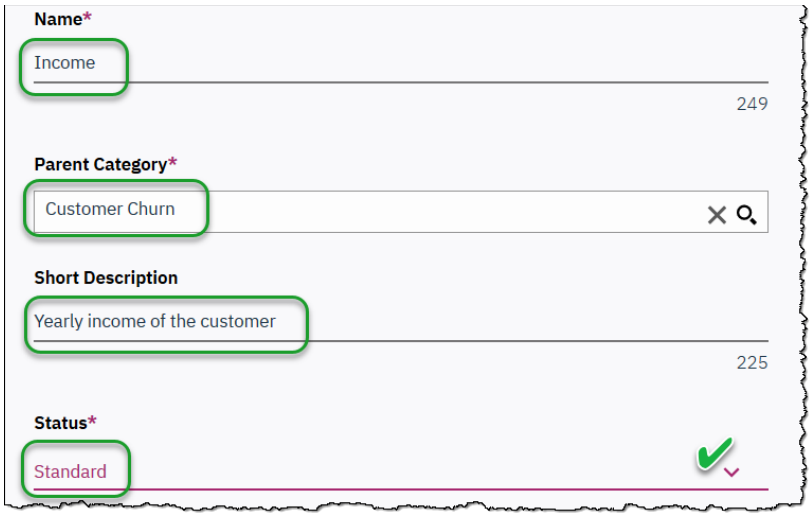
__76. Click `Save` (bottom right of the screen)



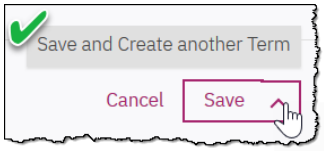
__77. Go to the second tab `Terms` and click `Create Term`.



__78. Create a term `Income` with a parent category `Customer Churn` (hint: type `C` in the box), provide a description as: `Yearly income of the customer` and choose *Status* as `Standard` from the drop-down menu.



__79. Click the down arrow next to `Save` and click `Save and Create another Term`.

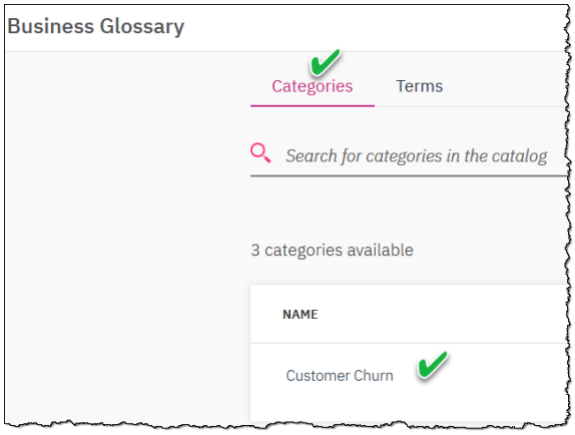


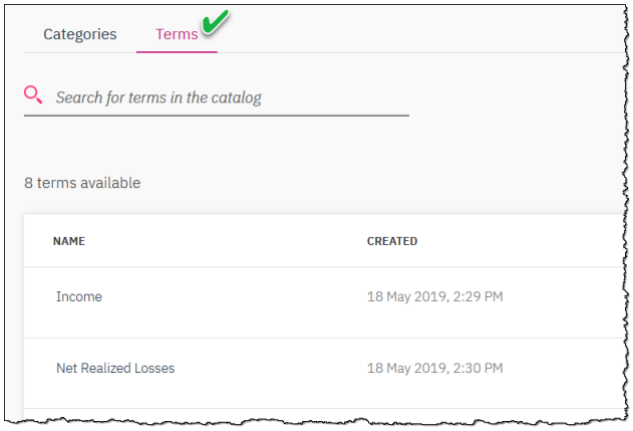
__80. Now, repeat above exercise to create the following `Terms` in the table below.

After each use `Save and Create another Term`, except for the last which you `Save`.

Name	Parent Category	Short Description	Status
Home Owner	Customer Churn	Flag indicating whether the customer owns a home.	Standard
Net Realized Losses	Customer Churn	The net dollar value of losses realized for a customer, usually a year to date metric	Standard
Net Realized Gains	Customer Churn	The net realized gains for a customer, usually a year to date metric.	Standard
Gender	Customer Churn	The gender of the customer, either 'M' or 'F'	Standard
Days Since Last Trade	Customer Churn	The number of days since the customer last executed a trade.	Standard
Total Dollar Value Traded	Customer Churn	The total amount the customer has traded with us since onboarding.	Standard

__81. Review your new Business Glossary `Category` and `Terms` related to that category





Data Steward

You can create your own glossary with categories and terms manually, or import them from a file. In addition, you can import Industry Models from IBM for industries such as finance, banking, healthcare, and insurance and import them into CPD.

See the add-ons screen  then `Industry Accelerators.`

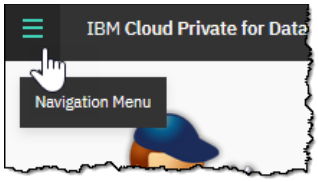
4.4 Create governance policies and rules

An information governance `policy` is a natural-language description of a governance subject area. It can contain multiple information governance sub-policies or reference one or more information governance rules. It must fulfill a business objective, and be relevant and understandable to all users of the policy.

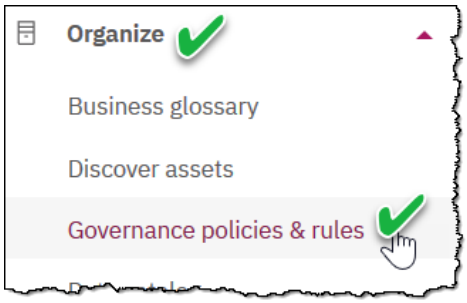
An information governance `rule` is a natural-language description of the criteria that are used to determine whether information assets are compliant with business objectives. Generally, information governance rules are derived from information governance policies and are more specific. The rules define the actions to take in specific situations to implement the policy. An information governance rule can be referenced by one or more information governance policies.

Let's explore creating these in CPD.

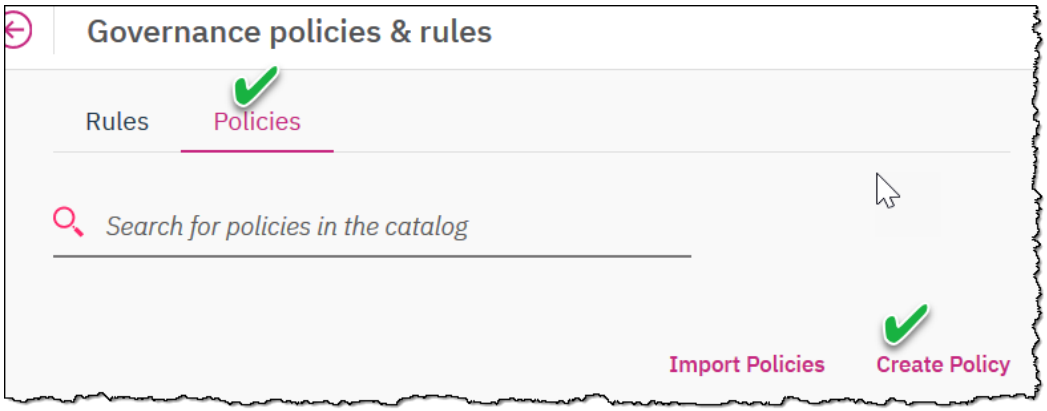
__82. Start at the `Navigation Menu`



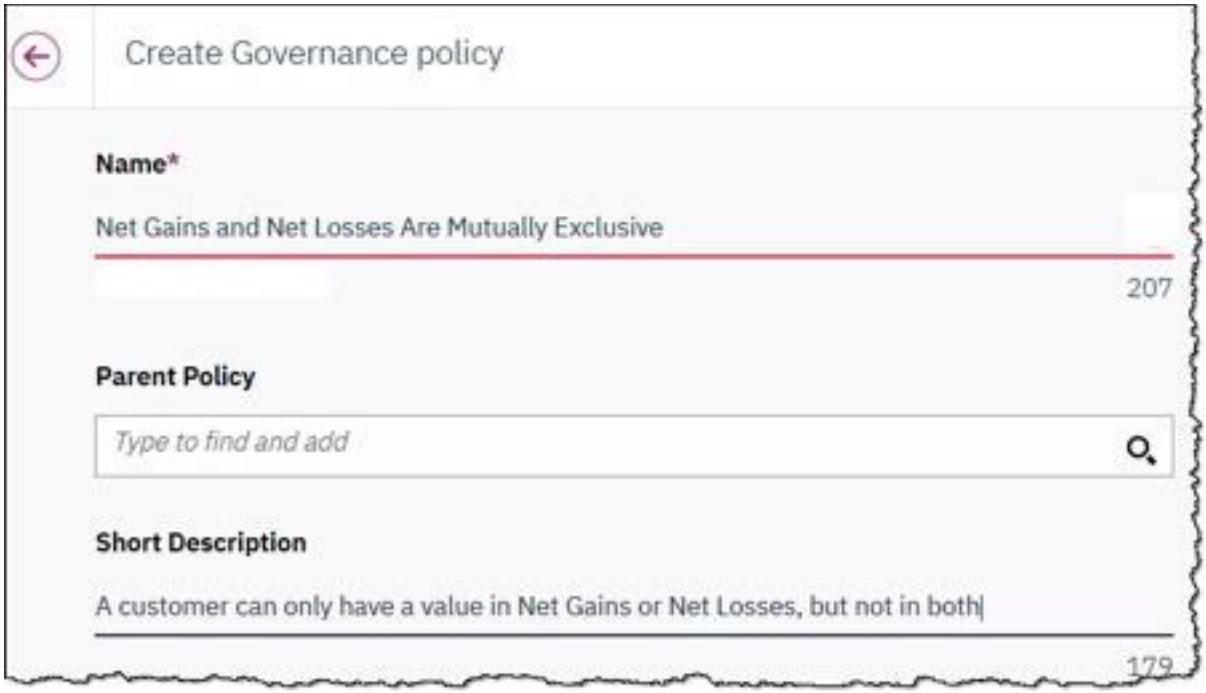
__83. Click `Organize` ⇨ `Governance policies and rules`



__84. You will start by creating a Governance policy. Click the tab `Policies` ⇨ `Create Policy`.



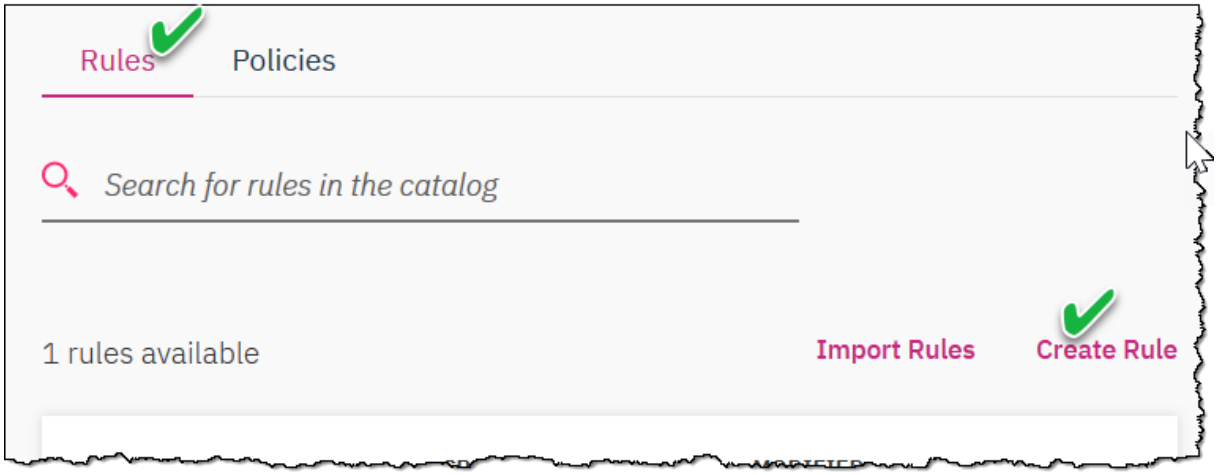
__85. Type *Name*: `Net Gains and Net Losses Are Mutually Exclusive` and *Description:* `A customer can only have a value in Net Gains or Net Losses, but not in both`.



__86. Click `Save`.

__87. We just created a policy. Now create a rule for it.

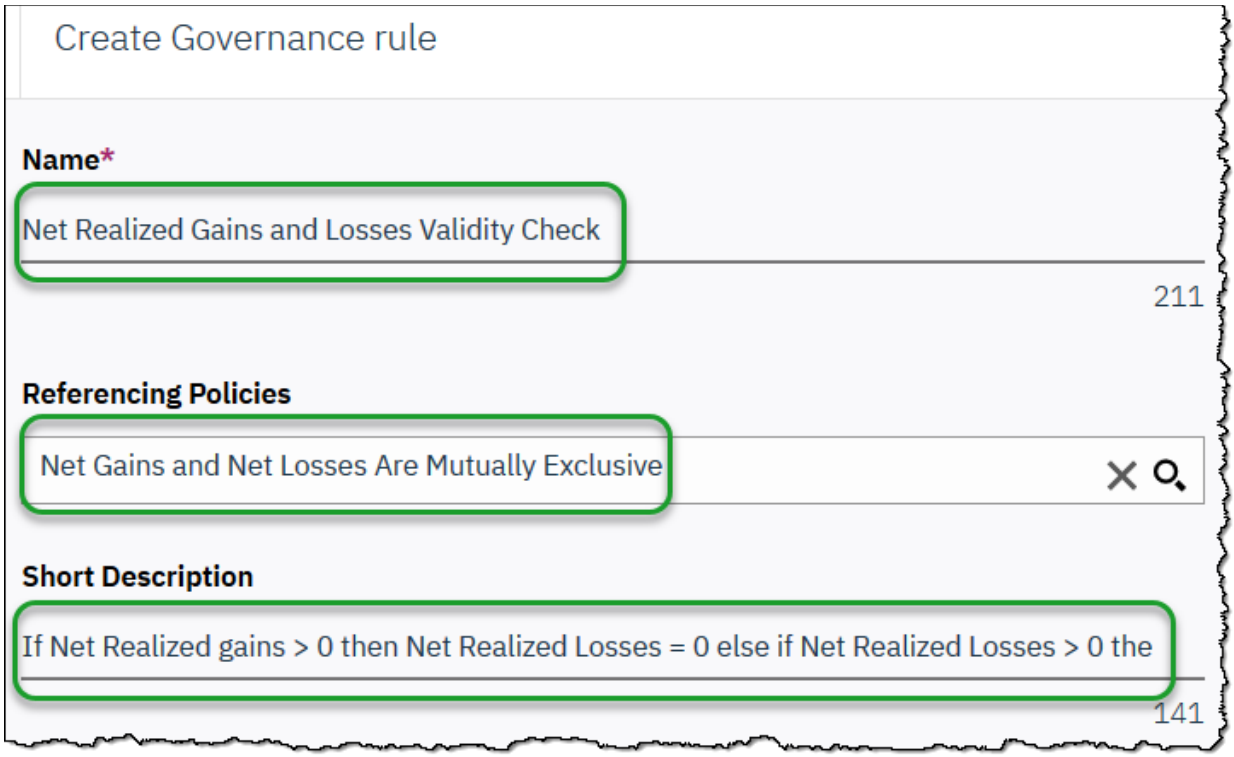
Click tab `Rules` ⇨ `Create Rule`.



__88. Type Name: `Net Realized Gains and Losses Validity Check`,

Type **'n'** to select the Referencing Policy

Type Description: `If Net Realized gains > 0 then Net Realized Losses = 0 else if Net Realized Losses > 0 then Net Realized Gains = 0.`



__89. Click `Save`.

 Data Steward	A `Data Dictionary` contains a business glossary (categories and terms) and information governance policies and rules to ensure data compliance with business objectives. In this lab, with the creation of your 1 category and 7 terms, 1 policy and 1 rule, you could now say the organization's Data Dictionary has a nice start. In reality, a Data Dictionary for any organization can and should be updated as frequently as new data sources, regulations and other criteria require it.

4.5 Discover assets

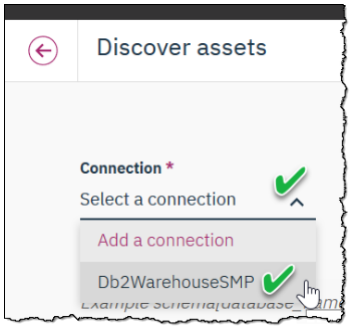
`Discover assets` allows you to do three things: 1) Profile and classify data, 2) Analyze the quality of data, 3) Assign business terms to data.

Let's see how this works.

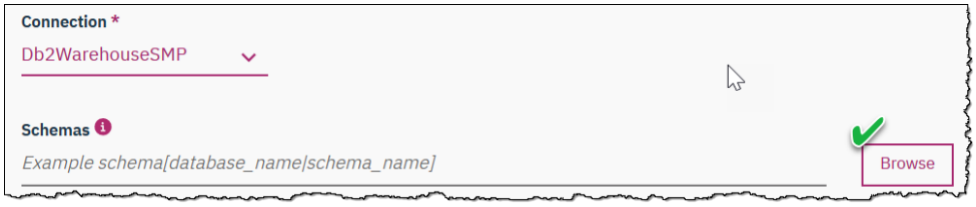
__90. Start at the `Navigation Menu`

__91. Click twistie `Organize` and expand it and click `Discover assets`.

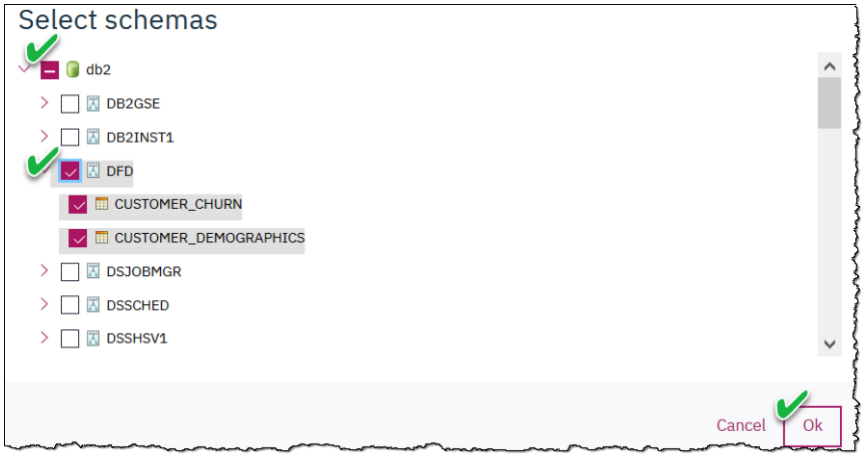
__92. Click on the `Connection` dropdown arrow, then select `Db2WarehouseSMP`



__93. In `Schemas`, click `Browse` (Note: Choose Db2Warehouse if the SMP is not there.)

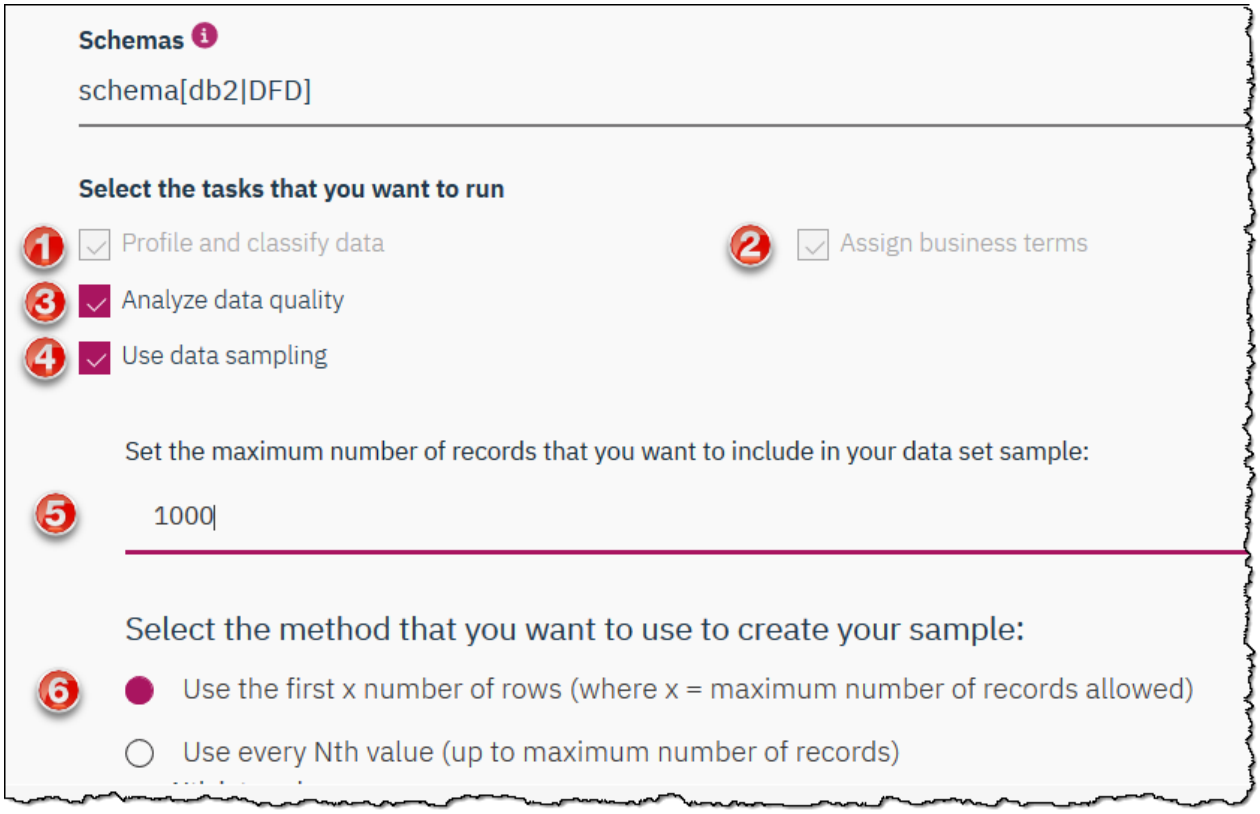


__94. Expand dropdown `db2` and select schema `DFD` ONLY, then `OK`



__95. The checked boxes below the Schemas field must be check in the following order:

1) Profile and classify data
2) Assign business terms
3) Analyze data quality
4) Use data sampling
5) Set the maximum number of records = 1,000
6) Select first button below maximum records field

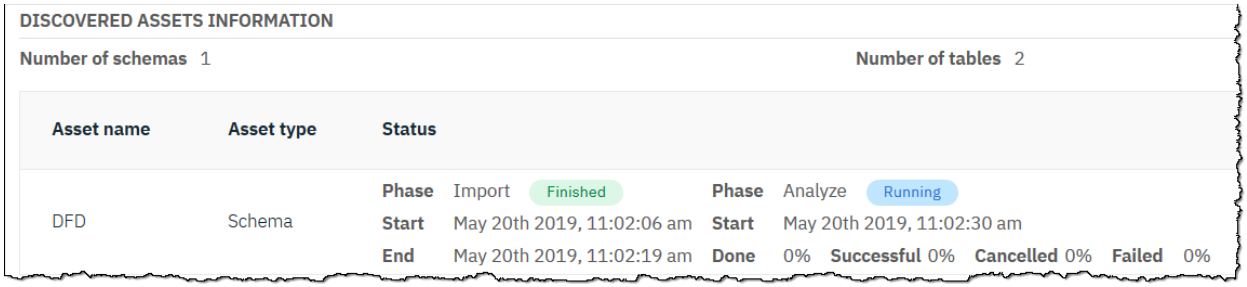


__96. Click Discover. (Bottom right of the screen.)

__97. A job is created and begins to execute. It may take few minutes to complete.

__98. You can click the Refresh icon to see if the job is still running.

__99. You can watch the progress of this job and refresh as needed. This should take 4 or 5 minutes.



Notice the Discovery phase has two stages – 1. `Import` and 2. `Analyze`.

The Discovery process can take some time, so be patient while it completes.

Hit the refresh button after a few minutes to see if it has completed.

Try again in another minute. (Note: If you get failure on analyze run analyze again.)

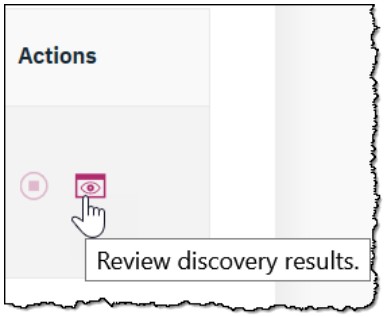
Data Steward	Occasionally the `Discover` process fails to finish after 10 minutes or so. If this happens to you, see Appendix B – Lab fixes – Lab 04 “Runaway Discover”. Note: a successful run of Discovery is NOT necessary to complete this lab, nor this workshop. If you wish to skip it, you may do so by going directly to the next section: Shop for Data.

__100. When it is finished, you will see that the ‘Analyze’ phase has a flag marked `Finished.`

Click the back arrow to return to the Discover screen.

Data Steward	This is an example of an incredibly powerful `automated discovery`, with an assignment of a quality score as well as `Terms` (if there was an appropriate match) so that it can be better used by the personas who may need it in their analytic processing. Simply by pointing to the data we want to process with CPD internal machine learning capabilities, CPD profiled and classified the data so that the business purpose is more easily understood. For example, it can discern PII data, the significance of some columns over others, and so on.

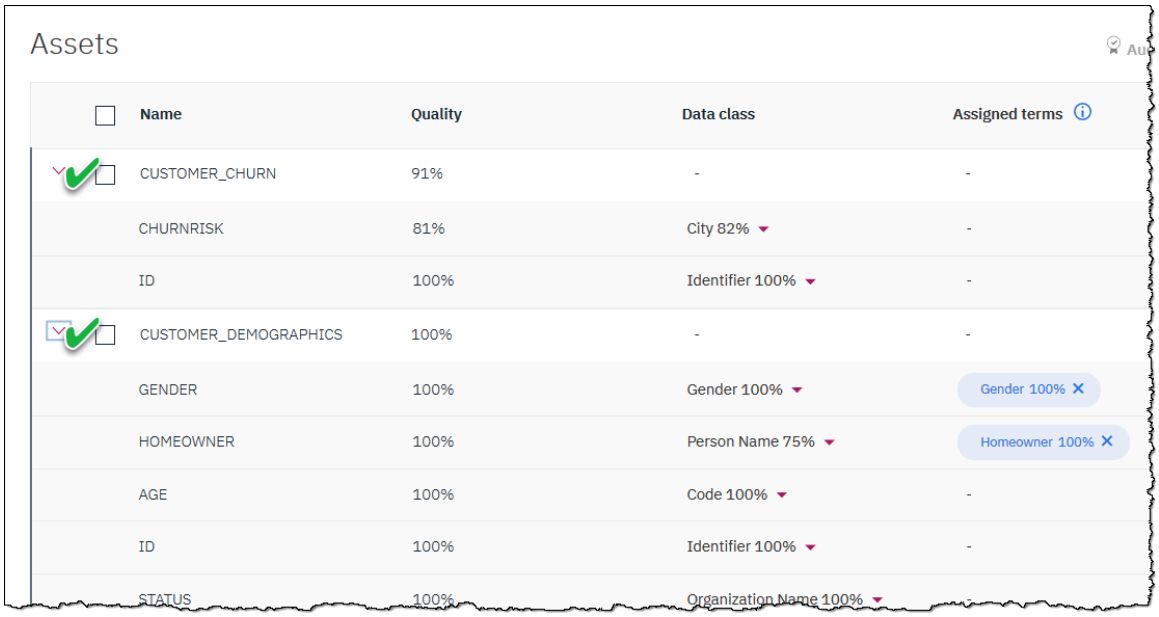
__101. Now you can review the results by clicking on the `review discovery results` (eye) icon:



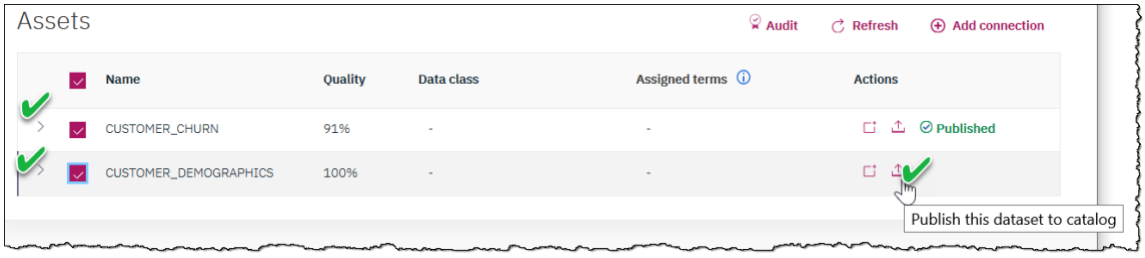
__102. Expand the two tables `CUSTOMER_CHURN` and `CUSTOMER_DEMOGRAPHICS`

Notice the data has been assigned a quality score at the table level as well as the individual table column level

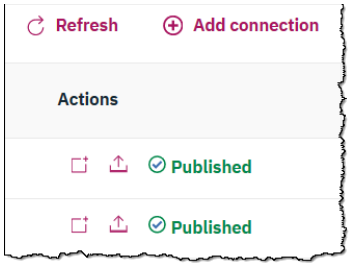
Notice too that column `Gender` was assigned the previously created term `Gender` with 100% certainty. Other columns were assigned terms with slightly less certainty.



__103. Publish these two tables to the catalog by clicking on publish icons as shown below:



__104. After submitting for publication, the screen will show this:

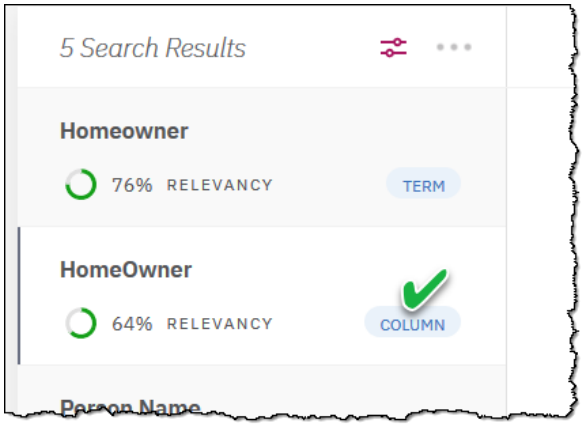


__105. Once published, we can search for the data. Try it by clicking on the `Search` icon in the top bar.

__106. Type `homeowner`

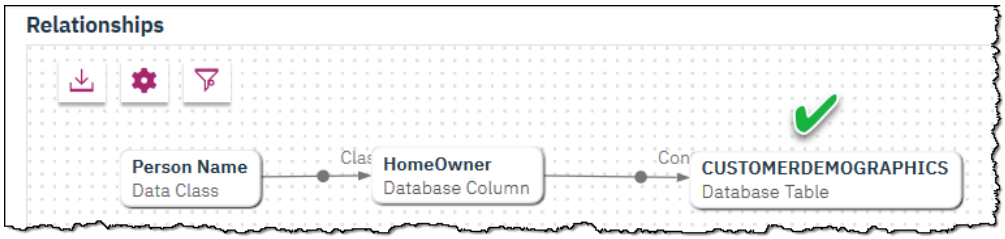
__107. Notice that a Term and a Column of a table come up.

Click on the `COLUMN` called `HomeOwner`



__108. In the COLUMN details screen, click on `Relationship Graph`

__109. After rearranging this screen, you will notice that this column belongs to the table CUSTOMERDEMOGRAPHICS. (Your illustration may vary)



 Data Steward	In our lab so far, you have only performed discover on a few tables so this relationship graph does not yet tell you of more complex relationships that it could know. You could do discovery for hundreds of databases to find relationships between them, as well as on your Data Dictionary items to uncover what Gartner calls “Dark Data” in your organization. This is data you are collecting, but not using because you do not know its value or how it relates to other data, or you cannot easily access it.

4.6 Shop for Data

The Data Scientist and Business Analyst personas may not always know what has been made available to them by the Data Engineers and Data Stewards in CPD. In fact, individual Data Engineers and Data Stewards may not always know what other users of the same persona have made available through their Collect and Organize activities.

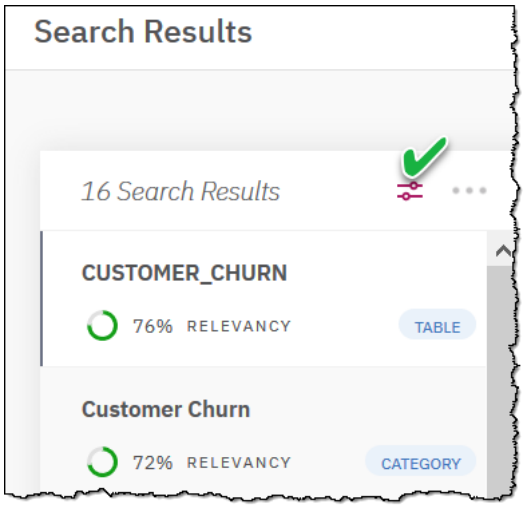
This is remedied by the ability to shop for data. You have already been introduced to the Search feature, which is the beginning step for shopping for data, so let’s explore it further.

__110. Click the Search icon.

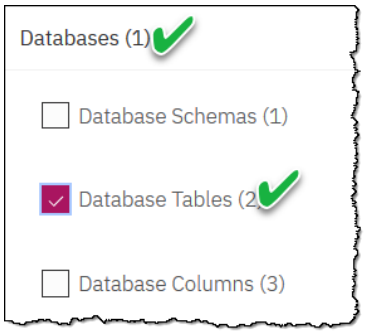
__111. Type churn and hit Enter.

__112. Assets related to churn are displayed.

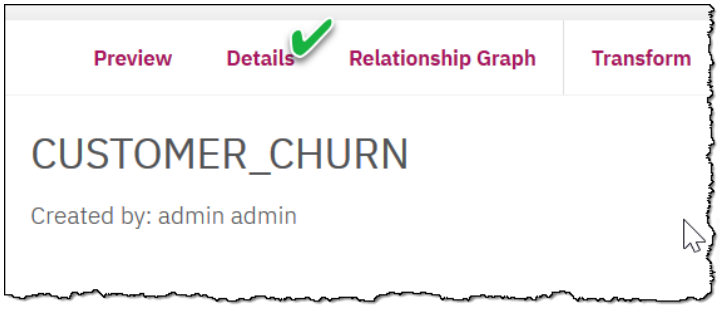
Click the filter icon to more easily explore the results of this search.



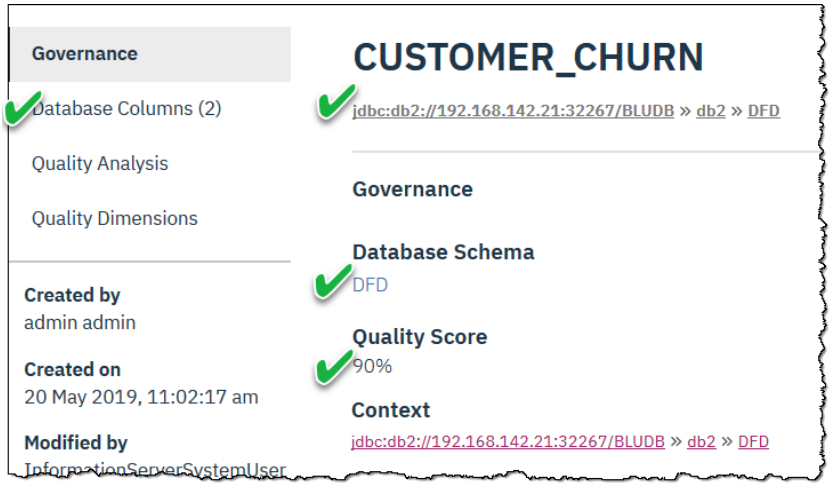
__113. Expand Databases then check Database Tables.



__114. Click on `Details` to explore this table `CUSTOMER_CHURN`



__115. Review the lineage of this table, the database, schema, quality score, and number of columns.

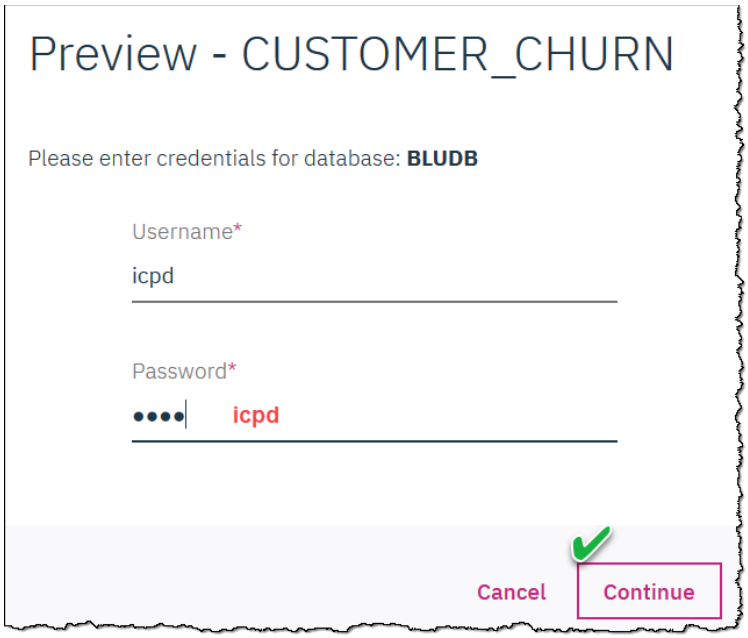


__116. Return to the previous screen for the table `CUSTOMER_CHURN`

__117. Click `Preview`

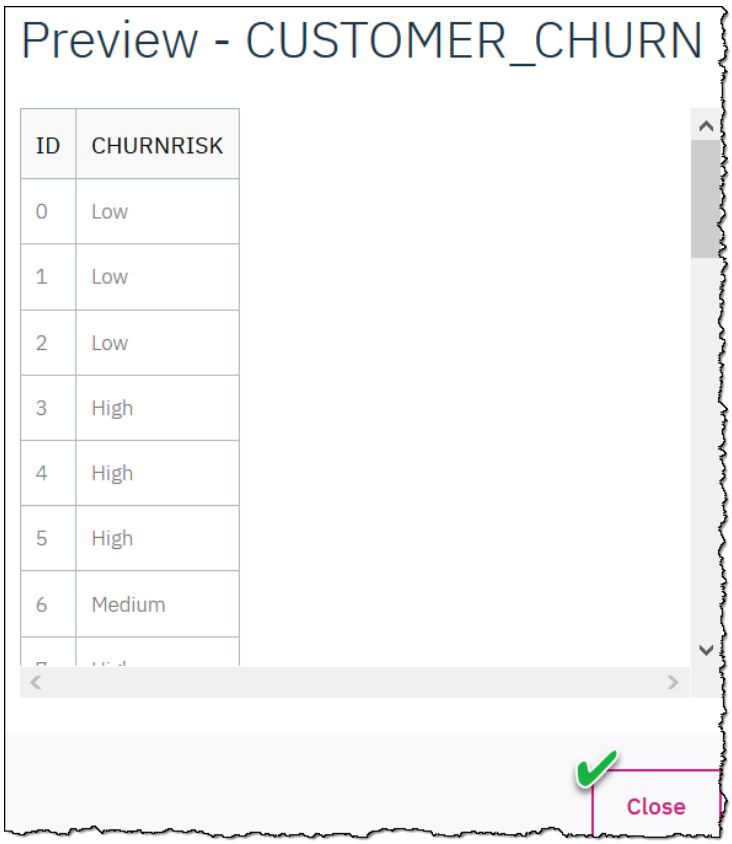
__118. For the credentials to this database, enter *Username* `icpd`, *Password* `icpd`

Then click `Continue`

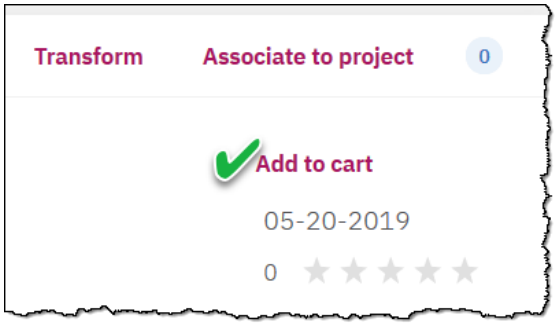


__119. Review the data itself to find out if it is suitable for your needs.

Then click `Close`



__120. The data looks good, so click `Add to Cart`

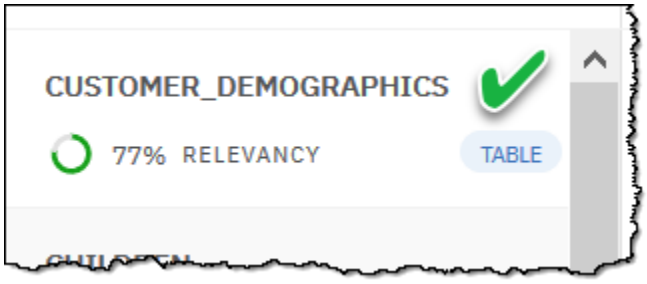


__121. You will need one more piece of data to complete your shopping experience.

The `Search` field should still be open – type `demographics` and hit `Enter`

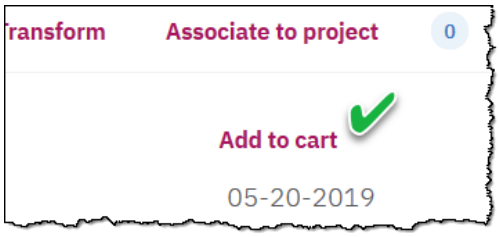
Note: if you closed the `Search` field, just click on the icon again to start another search.

__122. The results show another table called `CUSTOMER_DEMOGRAPHICS`. Click on it to choose it;



__123. To shorten this lab exercise, pretend you reviewed the details of this table and reviewed the data as well and that you have determined the data meets your needs.

Click `Add to cart`

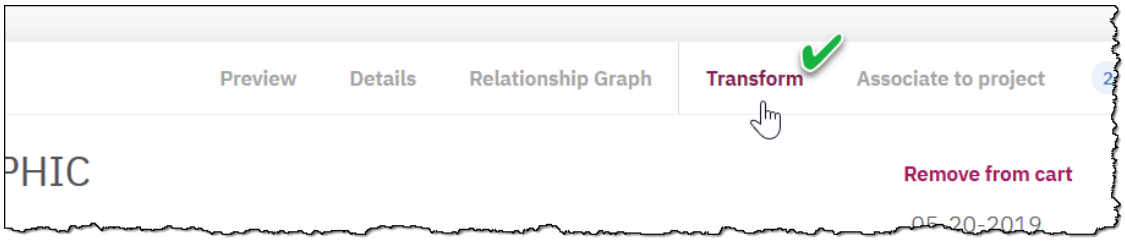


Data Steward	You have shopped for and chosen two tables for your cart, from which you will build a job to join and transform this data.

4.7 Transform Data

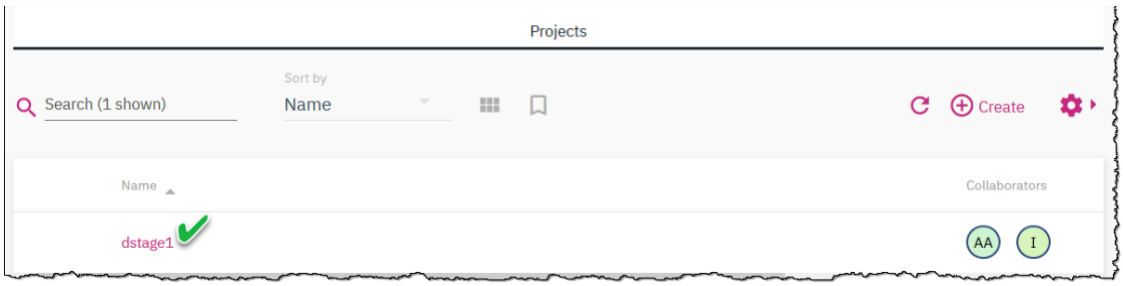
In this set of exercises, you will use the CPD built-in Data Flow Designer (DFD) to build a job that can join and transform your data.

__124. Click `Transform`.



__125. This is the DFD projects screen where an example is already listed.

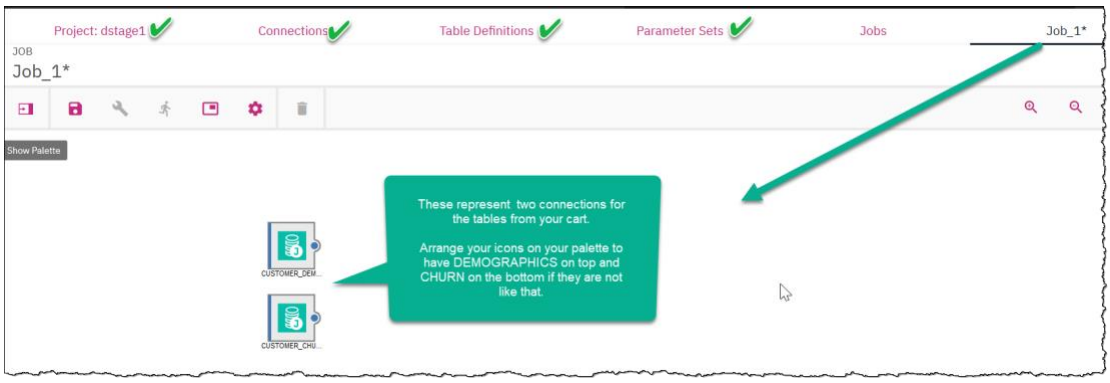
Click project `dstage1`.



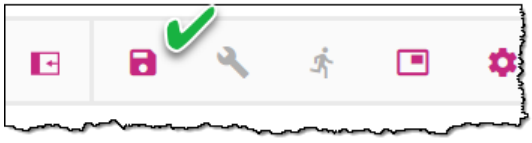
__126. The project `dstage1` canvas has across the top various options: `Connections`, `Table Definitions`, `Parameter Sets` and `Jobs`.

Notice DFD has created a new job for you called `Job_1*` and placed on the canvas the two connections for the data you added to your cart. These are for the CUSTOMER_DEMOGRAPHICS and CUSTOMER_CHURN tables.

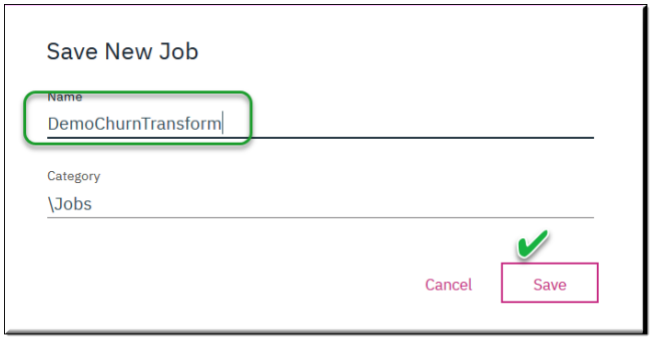
Arrange your icons to have DEMOGRAPHICS on top if they are not presented to you on your canvas that way. Note: If they do not link delete and re-add to cart.



__127. Save this job.



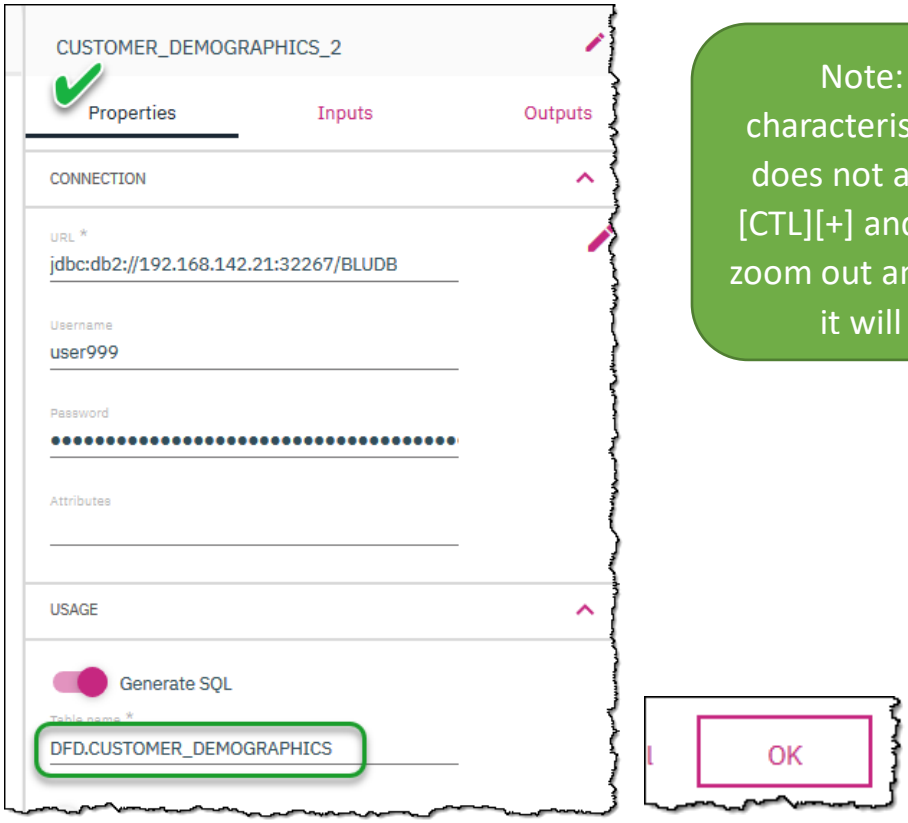
__128. Rename it to `DemoChurnTransform`



__129. Double click on the first connection on the canvas for `CUSTOMER_DEMOGRAPHICS`

__130. The pop up connection editor allows you to review and edit the characteristics of this connection. Add the schema to the Table name, so it looks like this: `DFD.CUSTOMER_DEMOGRAPHICS`

Click `OK` to save and close this screen.



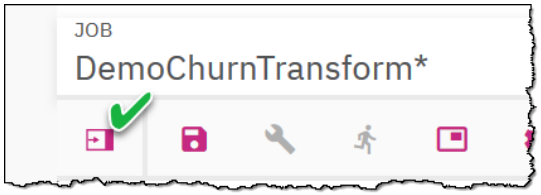
Note: If this characteristics screen does not appear, use [CTL][+] and [CTL][-] to zoom out and in so that it will show.

__131. Double click on the second connection icon on the canvas, which is `CUSTOMER_CHURN`

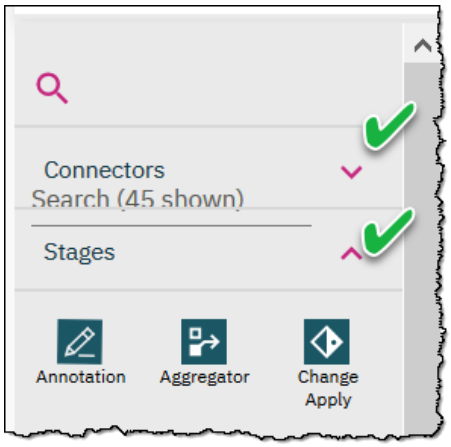
Add the schema to the Table name, so it looks like this: `DFD.CUSTOMER_CHURN`

Click `OK` to save and close this screen..

__132. Review the palette options by clicking on the show palette icon:

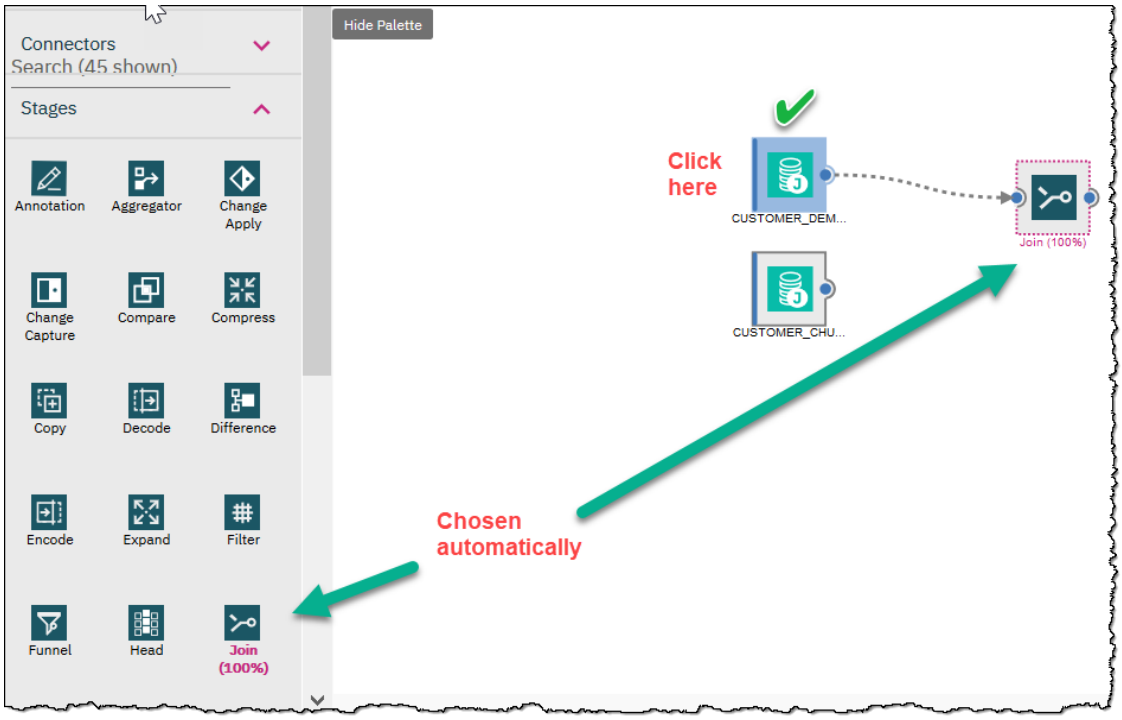


__133. Collapse the `Connectors` palette section and expand the `Stages` palette section:

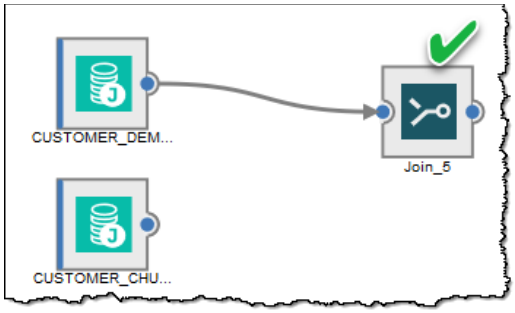


__134. Click once on the CUSTOMER_DEMOGRAPHICS connection icon on the canvas to choose it.

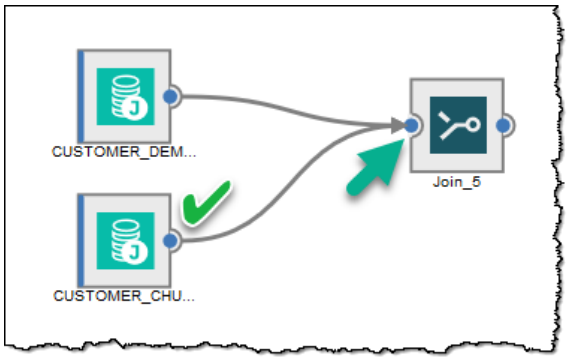
With the Stages palette options open, the Data Flow Designer does a “Smart Stage suggestion” picking the Join Stage from the palette as a possible next step.



__135. Click on the Join icon on the canvas to take the suggestion and make that choice. The icon will now have solid lines around it.



__136. Click on the nub of the CUSTOMER_CHURN connection icon and drag the connecting arrow to the Join icon. This provides the Join stage the two tables as sources for the join.

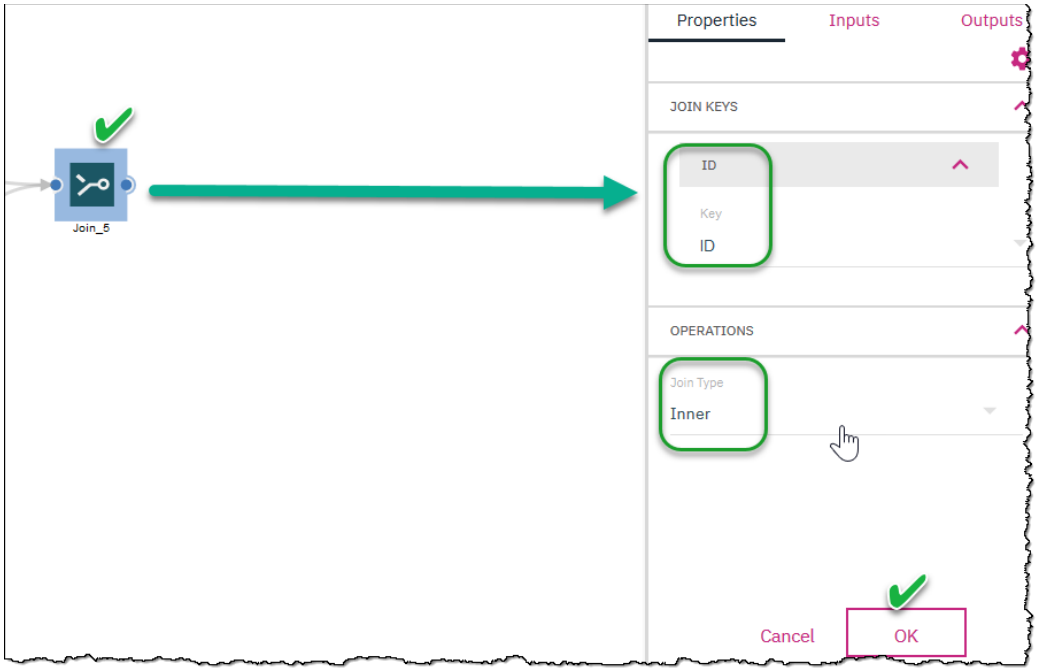


__137. Double-click on the Join stage icon to review its properties, which will pop up on the right.

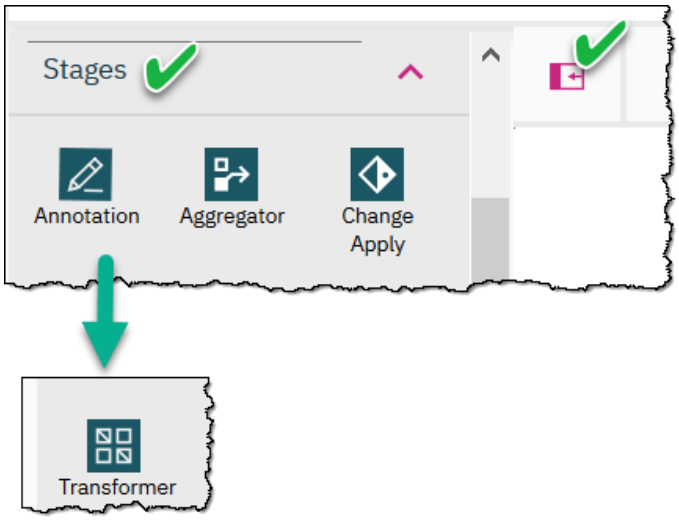
Data Flow Designer inferred the join to be on column `ID` for you already because both tables had the same column name and data type. Leave that as is.

DFD also chose an `Inner` Join Type, which means only rows that are common to both tables with a matching ID will result. Leave that as is.

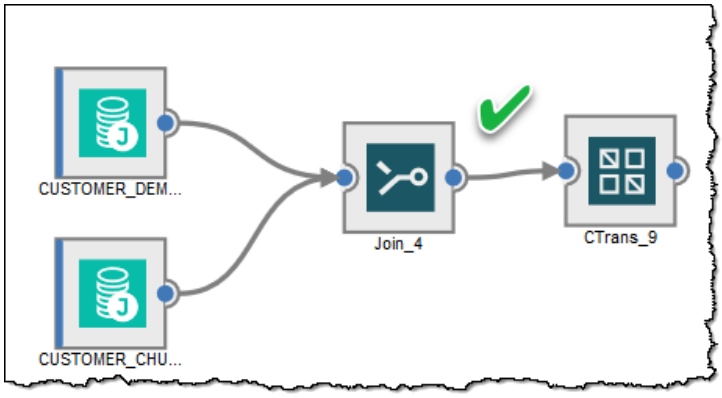
Click `OK` to keep these join options.



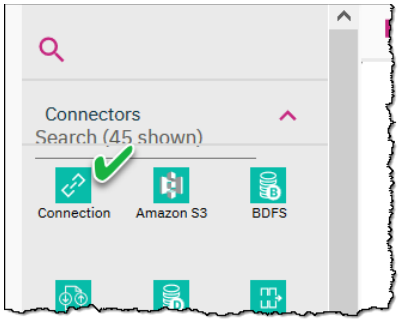
__138. Open the `palette` icon again and choose stage: `Transformer`



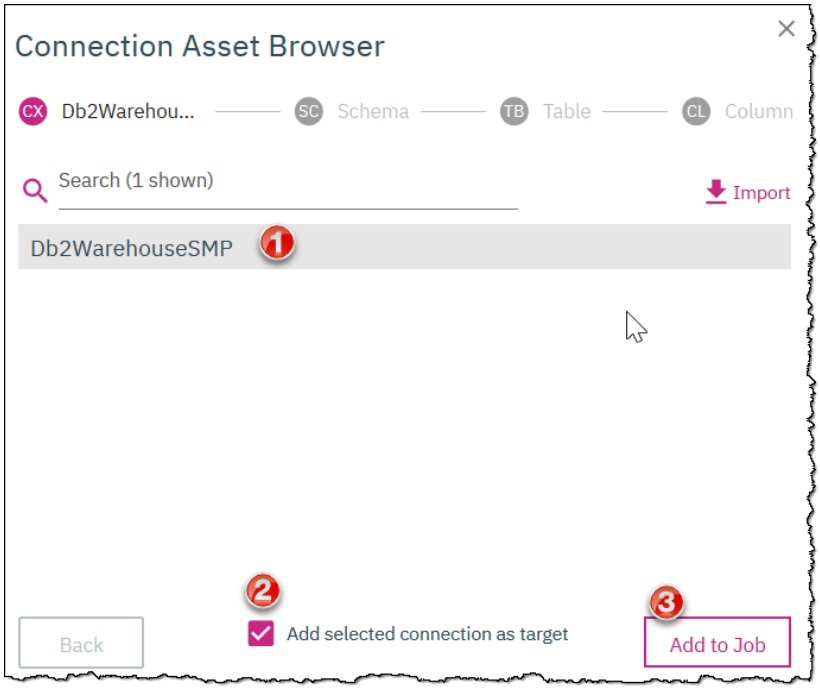
__139. Drag the `Transformer` icon to the right of the `Join` stage icon. DFD should automatically link it to the Join stage icon. If this doesn't happen, make the link yourself.



__140. From the `Connectors` palette, choose the `Connection` icon and drag it onto the canvas to the right of the `Transformer` icon

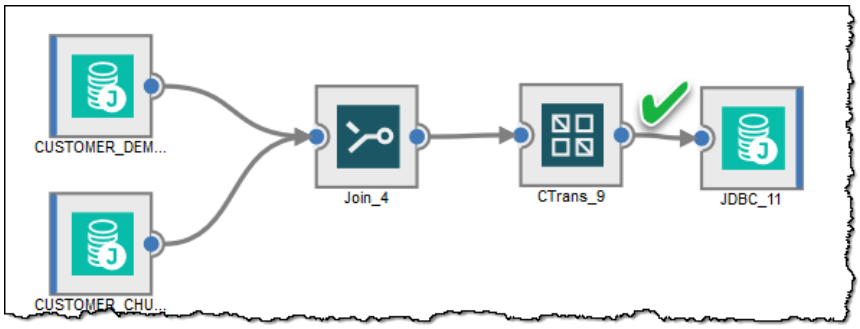


__141. The `Connection Asset Browser` will pop-up – choose connection `Db2WarehouseSMP` ⇨ `Add selected connection as target` ⇨ `Add to Job`



__142. Arrange your canvas icons as shown below. If DFD did not do so for you, connect the `Transformer` icon to the target `JDBC connection` icon.

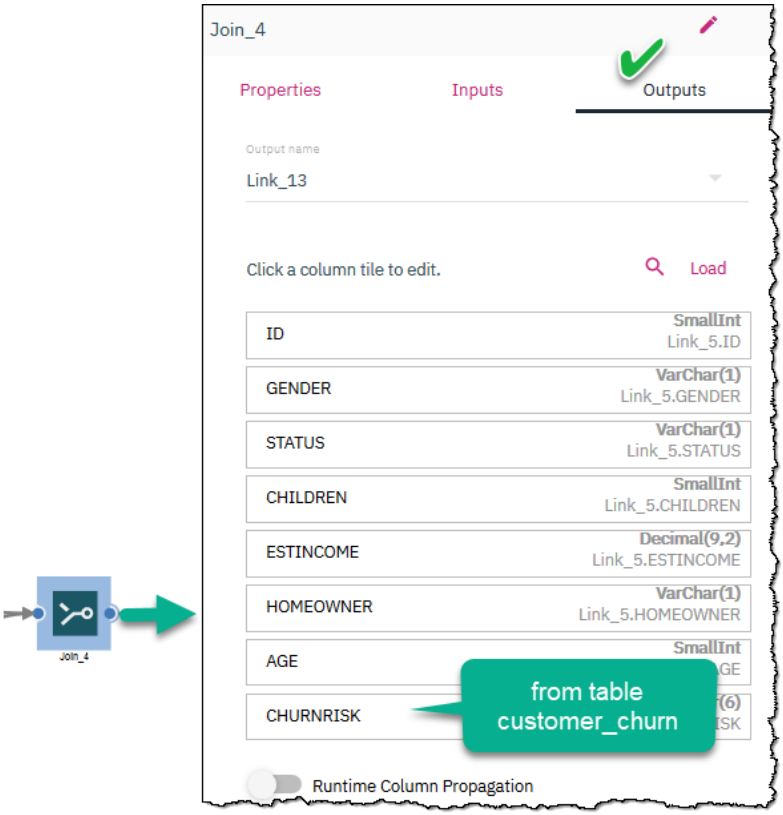
The icons on the palette should look like this:



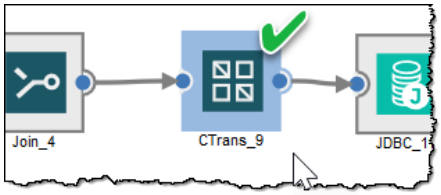
Data Steward	The default names given these stages (like Join_4, CTrans_9) may not exactly match yours. This really doesn't affect anything. You could in fact rename these stages to be more meaningful to you. Leave them as-is for now.

__143. Double-click on the `Join` stage and then click on the `Outputs` section which represents the columns coming out of the Join.

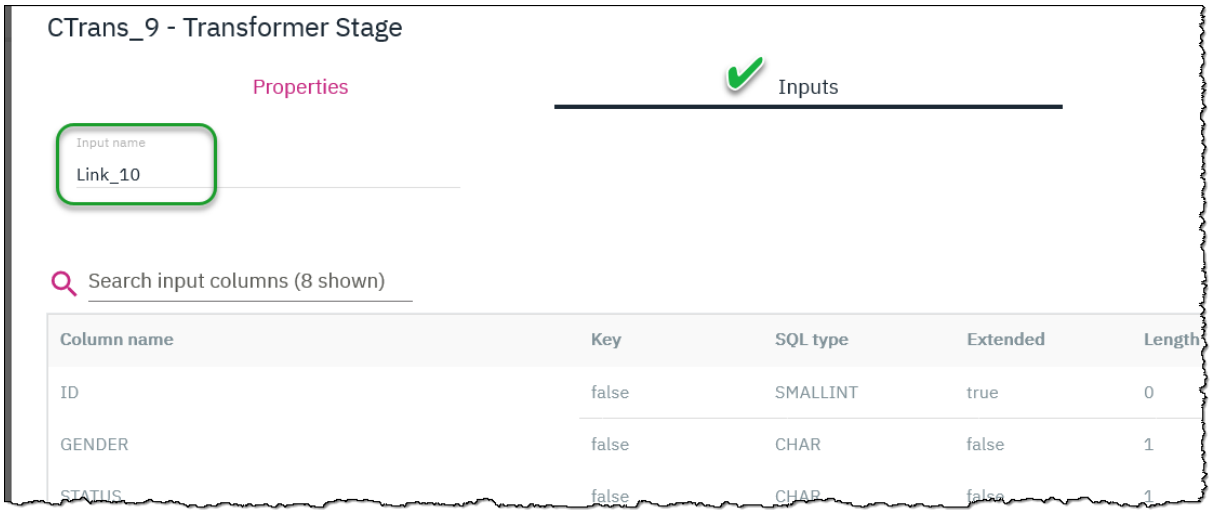
Notice that column CHURNRISK was from the CUSTOMER_CHURN table, whereas all the other columns are from the CUSTOMER_DEMOGRAPHICS table. ID is the common column the two tables are joined by.



__144. Double click on the `Transformer` icon to edit it.

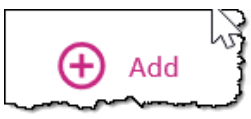


__145. Select the `Inputs` tab to review the columns coming into the `Transformer` stage. Notice the input name is generated as “Link_10”. (Your value may vary – if so, make note of it).



__146. Now select the `Outputs` tab, where we will build our transformation of this data. Notice some automatic derivations have been made for you to start with, which are simply the input column name with the Transformer `input name` in front of it. For example, `ID` becomes `Link_10.ID`.

__147. Click `+Add` to create a new derived column.

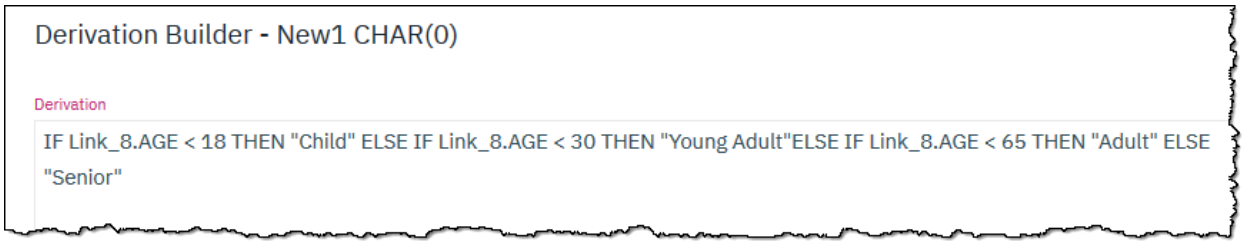


__148. Scroll down to find the new derived column and click in it to edit it.

__149. Fill in the Derivation field with the following text.

Note: if your Input name Link_n number is different than this example, use your value and not the one from this example for all three of the highlighted references shown in red.

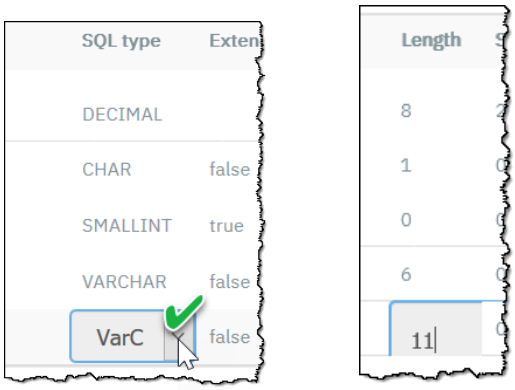
```
IF Link_8.AGE < 18 THEN "Child" ELSE IF Link_8.AGE < 30 THEN "Young Adult"
ELSE IF Link_8.AGE < 65 THEN "Adult" ELSE "Senior"
```



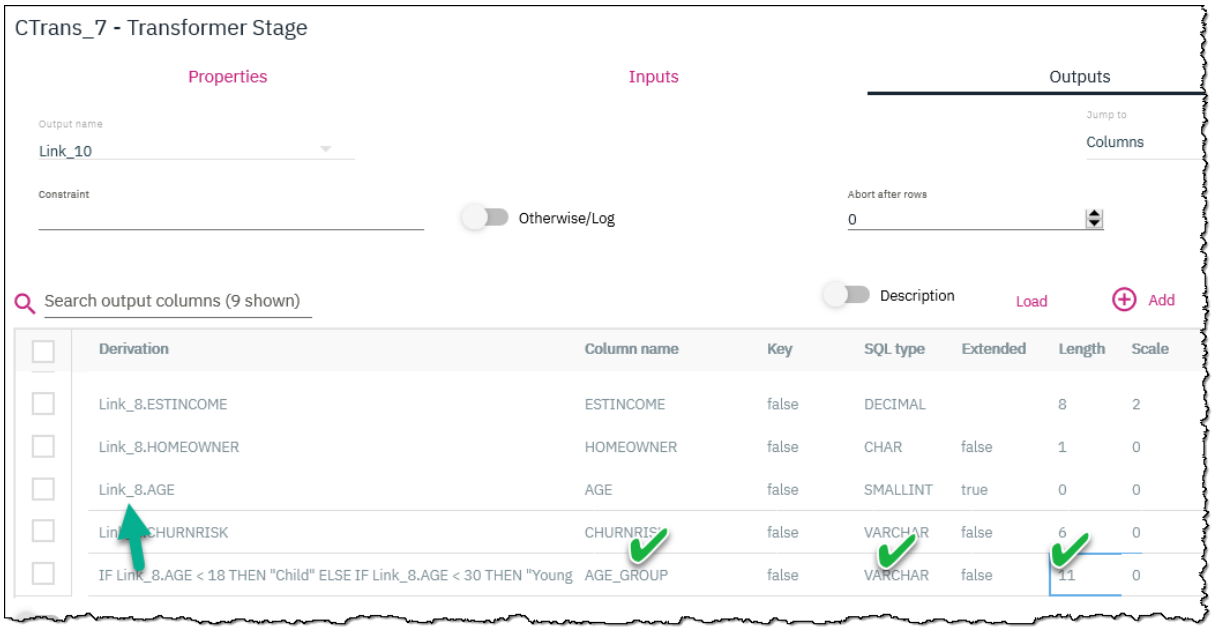
__150. Double-click OK to save the derivation.

__151. For this derivation output field, click on the temporary column name (New1) and change it to AGE_GROUP.

__152. Change the SQL type to VarChar and Length to 11.



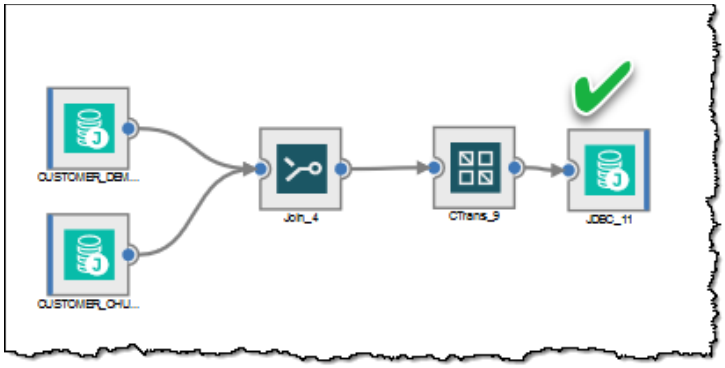
__153. The final results should look like this (Note: the link_n.AGE references in the derivation matches the value for Link_n.AGE shown in the output columns)



__154. Click `OK` to save your work in the `Transformer` stage.

__155. Save this job again to preserve the work you have done so far.

__156. Now click on the output (target) connection stage icon.



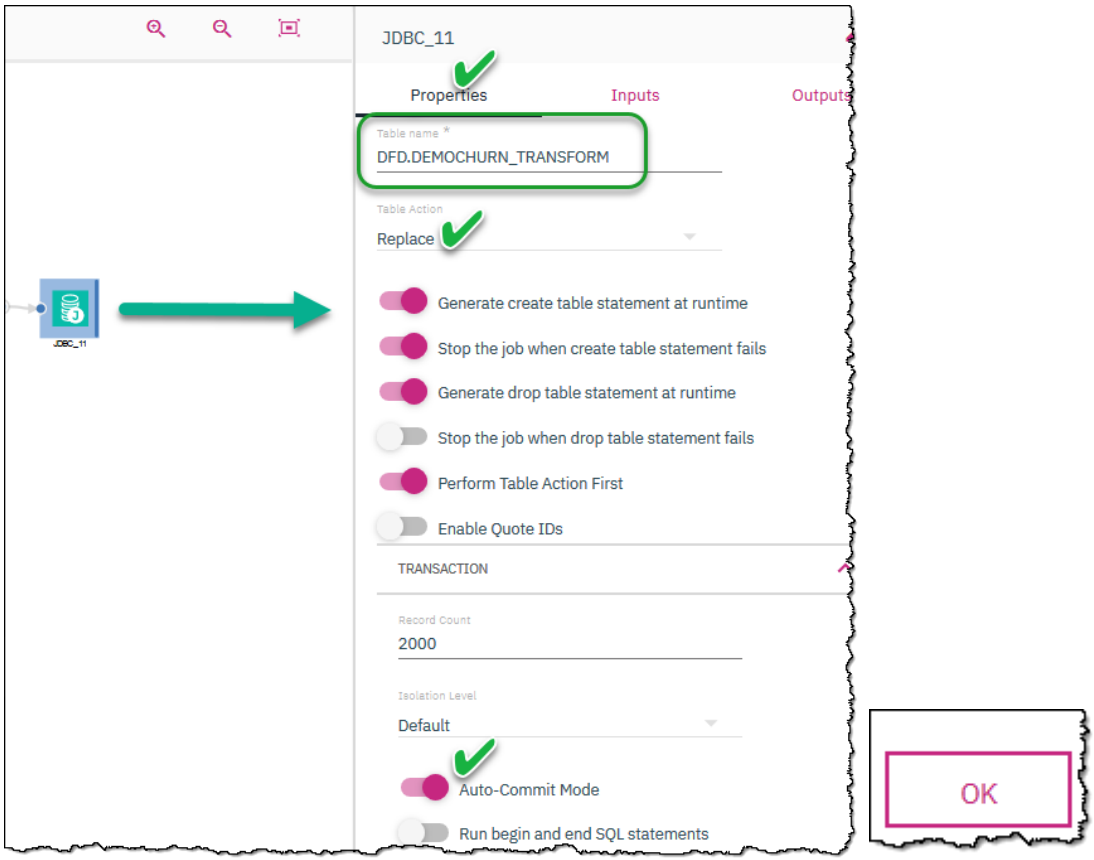
__157. Click on `Properties` tab and scroll down to find Table name.

Fill Table name in as `DFD.DEMOCHURN_TRANSFORM`.

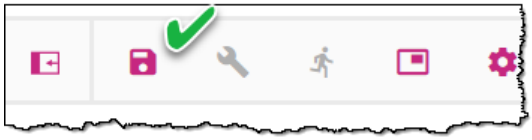
In Table Action, choose `Replace` (Leave the defaults for this option as-is)

Slide on the button: `Auto-Commit Mode`

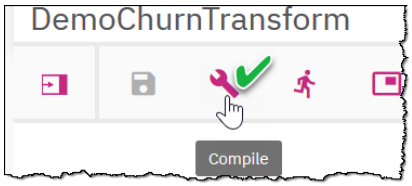
Click `OK` to save.



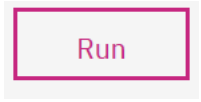
__158. `Save` this job again.



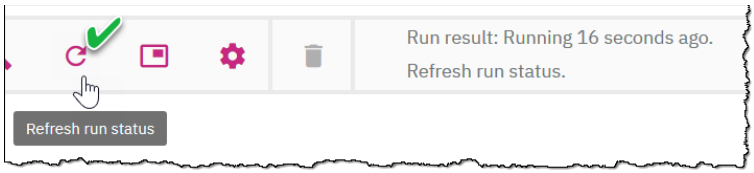
__159. This enables the `Compile` option (the wrench icon) Click on it to Compile your job.



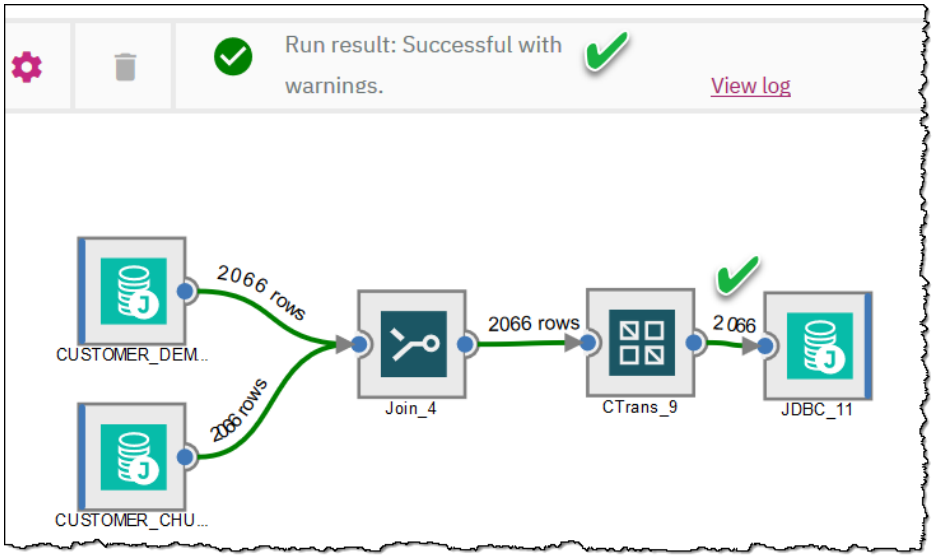
__160. After it compiles successfully, click on the Run icon (the running man), then Run



__161. Let the job run for 1 minute, then click on the Refresh run status icon

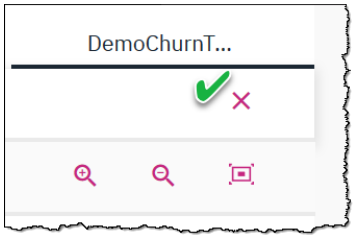


__162. You should now see the job successfully finish with a record of the rows that flowed through the job and written to the output table in the target connection stage.



Data Steward	If the job did not complete successfully, check for these common mistakes: • Make sure the input and output tables in the connections use the schema DFD (entered before the table name) • Make sure the connections between the stages are made as shown in the above lab steps • Make sure your Transformer stage derived column formula is correct.
Data Steward	Reviewing the View Log should produce only one warning and that is the Db2 drop table statement for the output table created from the run. This is normal because the output table did not exist on the first run. On subsequent runs this warning would not be there.

__163. Close the job window to end your edit session.

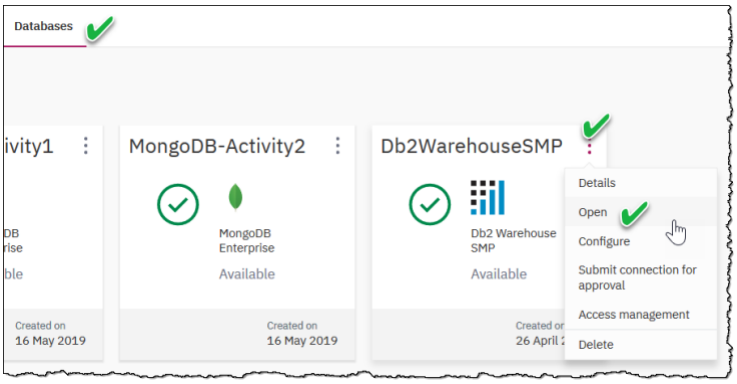


__164. You will see the job `DemoChurnTransform` you just created in project `dstage1`.

__165. Let's make a check on our data by reviewing the Db2 Warehouse again:

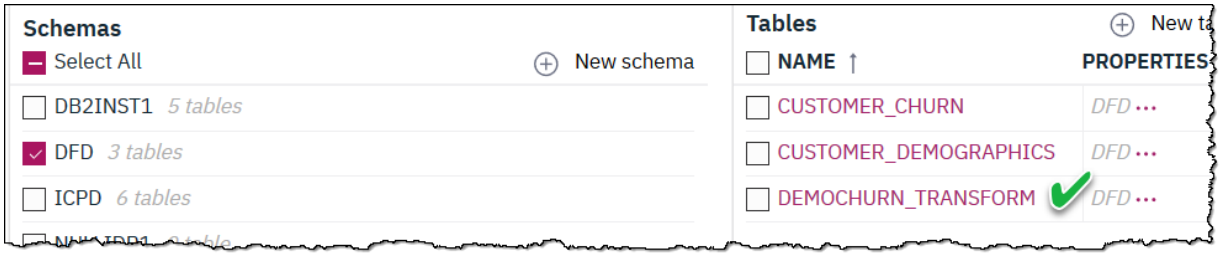
`Navigation Menu` ⇨ `Collect` ⇨ `My data` ⇨ `Databases`

`Db2WarehouseSMP` ⇨ `ellipsis` ⇨ `Open`



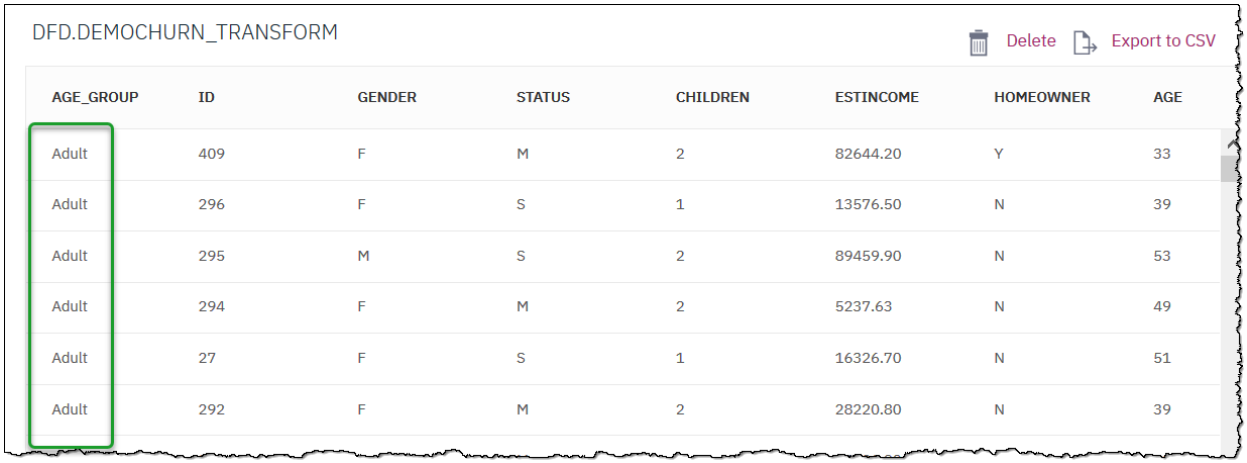
__166. Then `Menu` ⇨ `Tables`

__167. Schema DFD ⇨ DEMOCHURN_TRANSFORM



__168. Then View Data

__169. Browsing through this transformed output table shows our derived column AGE_GROUP



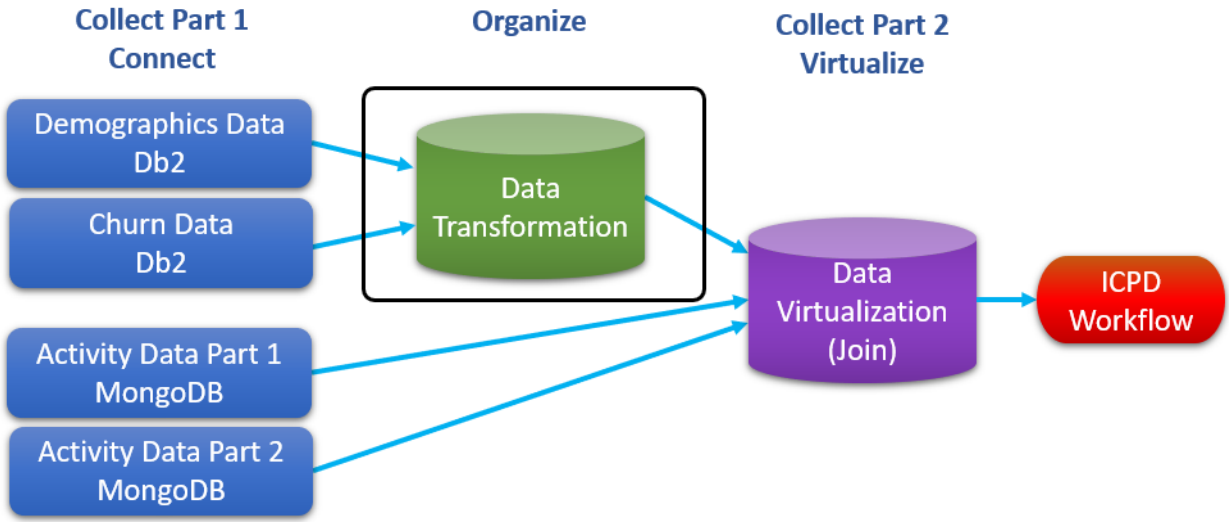
AGE_GROUP	ID	GENDER	STATUS	CHILDREN	ESTINCOME	HOMEOWNER	AGE
Adult	409	F	M	2	82644.20	Y	33
Adult	296	F	S	1	13576.50	N	39
Adult	295	M	S	2	89459.90	N	53
Adult	294	F	M	2	5237.63	N	49
Adult	27	F	S	1	16326.70	N	51
Adult	292	F	M	2	28220.80	N	39

4.8 Lab conclusion

We have seen the value in creating a Data Dictionary by creating a glossary of categories and terms to make data searchable so that data scientist, data engineers and business analysts can shop for data. The CPD platform reduces the time-consuming data organization task, making it easier for the consumers of the data.

This lab has shown you how to automatically discover data sources, automatically classify those sources with business classifications using CPD machine learning methods and automatically assign business terms to those data sources.

It showed you how to shop for specific data, and join and move that data to an analytics warehouse in a Data Transformation step for further use by the data consumers in the CPD workflow.



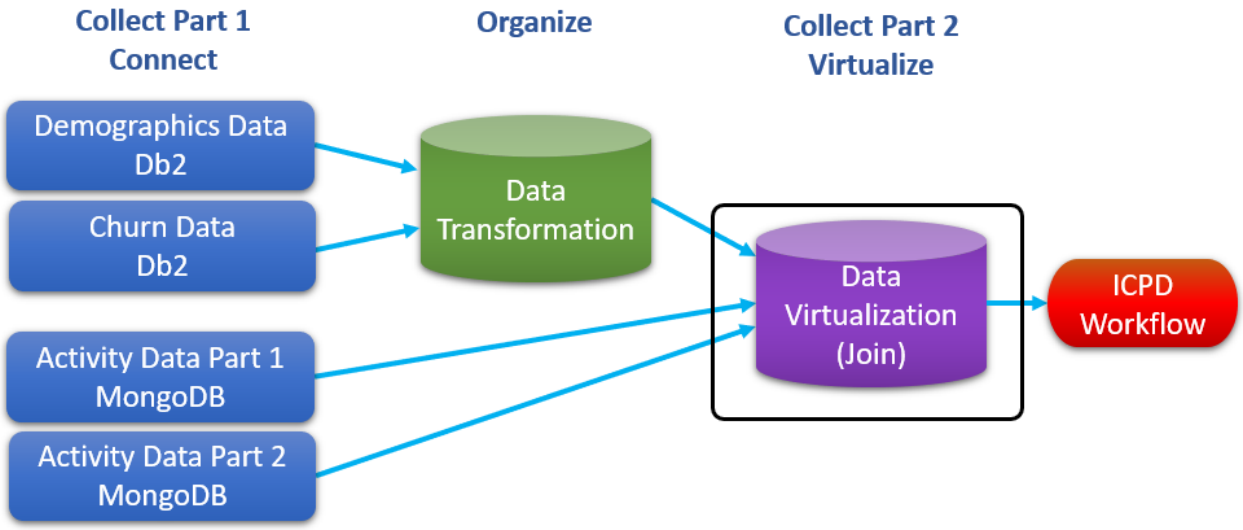
The steps covered here could normally take months, and sometimes years, to complete using traditional manual methods. CPD automates these things so that you can accelerate the time to value of your analytics projects.

**** End of Lab 04: Organize**

Lab 05 Collect Part 2 - Virtualize

5.1 Lab overview

In this lab, you will learn about `Data Virtualization` to complete the `Collect` tasks by creating a virtualized view of the transformed Db2 Demographics and Churn data, joined with the MongoDB Activity data.



5.2 Persona represented in this lab

The `Data Engineer` persona will be taking over again to perform the various `Collect` tasks shown in this lab.

Persona (Role)	Capabilities
 Data Engineer	Data Engineers build and optimize the systems to allow data scientists and business analysts to perform their work. The Data Engineer ensures that any data is properly received, transformed, stored, and made accessible to other users.

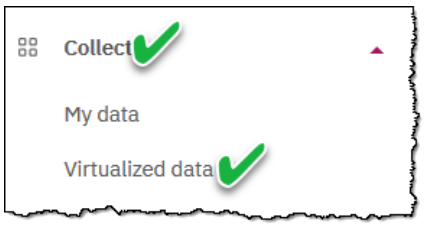
Note: Data Virtualization capabilities could be granted to the Data Steward as well, but in this workshop scenario we are assigning this task to the Data Engineer.

5.3 Data Virtualization data sources

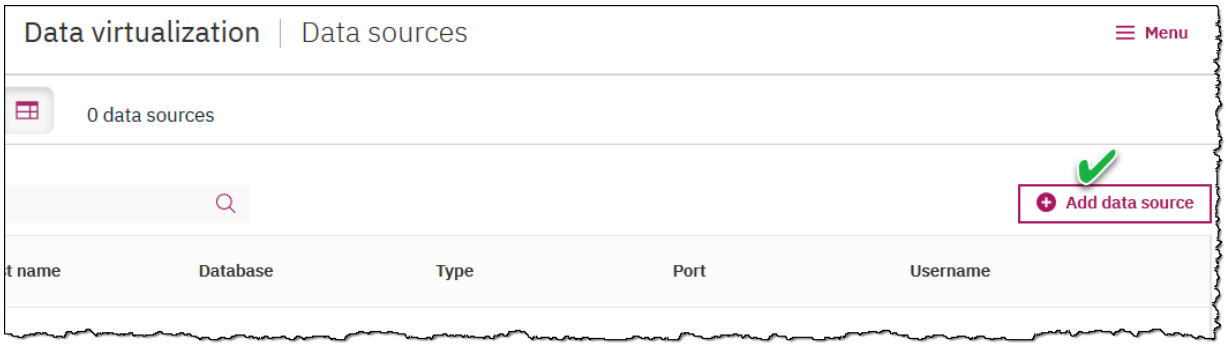
Let's explore the Data Virtualization process by adding data sources to virtualize.

__170. Start at the `Navigation Menu`

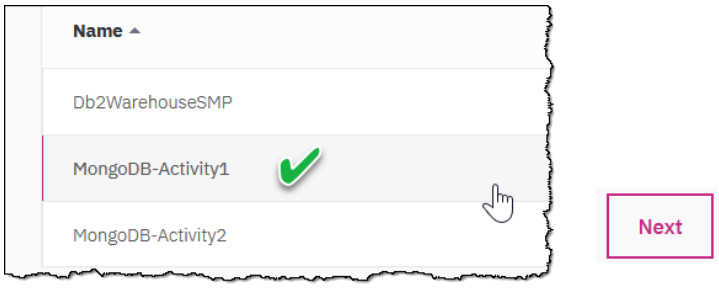
Click `Collect` ⇨ `Virtualized data`



__171. From the Data sources screen, click `+ Add data source`

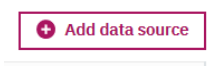


__172. Select (click on) `MongoDB-Activity1` ⇨ `Next`



Data Engineer	If any of the Add data source activities fails in this exercise, go back to the `Navigation menu` ⇨ `Home`. Then enter the Virtualized data and try again.

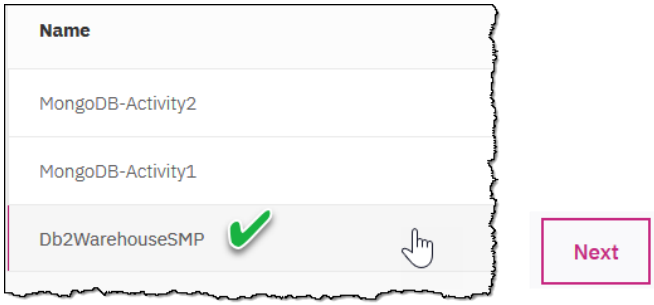
__173. For a second time, click `+ Add data source`



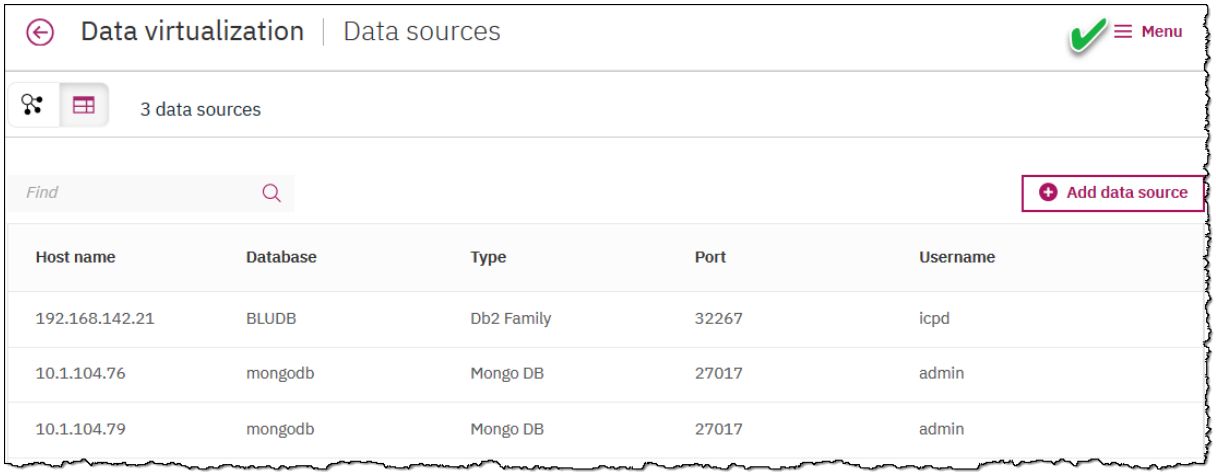
__174. Select (click) `MongoDB-Activity2` ⇨ `Next`

__175. For a third time, click `+ Add data source`

__176. Select (click) `Db2WarehouseSMP` ⇨ `Next`



__177. The `Data Virtualization | Data sources` screen should have three connections, as shown here. They are two Mongo and one Db2 source. (Hint: make sure the Mongo Host names are different)

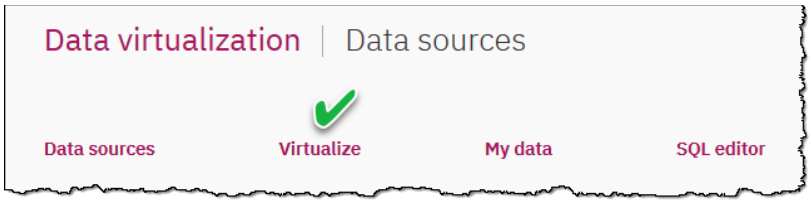


Data Engineer	Even though you used previously created connections to create these sources, the ones shown here are now specific to Data Virtualization processing.

5.4 Virtualize the MongoDB data with the Db2 data

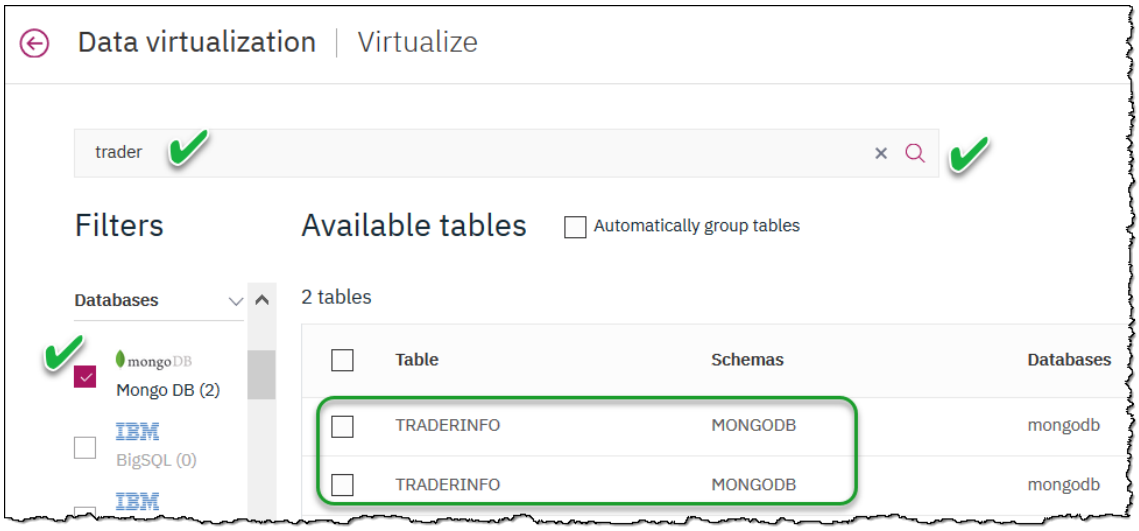
__178. Click `Menu` to bring you to the Data Virtualization options.

__179. Click `Virtualize`



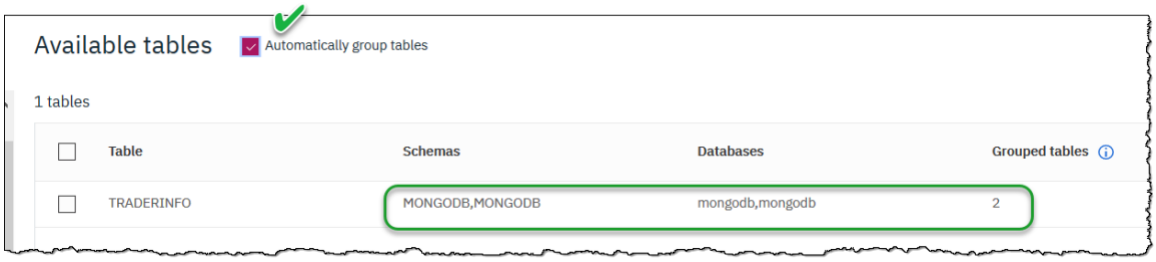
__180. In the search box, type in `trader`

Check the box for `Mongo DB (2)` and click the search icon



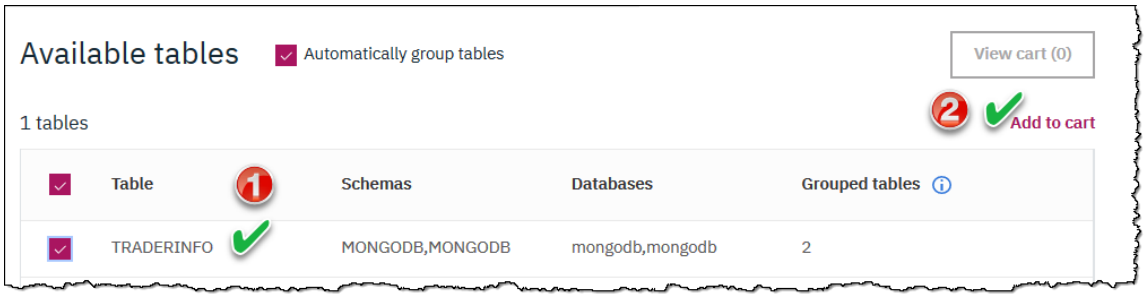
__181. Check the box `Automatically group tables`

Notice that tables with the exact same name (regardless of the database and schema they reside in) will be grouped (or folded) together. This essentially makes them one table.



 Data Engineer	You could have grouped many more databases than the two in this workshop.

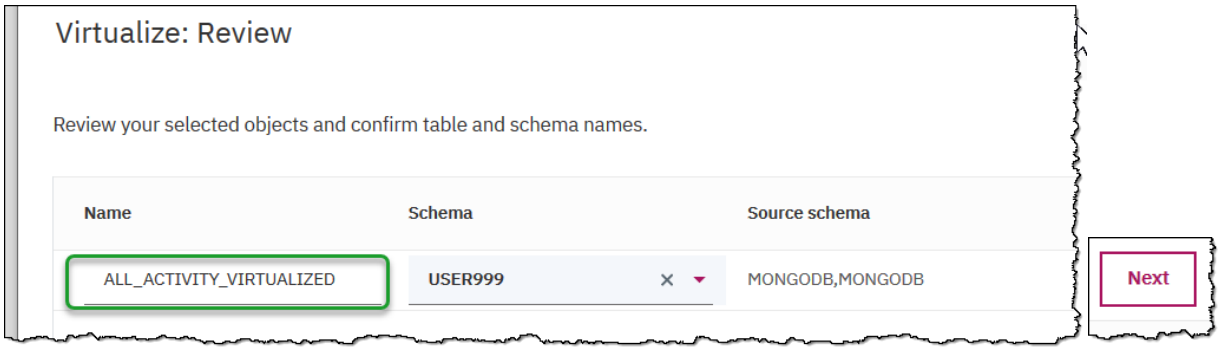
__182. Now check the table `TRADERINFO` and click `Add to cart`



__183. Click `View cart` then `Next`

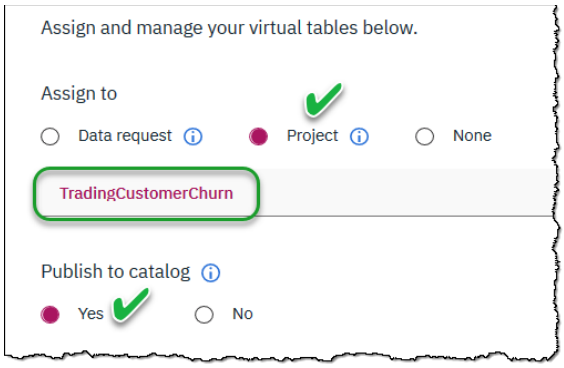
__184. Under `Name`, fill in the name for this virtualized table: `ALL_ACTIVITY_VIRTUALIZED`

Click `Next`



__185. Click button `Assign to` ⇨ `Project`, then choose project: `TradingCustomerChurn`

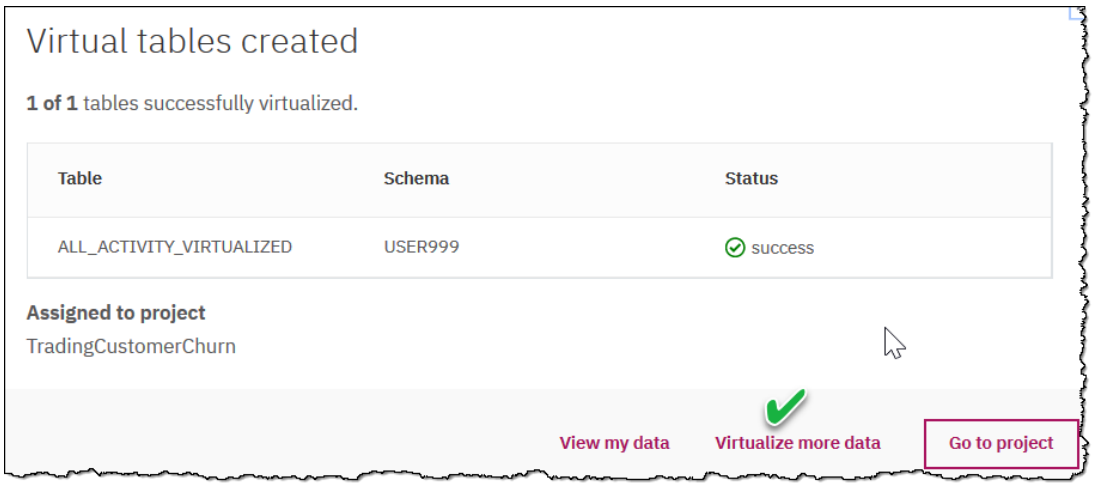
Click button `Publish to catalog` ⇨ `Yes`



Click `Virtualize`

__186. The virtualized table of the two MongoDB sources was created.

Click `Virtualize more data`

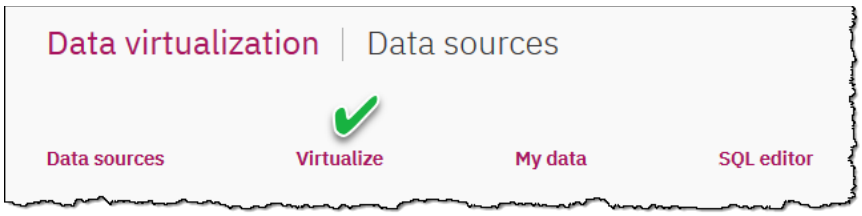


5.5 Virtualize the Db2 data

Now you will virtualize the Db2 table DEMOCHURN_TRANSFORM that you created in the Organize lab to prepare it for a join with the ALL_ACTIVITIES_VIRTUALIZED table.

__187. At the top right of the screen click on the hamburger icon Menu.

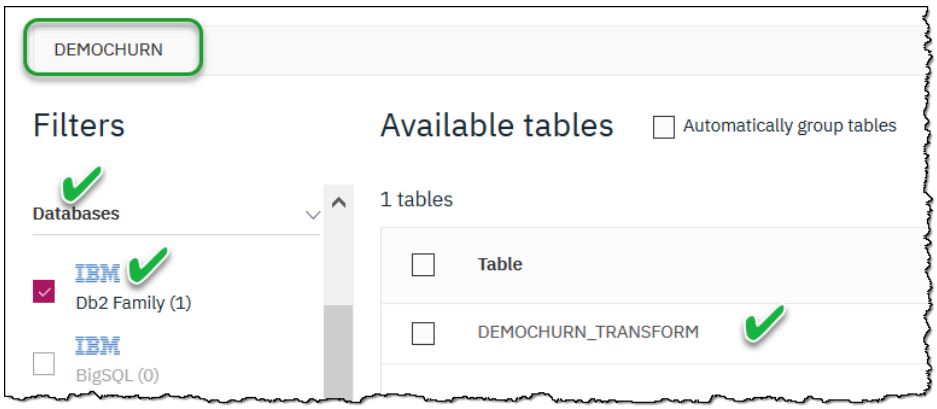
__188. Click Virtualize



__189. Under Databases, check box Db2 Family (Make sure Mongo DB is not checked.)

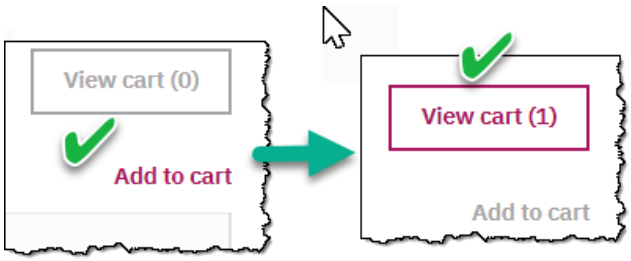
Fill in Search bar with DEMOCHURN then hit Enter

Results should show table DEMOCHURN_TRANSFORM



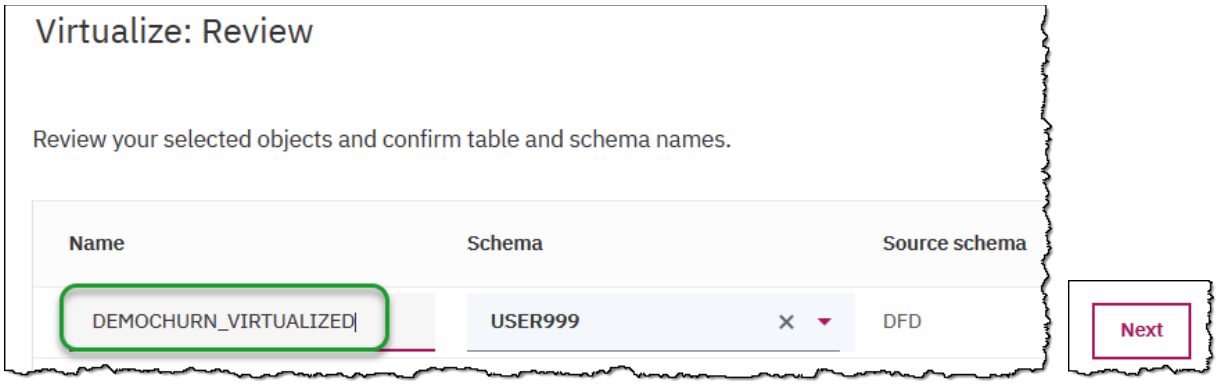
__190. Check table DEMOCHURN_TRANSFORM then Add to cart

Click on View cart (1)



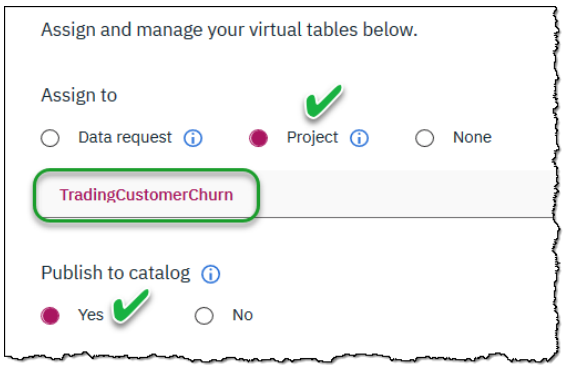
__191. In Virtualize Review screen, change the Name to DEMOCHURN_VIRTUALIZED

Click Next



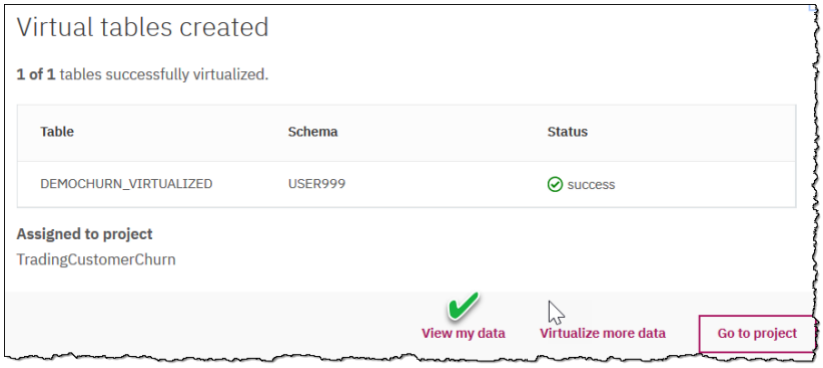
__192. Click on Assign to button ⇨ Project, then choose project: TradingCustomerChurn

Click on Publish to catalog button ⇨Yes



Click Virtualize

__193. Click View my data



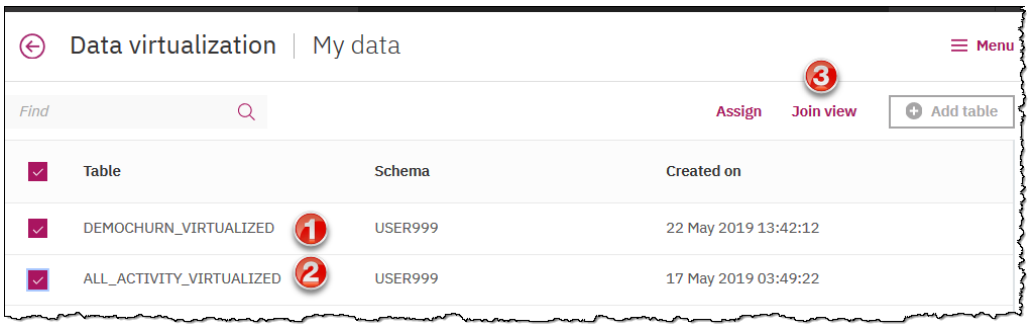
5.6 Join the two virtualized tables

Now we will join the two virtualized tables `ALL_ACTIVITY_VIRTUALIZED` and `DEMOCHURN_VIRTUALIZED`

__194. If you did not choose `View my data` from the previous screen, from the Virtualization screen you can select `Menu` ⇨ `My data`

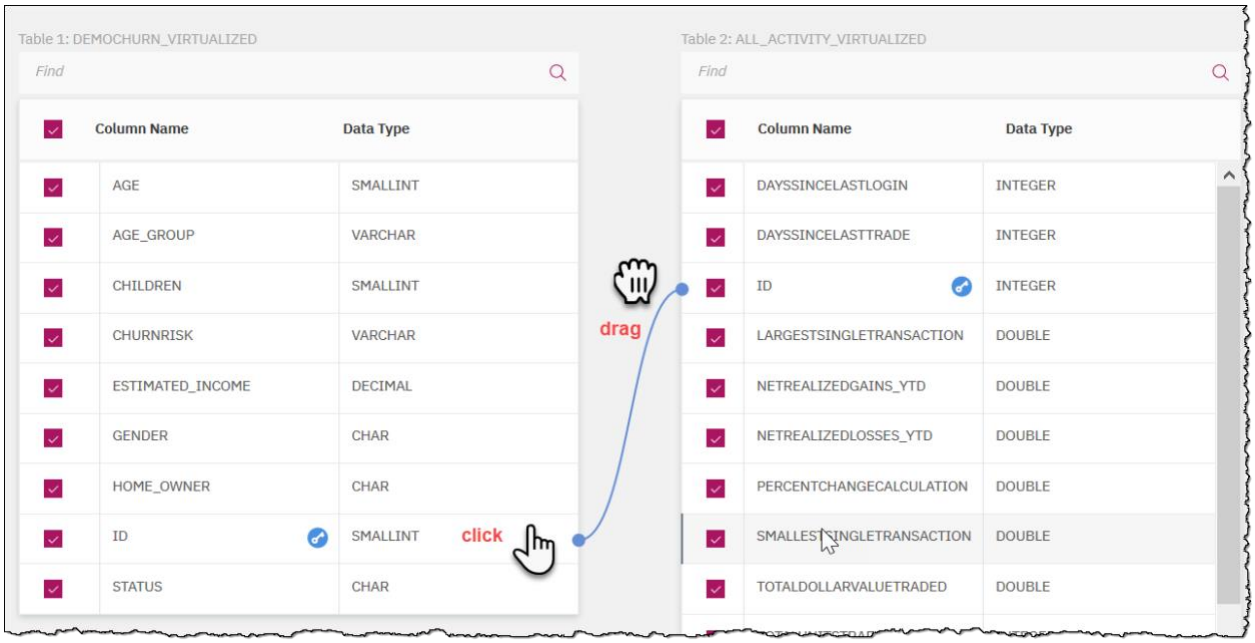
__195. Check the two virtualized tables you just created in the previous exercises

Click on `Join view`

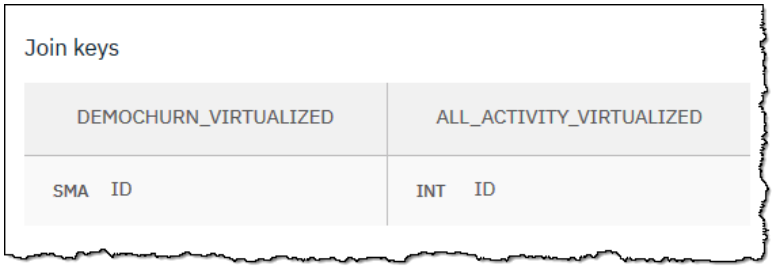


__196. Click and hold on `ID` in the first table and drag it to `ID` in the second to connect them.

Note: the order of columns on your screen may appear differently than this example.

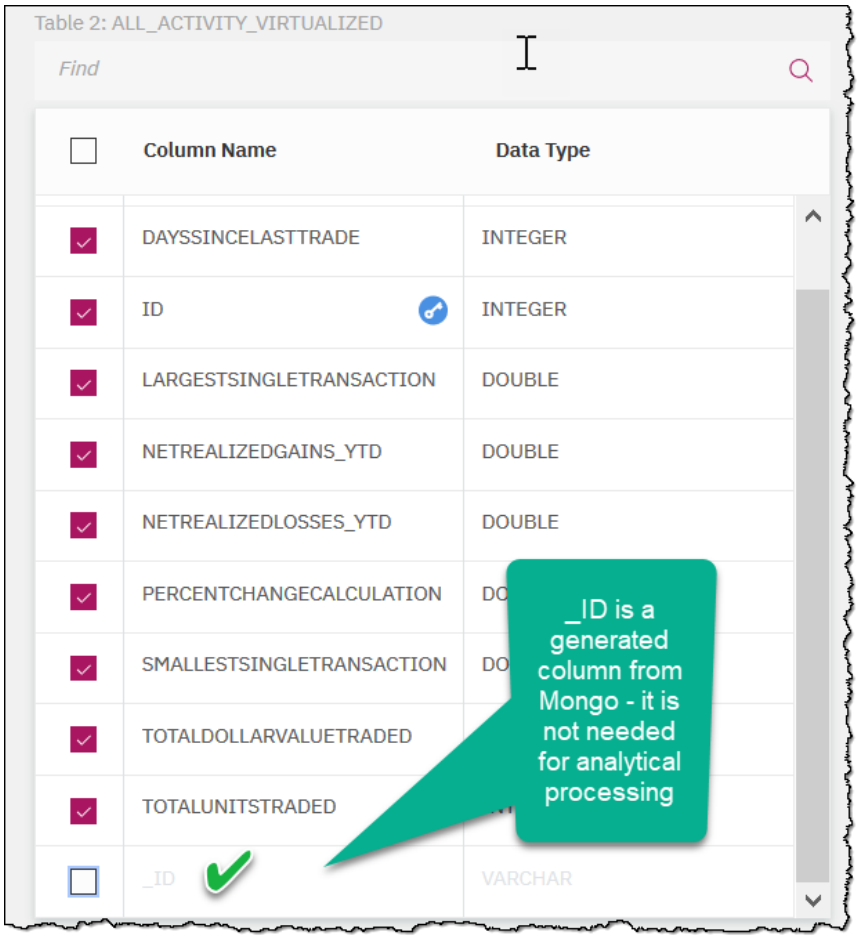


__197. This should populate the `Join keys` as shown below.



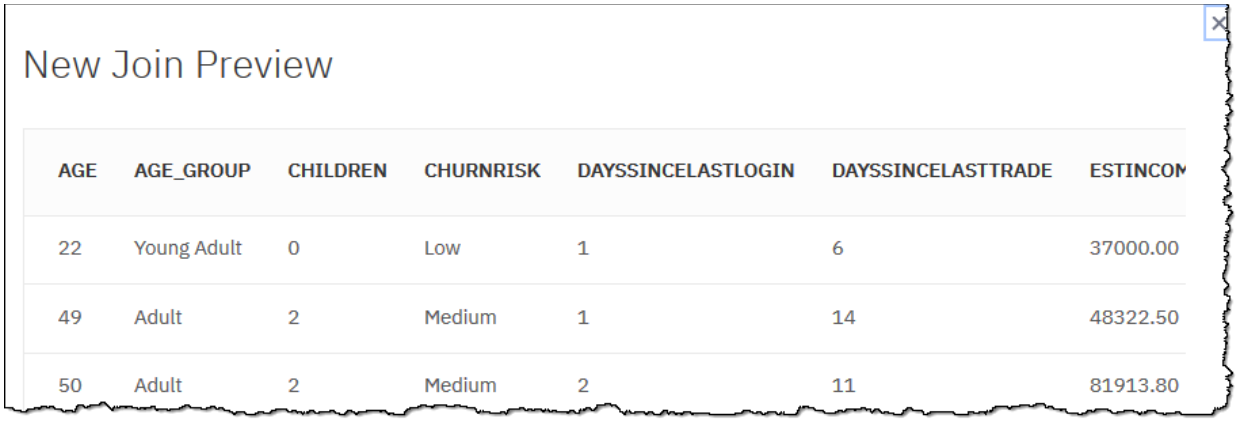
__198. Scroll to find column `_ID` in the virtualized table `ALL_ACTIVITY_VIRTUALIZED`

Uncheck it. This column is a generated column by Mongo and you will not be needing it for analytics processing.

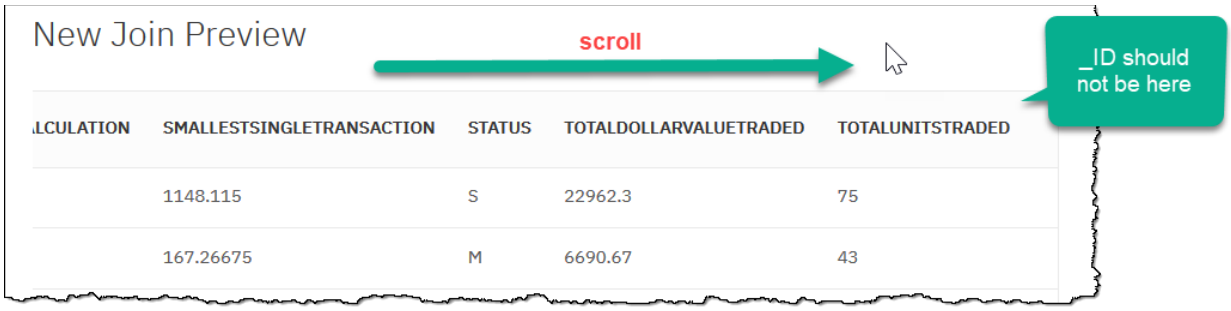


__199. Click `Preview`

The `Preview` of the `Join` should look similar to this



__200. Scroll to review the columns in the virtualized table you will be creating. `_ID` should not be there.

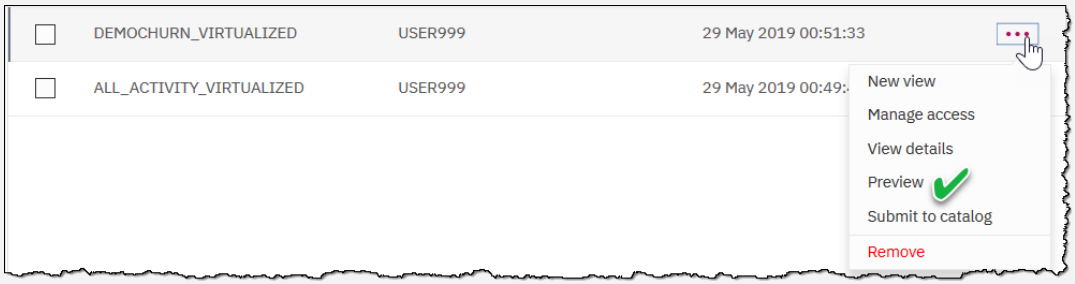


Data Engineer

If your join preview does not produce data (it is blank) one or more of your virtualized tables may not have been created correctly. To preview each virtualized table, go to:

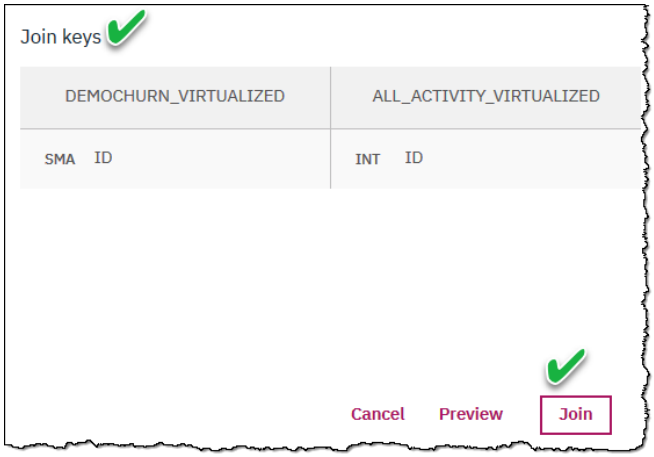
`Menu` ⇨ `My data`

Preview each virtualized table to make sure they show data.

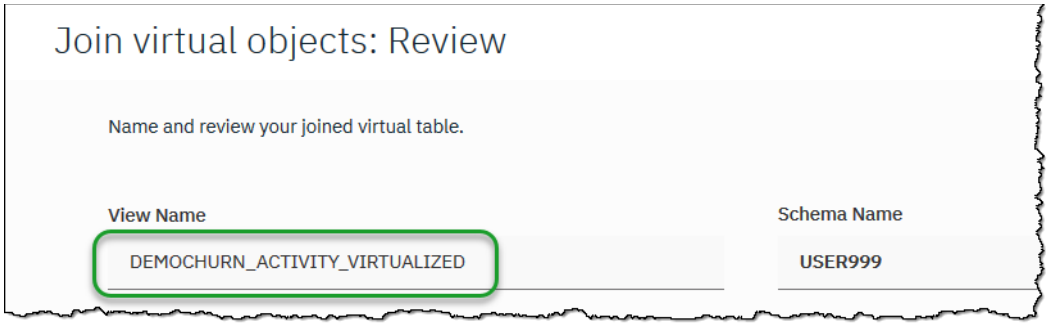


Then recreate the virtualized table that is not showing data, rejoin them, and preview again.

__201. If the preview looks good, click `Join`



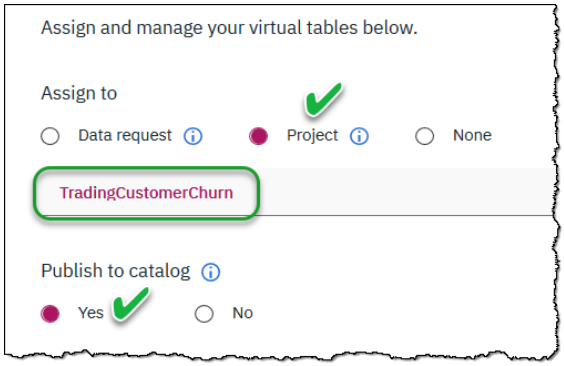
__202. Fill in View Name `DEMOCHURN_ACTIVITY_VIRTUALIZED`



__203. Click `Next`

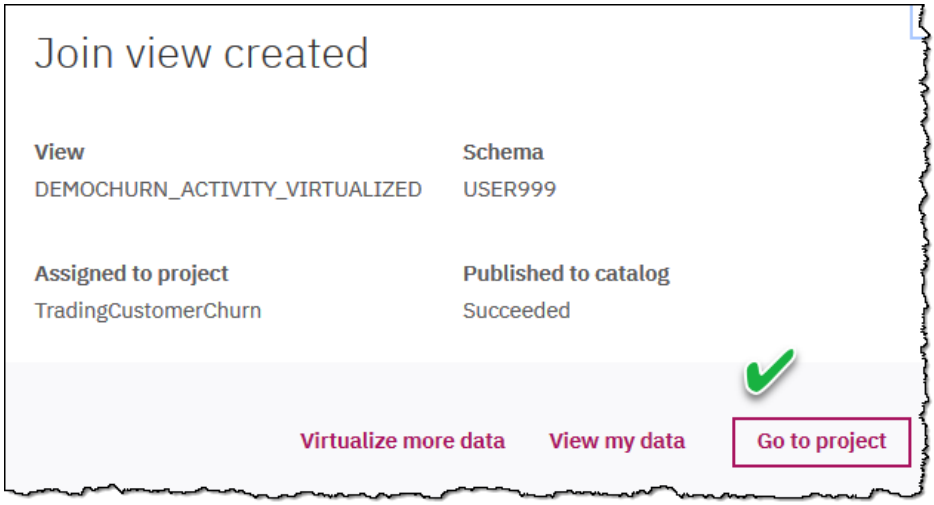
__204. Click button `Assign to` ⇨ `Project`, then choose project: `TradingCustomerChurn`

Click button `Publish to catalog button` ⇨`Yes`



Click `Create view`

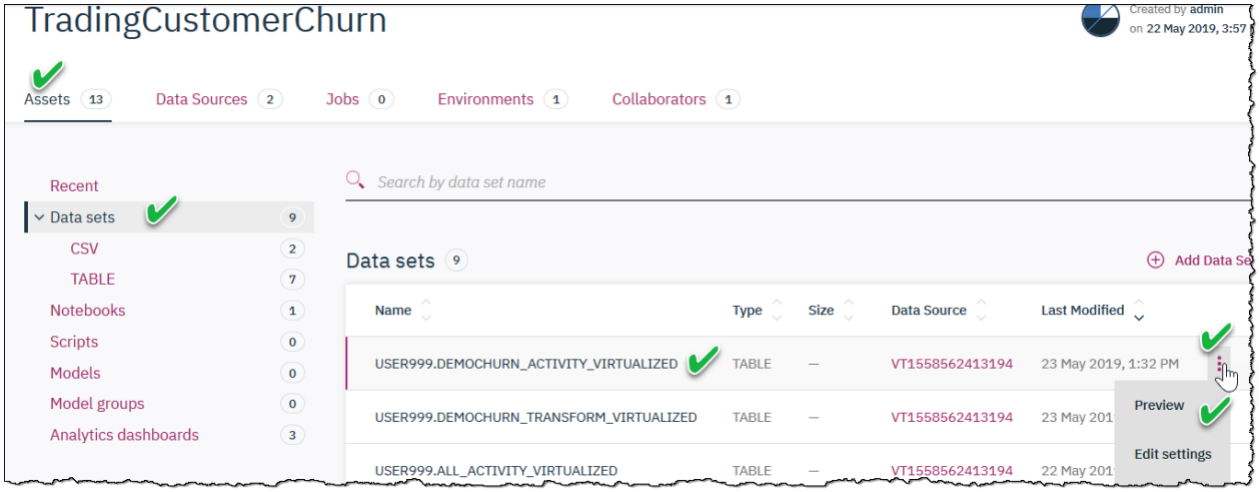
__205. Review the message and then click `Go to project`



 Data Engineer	Notice that CPD refers to joins of individual virtualized tables as “virtualized views”.

__206. `Preview` your new virtualized view in your project.

You will be using this data in the next lab to produce a Machine Learning model



__207. Remember this virtualized view represents:

1) All the Db2 demographics data (with one column transformed)

2) All the Db2 Churn data

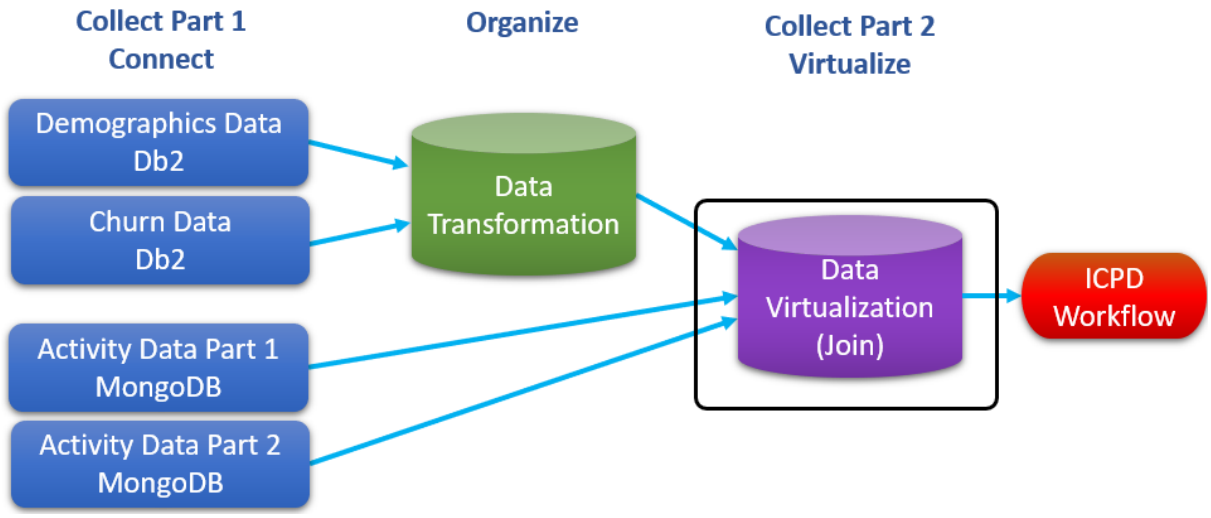
3) All the MongoDB Activity data (folded together from two databases)

Preview - USER999.DEMOCHURN_ACTIVITY_VIRTUALIZED

AGE	AGE_GROUP	CHILDREN	CHURNRISK	ESTIMATED_INCOME	GENDER	HOMEOWNER	ID	STATUS	DAYSSINCELASTLOGI
24	Adult	1	Low	38000.00	F	N	0	S	3
65	Adult	1	Medium	75004.50	M	N	6	M	4
60	Adult	0	High	19749.30	M	N	7	M	3

5.7 Lab conclusion

In this lab, you learned about `Data Virtualization` to complete the `Collect` tasks by creating a virtualized table of the transformed Db2 Demographics and Churn data, joined with the MongoDB Activity data.



**** End of Lab 05: Collect Part 2 - Virtualize**

Lab 06 Analyze Part 1 - Dashboards

6.1 Lab overview

Analyze is the third phase in the CPD platform and workflow. This is where business analysts and data scientists join forces to gain insights from their organization's data.

When embarking on machine learning projects, many organizations immediately engage these personas in hopes that they can quickly gain insight into their data, only to realize that the data they need is not available in the format required. This is where CPD helps because the Collect and Organize steps we have seen in the previous labs easily prepares the data for analytic processing.

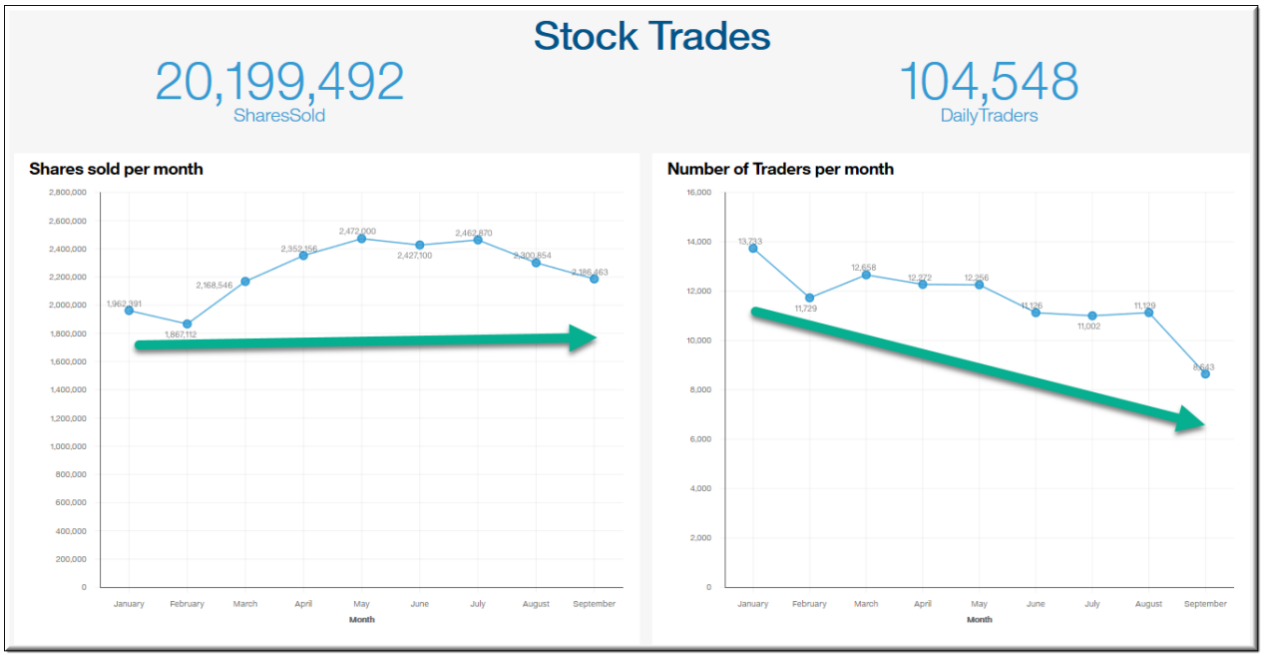
6.2 Persona represented in this lab

The Business Analyst persona is the likely to perform the exercises in this Analyze part 1 lab, and that is to create dashboards to make sense of the problems the organization is facing.

Persona (Role)	Capabilities
 Business Analyst	Business Analysts deliver value by taking data, using it to answer questions, and communicating the results to help make better business decisions.

6.3 Dashboards to help identify the focus area

In the Executive Demo lab, you were presented with a Stock Trade Opening Bell business analysis that painted the picture of how the business was doing. The dashboard provided showed a flat shares-sold, and a declining number of daily traders. A Boatswain Trading business analyst provided this information to his executives.



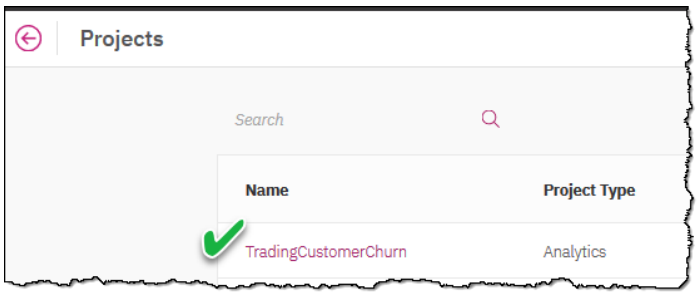
In the previous Collect Part 1 lab, we used Activity, Demographics and Churn data together. In this fictitious scenario, the Churn data shows `Low`, `Medium` and `High` risk of the customer potentially leaving Boatswain Trading.

The output is a dataset called `MERGED_DEMOGRAPHICS_TRADING_CUSTOMER` that business analyst used to build a dashboard to better understand the details of the problem.

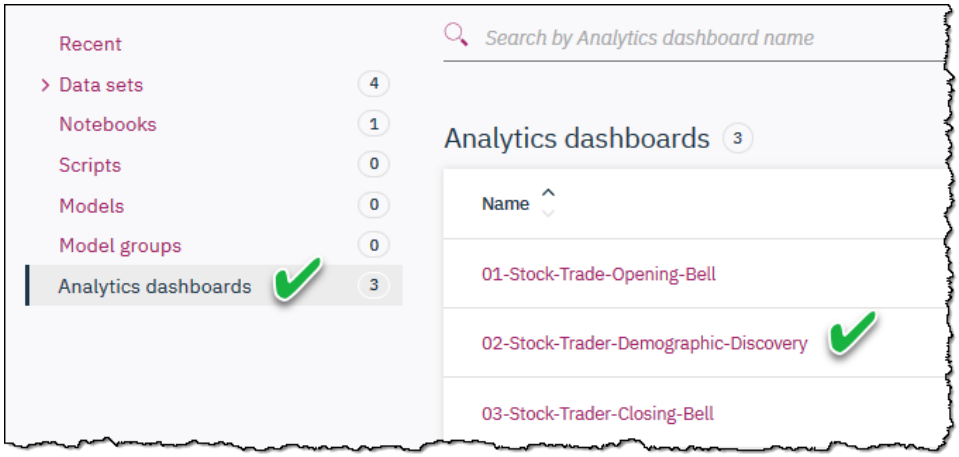
Let us see how this looks.

6.3.1 Business dashboard – analyze customer demographics

__208. From the `Navigation Menu` ⇨ `Projects` and select project `TradingCustomerChurn`.



__209. Click `Analytics Dashboards` ⇨ `02-Stock-Trader-Demographic-Discovery`



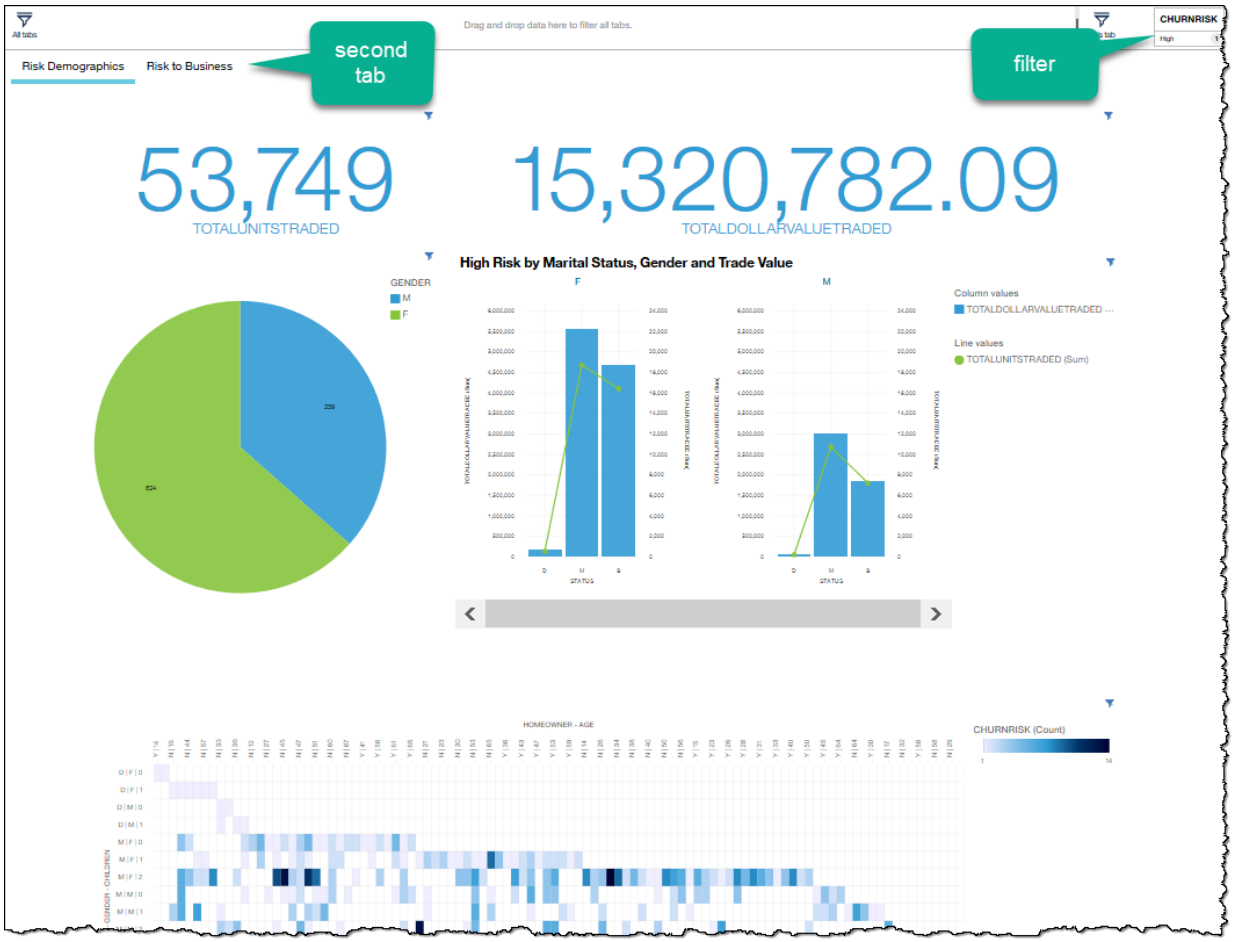
Business Analyst

The `Analytics dashboards` you will be using in this exercise use the built-in `Cognos dashboards` of CPD.

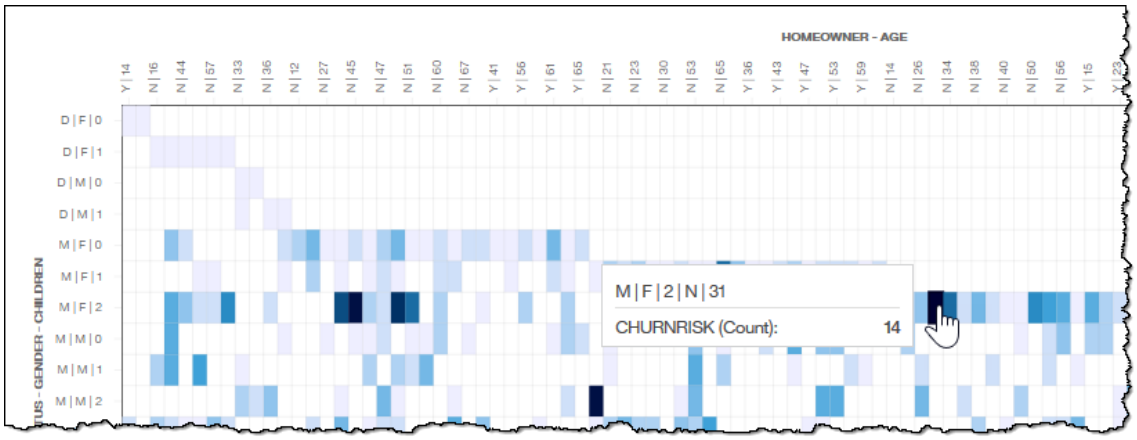
You can install and enable the `Cognos Analytics` add-on for more power and capability, with features such as:

- Automated data preparation, modeling and modules
- Automated creation of visualizations and dashboards
- Reporting and stories
- Data exploration with a natural language interface
- Dynamic query

__210. This is the dashboard that the Boatswain Trading business analyst created to break down the demographics of their customer base, based on the likeliness of churn.



__211. Note that this dashboard is interactive. For example, if you click on a dark square in the heatmap (the darker the square, the more likely to churn) you can see in the example below that this data point represents a married female with 2 children, not a homeowner, and is 31 years of age. The rest of the visualizations on this tab will change based on this selection as well, which provides even more insight into this data point.



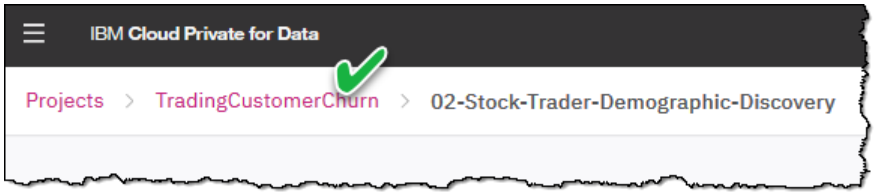
Dashboards like this can paint a high-level picture of the customer demographic Boatswain Trading needs to target in order to reverse their declining revenue trend.

Note too that you can change the `CHURNRISK` filter using `Low`, `Medium` and `High` to help better understand the data.

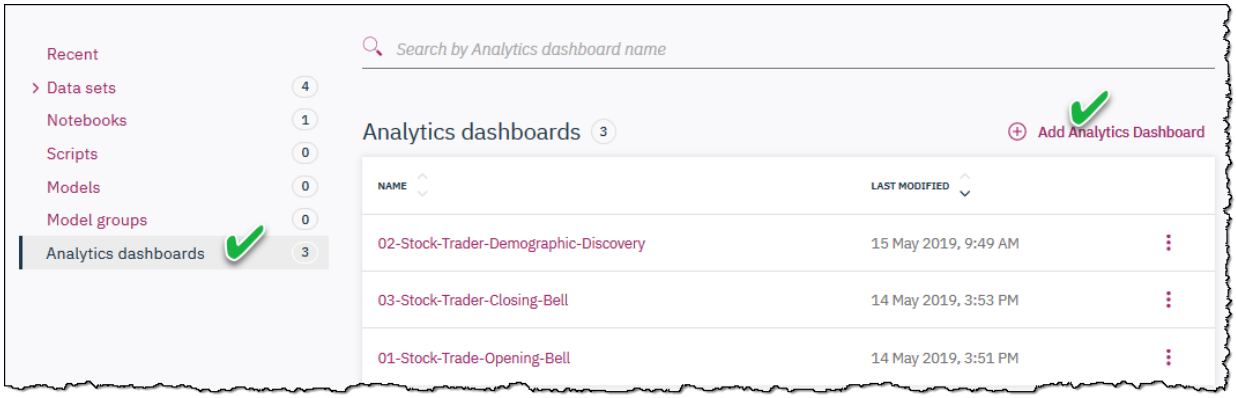
6.3.2 Building the dashboard

Let us build this dashboard from the beginning to understand how this works in CPD.

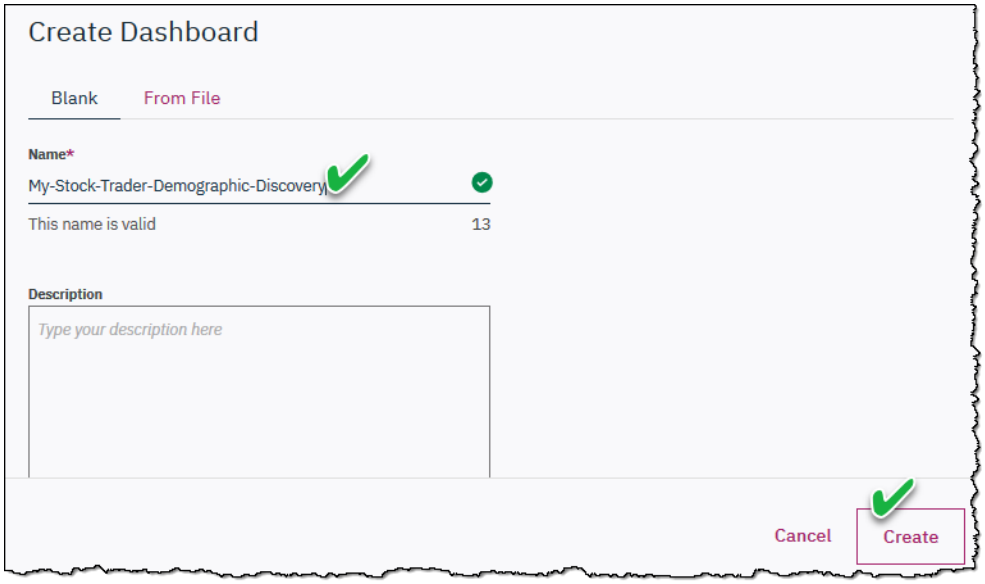
__212. In the navigation bar on the top right, click on the project: `TradingCustomerChurn`.



__213. Click `Analytics dashboards` ⇨ `Add Analytics Dashboard`

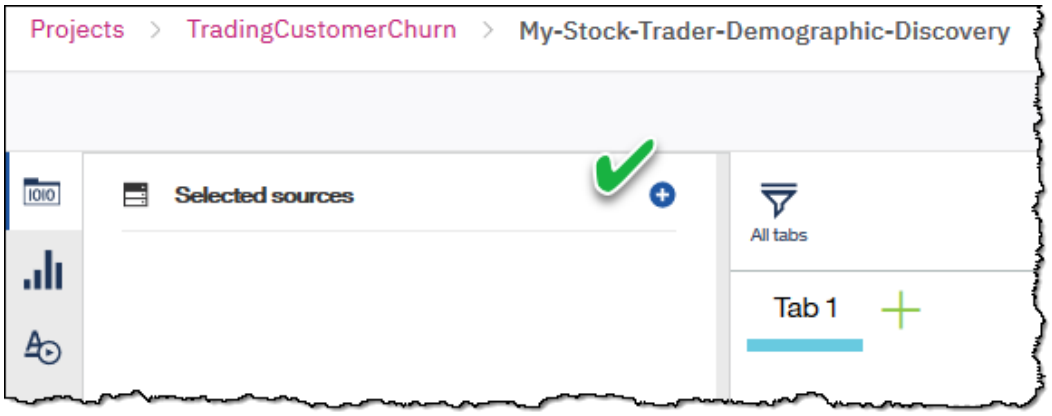


__214. Name this new dashboard: `My-Stock-Trader-Demographic-Discovery` and click `Create`.



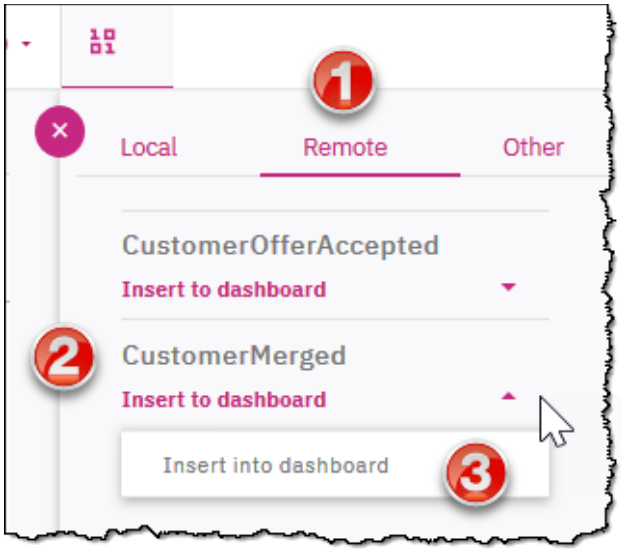
__215. Select the template shown below and click `OK`.

__216. Click the + next to `Selected sources`



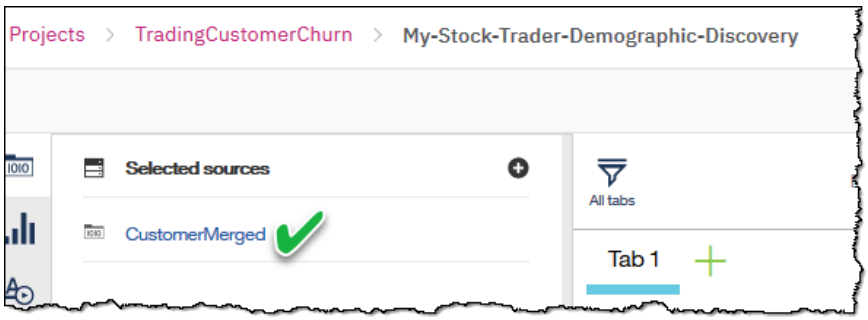
__217. In the top right portion of the screen, click `Remote` on the flyout menu.

Expand `CustomerMerged` using the down arrow and the click `Insert to dashboard`.

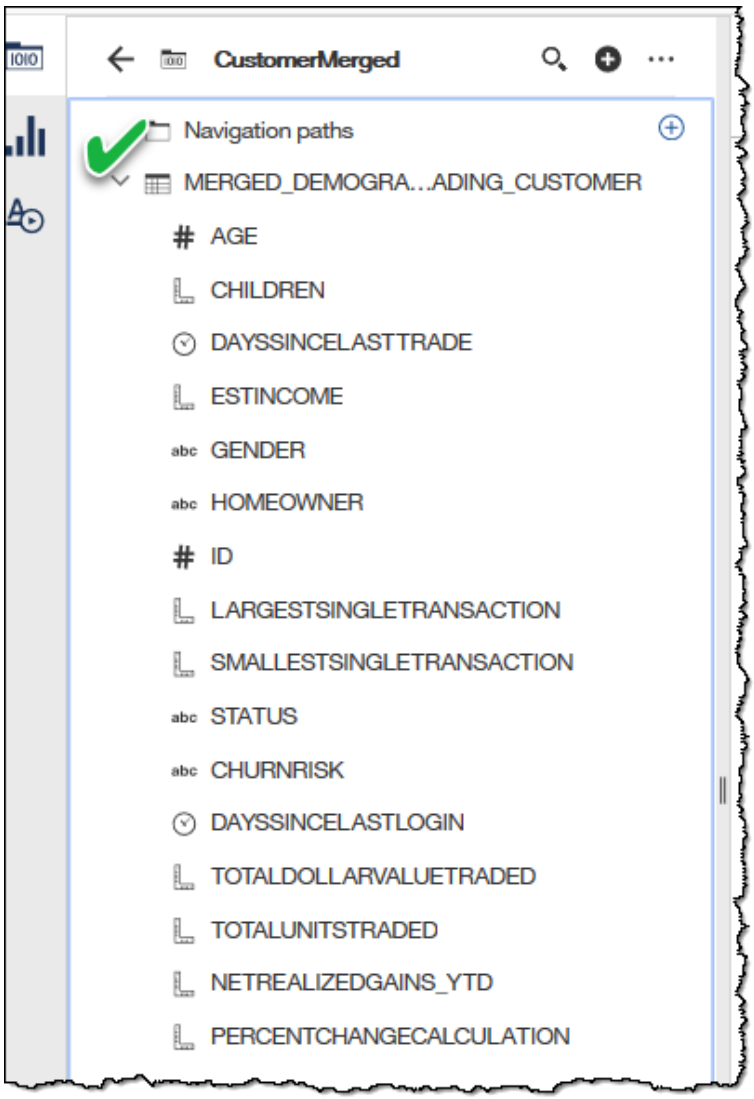


__218. Notice in the top left of your screen a data set named `CustomerMerged`, is inserted in the `Selected sources` box. This is actually a name representing a table we saw earlier in the Db2 Warehouse when we were reviewing the connection to that database.

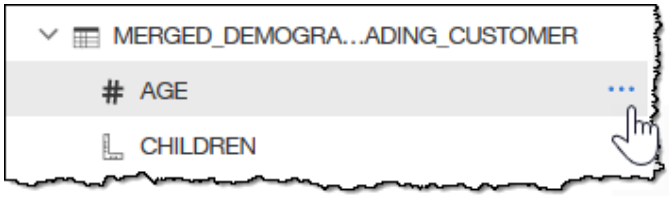
Click `CustomerMerged` to select it for use in our dashboard.



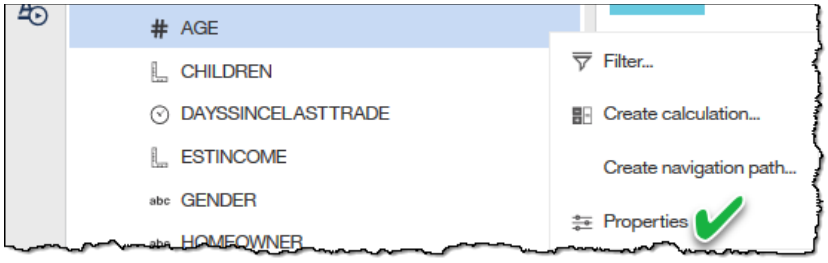
__219. Expand the table details by clicking the twistie next to the table name `MERGED_DEMOGRAPHICS_TRADING_CUSTOMER`.



__220. Click `Age` and then click ellipse (right side).

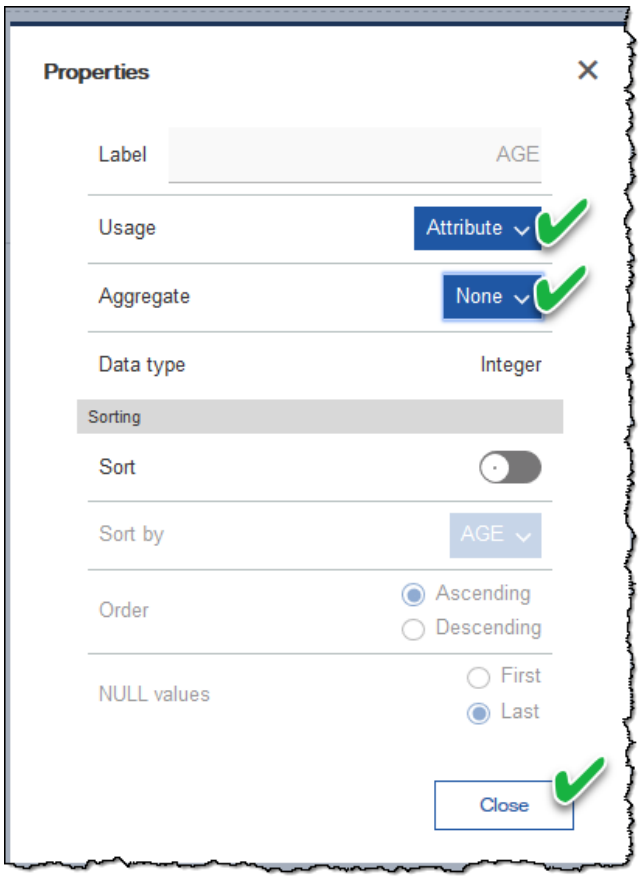


__221. Click `Properties`.



__222. Select `Attribute` from the *Usage* drop-down menu and `None` from the *Aggregate* drop-down menu.

Click `Close` to finish the selection.

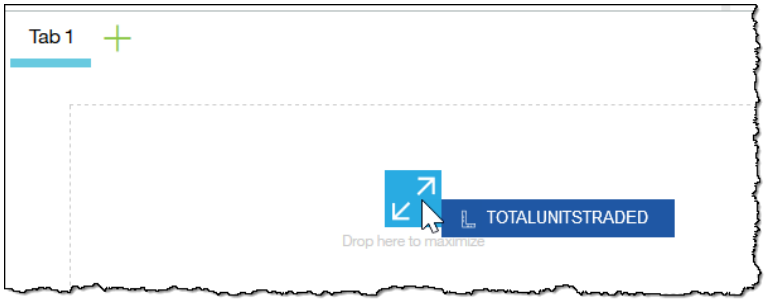


__223. Repeat same for all columns in this table, setting their `Usage` and `Aggregate` values as shown below. Note - the key ones that can affect the dashboard are in red, be sure to get those correct.

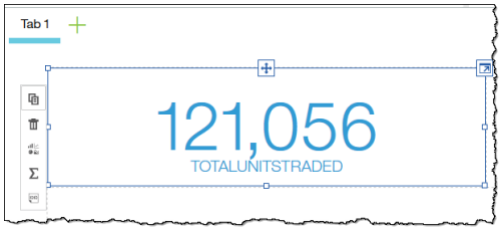
Column Name	Usage	Aggregate
AGE	Attribute	None
CHILDREN	Attribute	None
DAYSSINCELASTTRADE	Measure	None
ESTINCOME	Measure	None
GENDER	Attribute	None
HOMEOWNER	Attribute	None
ID	Identifier	Count
LARGESTSINGLETRANSACTION	Measure	None
SMALLESTSINGLETRANSACTION	Measure	None
STATUS	Attribute	None
CHURNRISK	Attribute	None
DAYSSINCELASTLOGIN	Measure	None
TOTALDOLLARVALUETRADED	Measure	Total
TOTALUNITSTRADED	Measure	Total
NETRALIZEDGAINS_YTD	Measure	None
PERCENTCHANGECALCULATION	Measure	None

__224. When you are done adjusting the properties of all the data columns, click the save icon to save the work you have done so far.

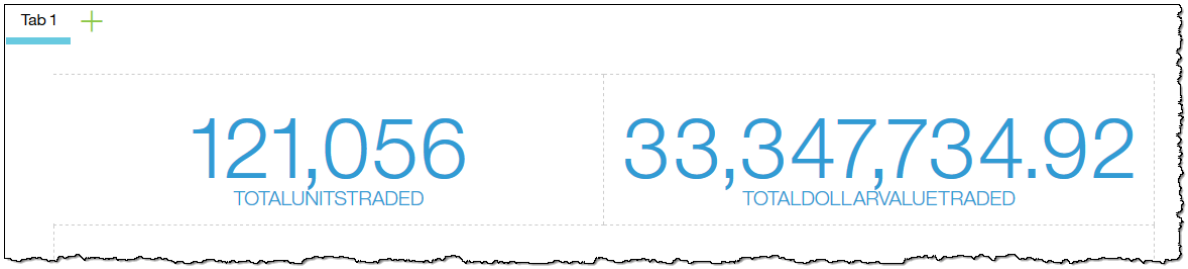
__225. Drag `TOTALUNITSTRADED` column towards the text `Drop here to maximize` in the top left of the screen. The icon color will change to light blue when you hover over it. Drop the column there to occupy the full space of the dotted lined box in the dashboard template.



__226. The result should look like this:



__227. Similarly drag TOTALDOLLARVALUETRADED to the top right and drop it when the icon turns light blue. The result should look like this:



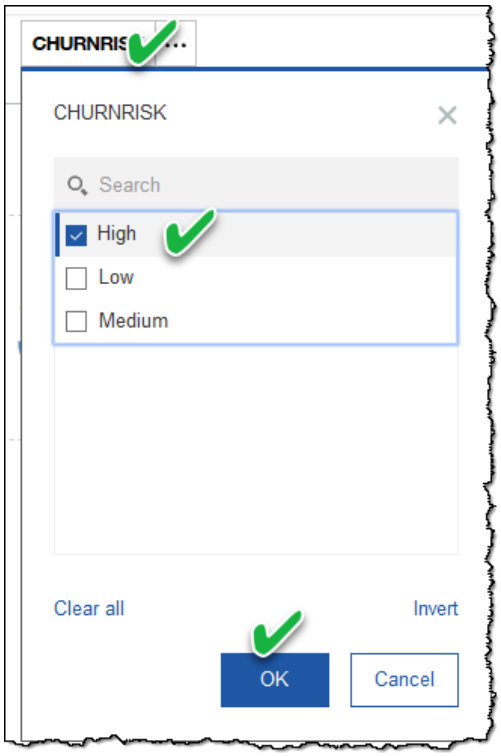
__228. Now that we have scored a churn risk of existing customers in our data, let's discover the traits of those customers are by using dashboard visualizations.

Drag CHURNRISK and drop it on the **This Tab** text at the top right next to the filter icon.

(Note how one is "This Tab" and the other is "All tabs". We want "This tab".)

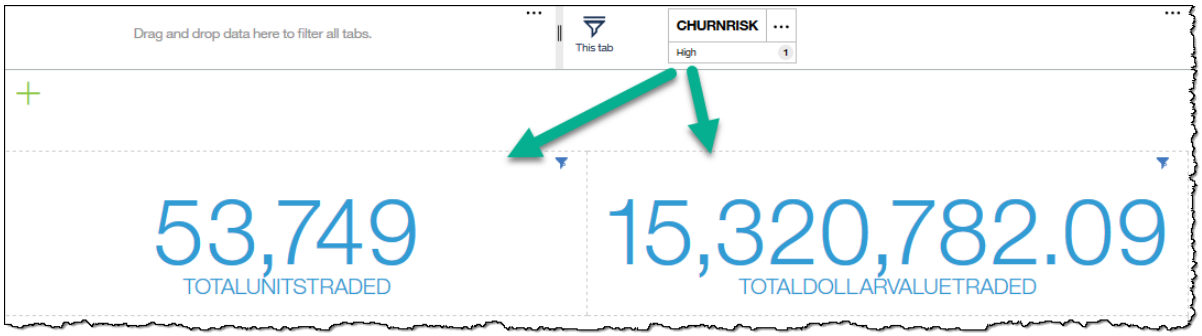
__229. Click CHURNRISK to expand it and then select the box: High.

Click OK to finish this filter selection.

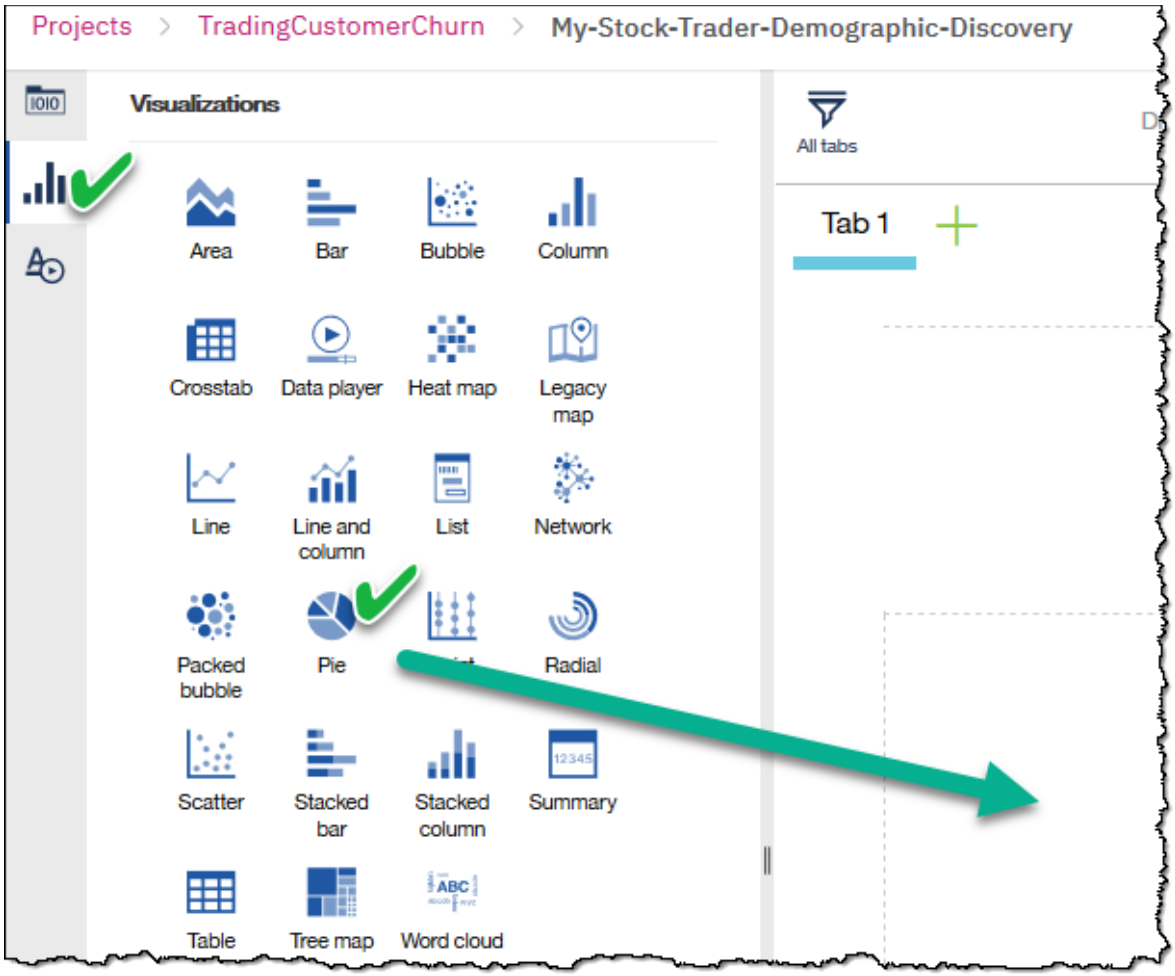


__230. Notice the amounts change in this tab's visualizations after applying a filter of customers with a `High` churn risk. Because the filter is in box called `This tab`, all visualizations in this dashboard's tab will be affected by the filter. Other tabs created later will not affected by this particular filter.

Each visualization in the tab now shows a tiny filter icon in the top right corner to signify that the displayed data has a filter applied to it.

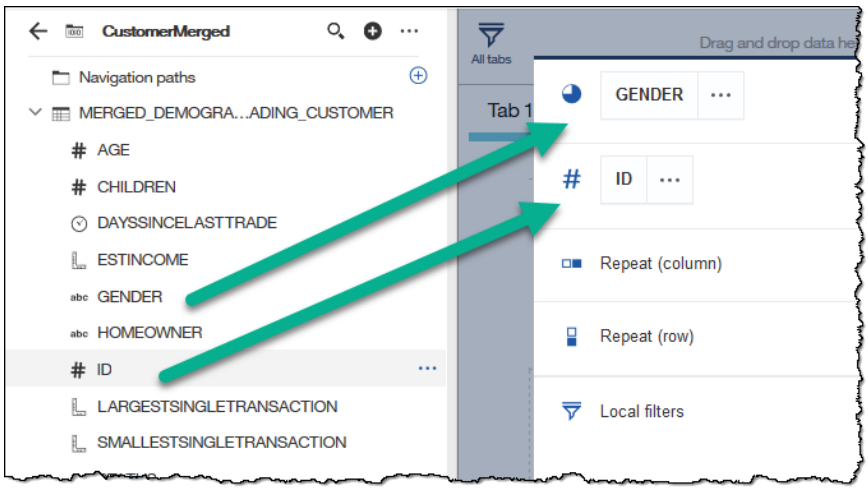


__231. Select the Visualizations icon on the left side of your dashboard work area to expand the options for Visualizations. Choose the `Pie` chart option and drag it to bottom left side of the dashboard work area.

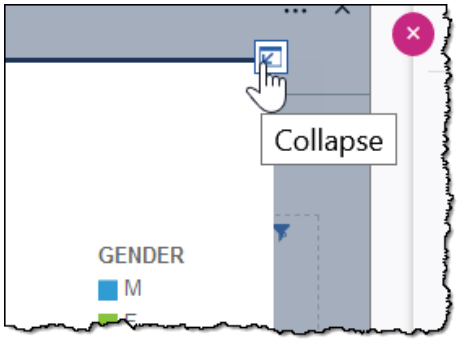


__232. The Pie chart build screen will automatically expand and allow you to begin selection of the columns required to complete the chart.

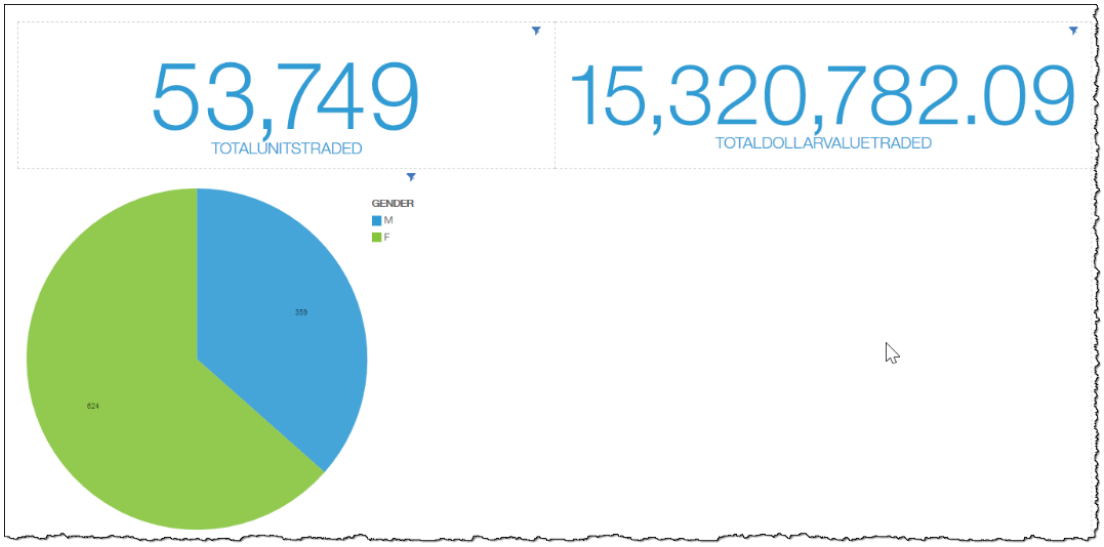
Drag `GENDER` to the top `Segments` section (next to the pie icon) and drag `ID` to the `Size #` section.



__233. Click the `Collapse` icon on the top right corner of the pie chart build screen.

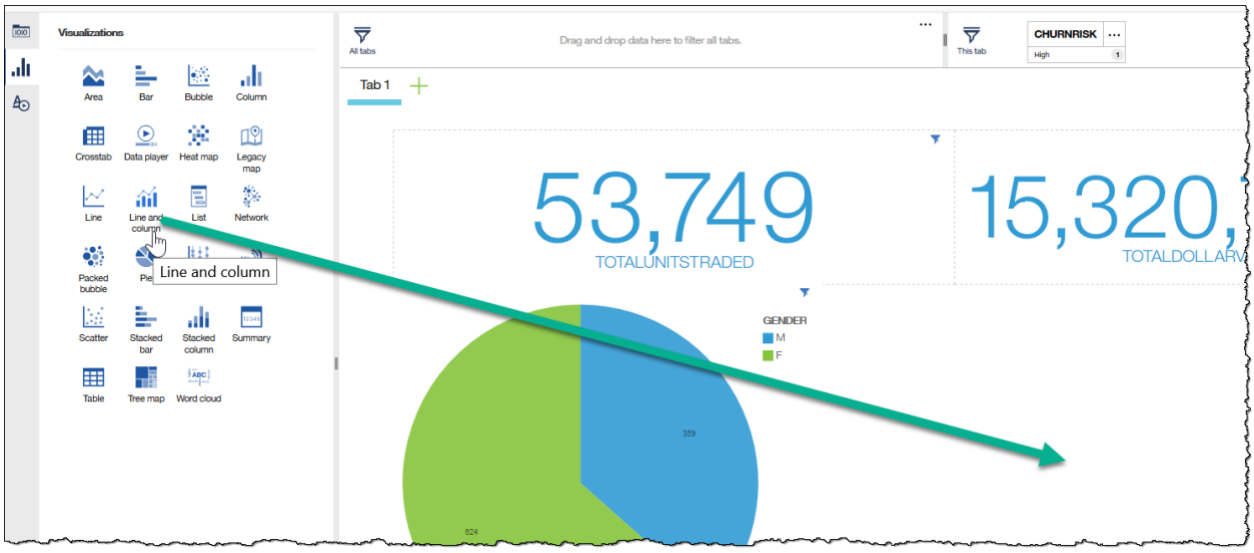


__234. Your dashboard should look something like this. If it does not, take some time to resize and move your pie chart and other visualizations to look like this.

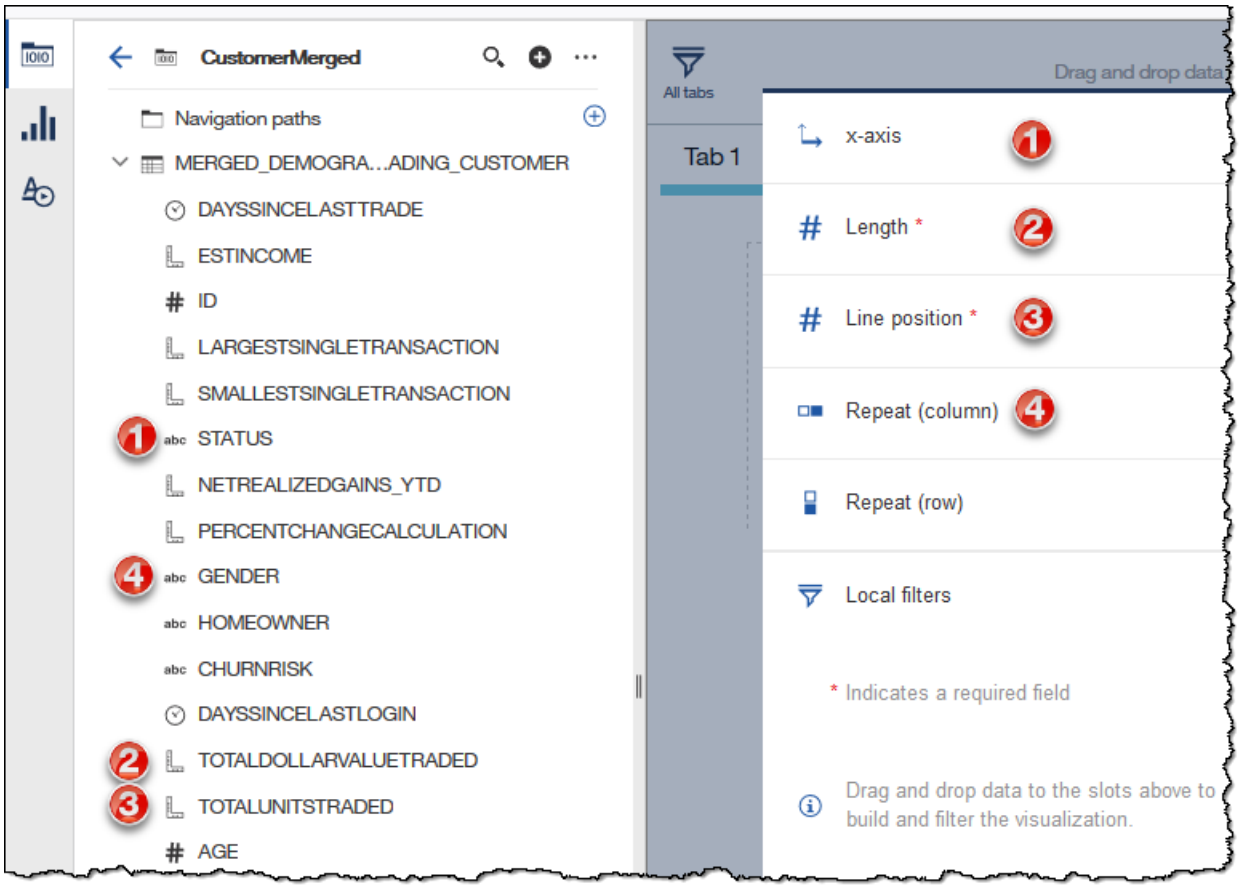


__235. Let us now find out who makes the most trades for the most amount of trade value.

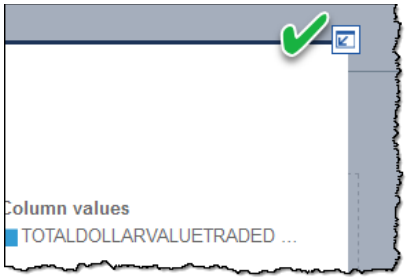
From the Visualizations options, drag `Line and Column` to the bottom right side of the work area.



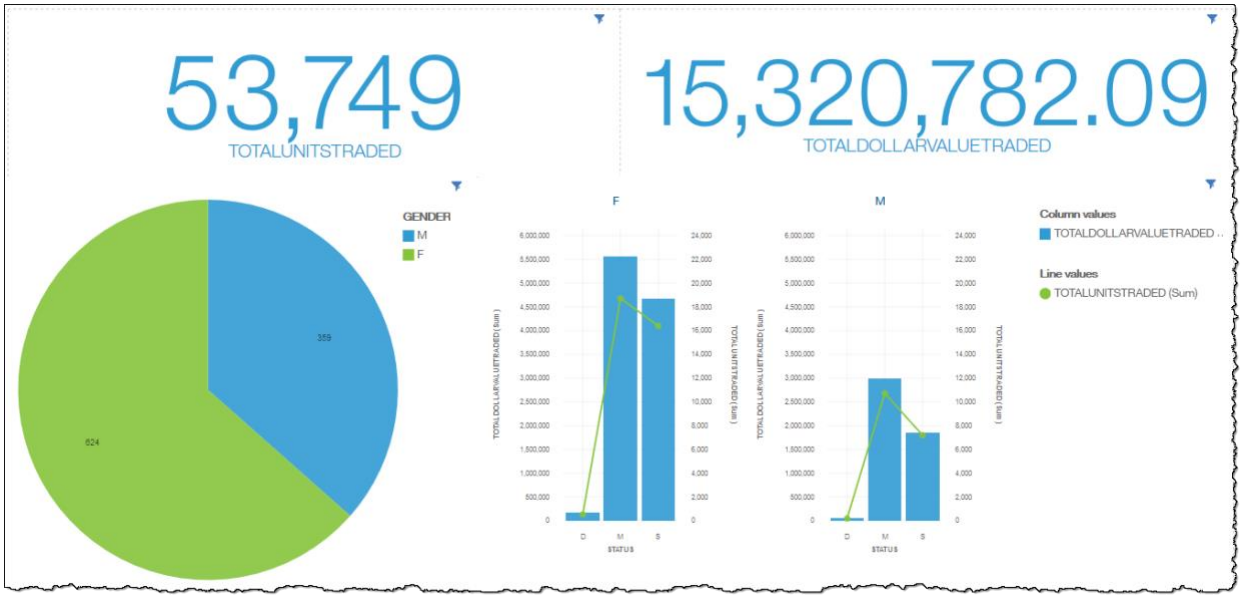
__236. Drag each of the numbered items to the right to the position indicated (drag `STATUS` to the `x-axis`, then `TOTALDOLLARVALUETRADED` to `Length`, and so on.)



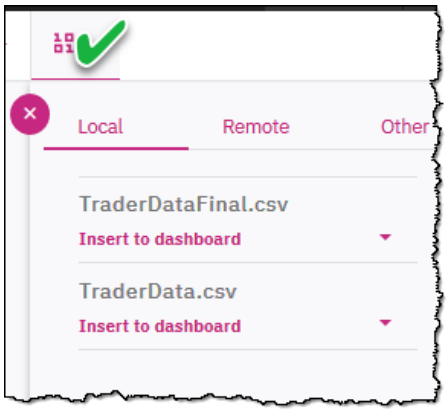
__237. Collapse the chart.



__238. The dashboard should look like below. If not, take time to resize and move the last visualization to make your dashboard like this.

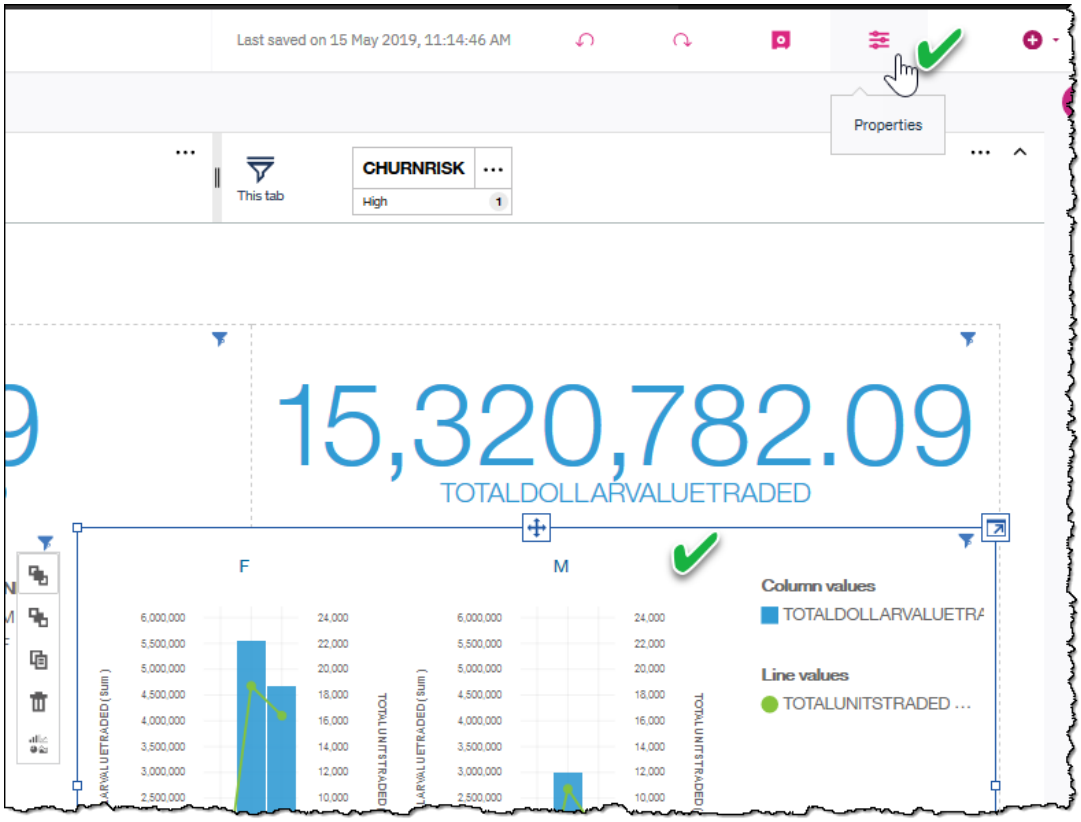


__239. You can optionally make some room in your work area by hiding the right palette by clicking the `Find Data` icon.

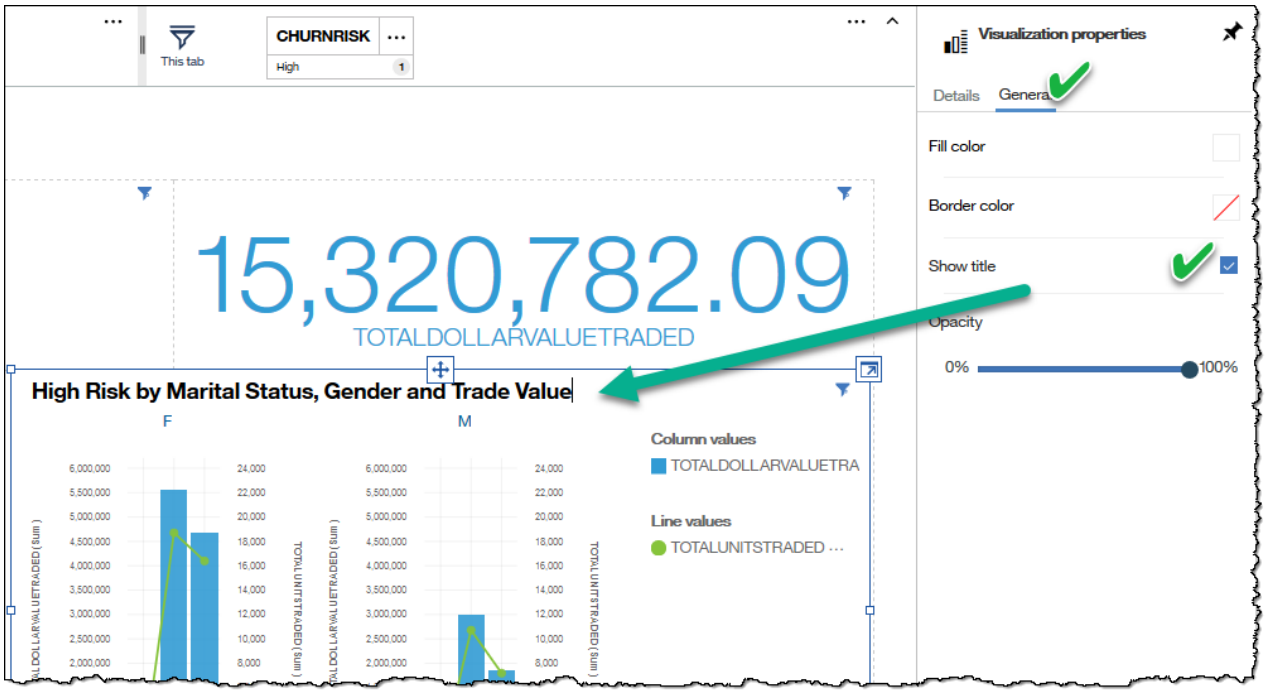


__240. Click the save icon to save the work you have done so far.

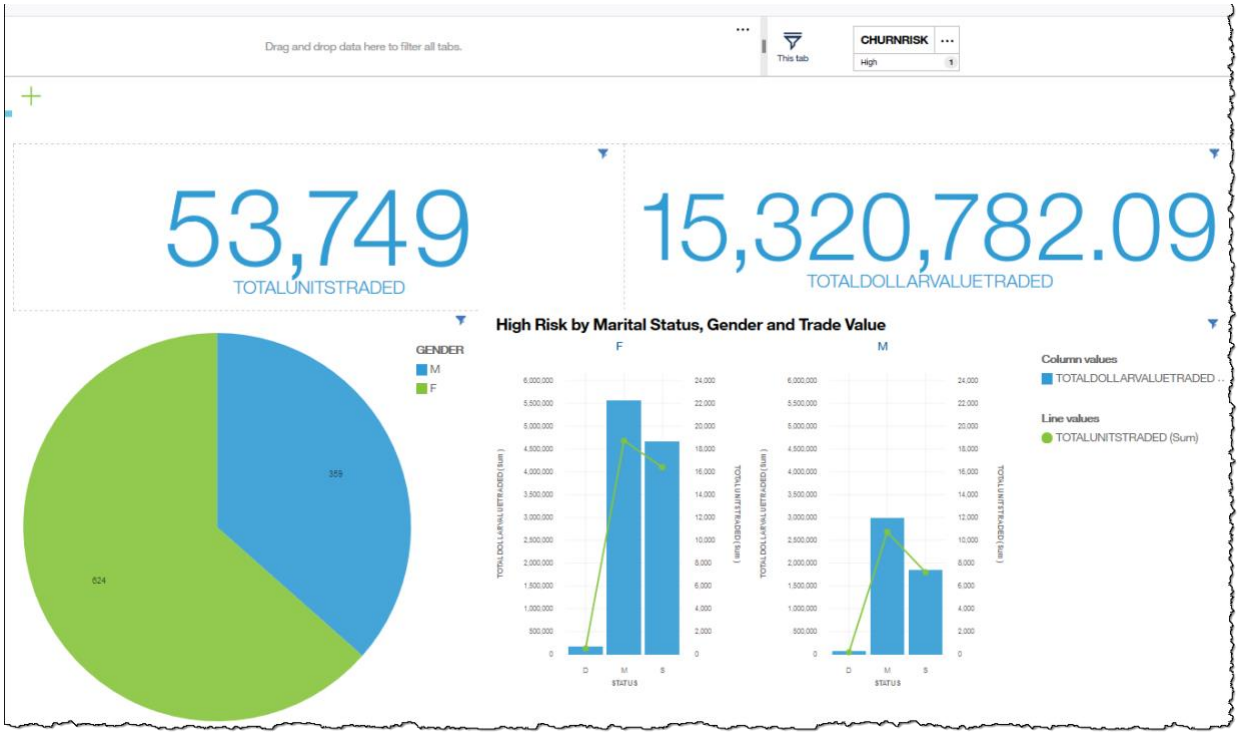
__241. Click on the line and column visualization that you just created and click the `Properties` icon from top menu bar. This allows you to edit the properties of this specific visualization.



__242. Click the `General` tab. Check `Show Title` and in the visualization you can now type: `High Risk by Marital Status, Gender and Trade Value`



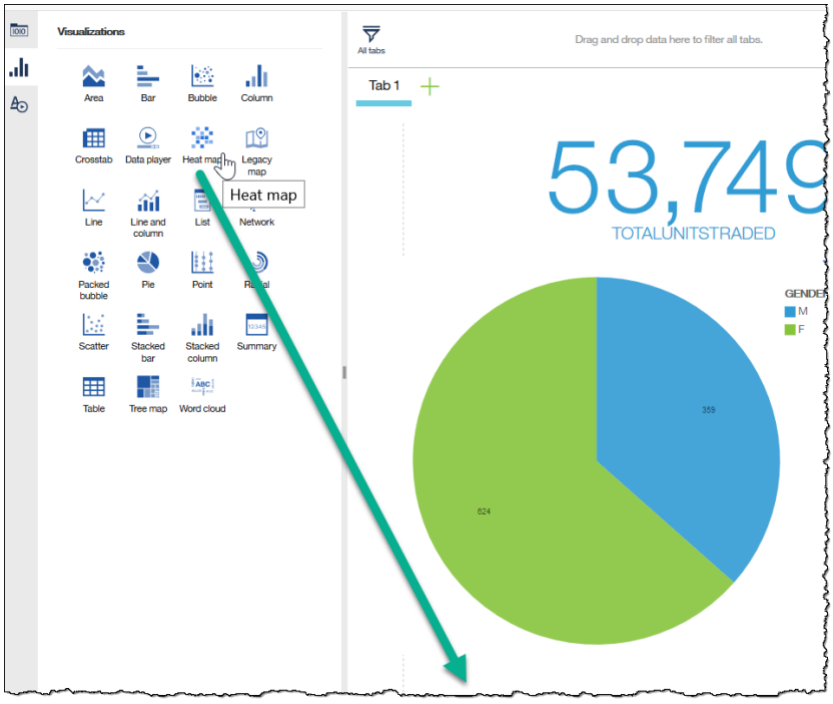
__243. Close the properties section when done. Your dashboard should look like this:



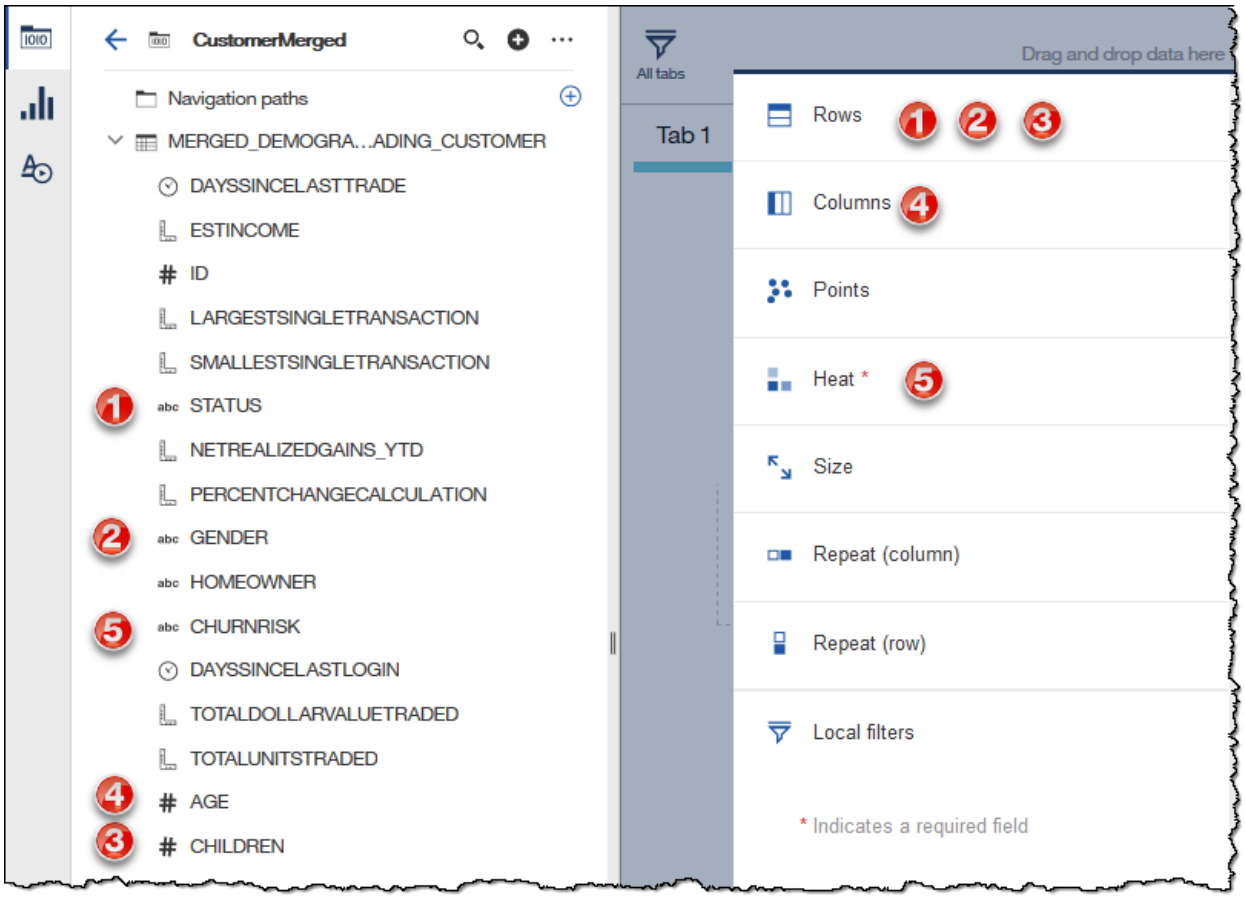
It appears from the chart that the greatest number of high-risk customers who trade and spend the most with us are Married and Single Females.

Is there any other insight we can gain? For example, does the number of children or home ownership play a part?

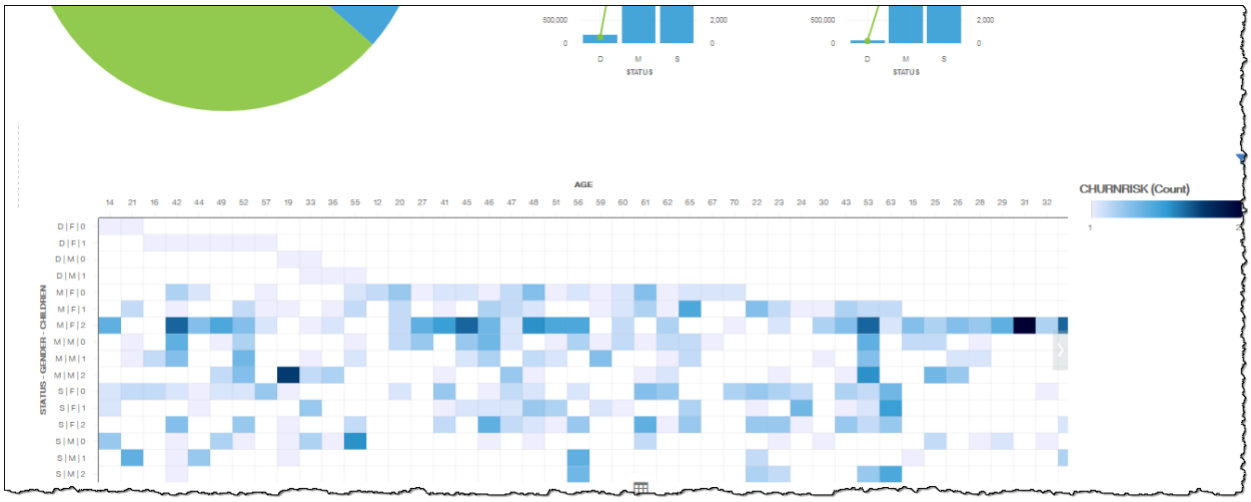
__244. From the `Visualizations` palette options, drag `Heat Map` to the bottom, below the pie chart.



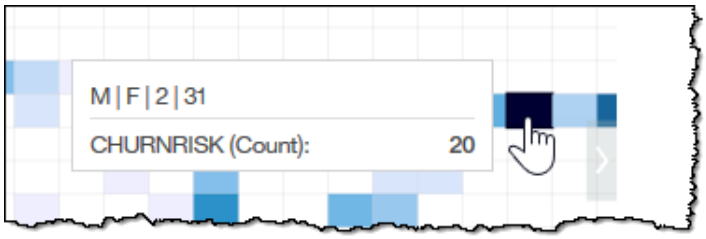
__245. Drag the `STATUS`, `GENDER`, and `CHILDREN` individually to the `Rows` section, make sure to keep them in that order. Drag `AGE` to the `Column` section and drag `CHURNRISK` to the `Heat` section, as shown below



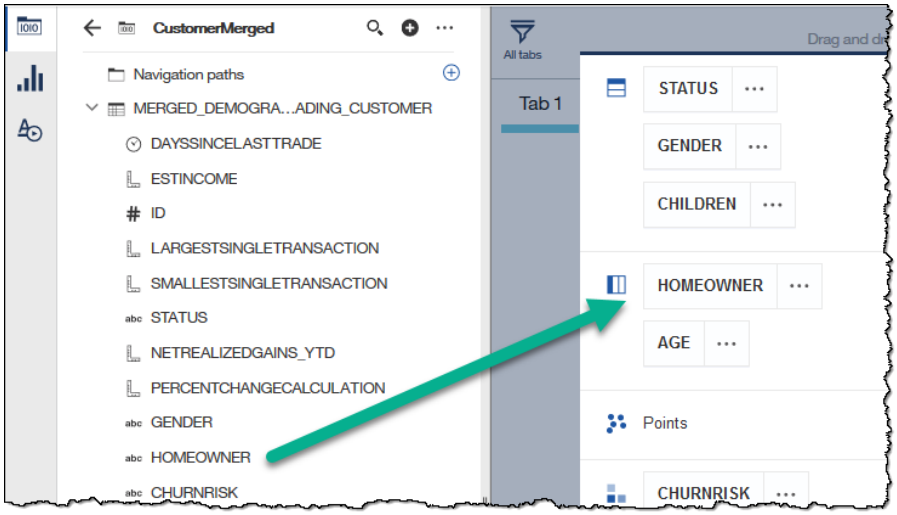
__246. Collapse the `Heat Map` visualization and then resize and move to take up all the space below the previous visualizations.



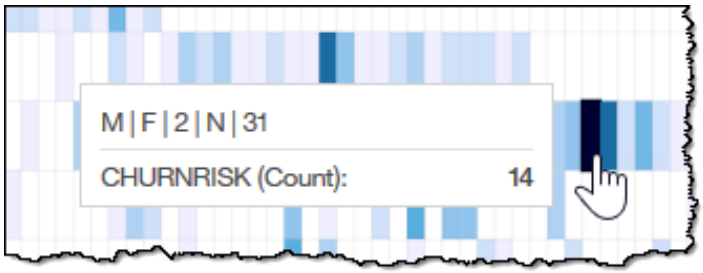
__247. Highlighting one dark data point on this visualization (the darker, the more likely to churn) shows at risk are married females 31 years old with 2 children.



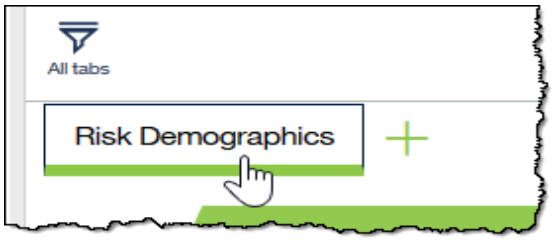
__248. Expand the `Heat Map` visualization again and drag `HOMEOWNER` above `AGE` , but do not replace `AGE`. Collapse the visualization again. (Note: Do not replace age. Make sure to place right above it.)



__249. Highlighting one of the darkest data points on this heatmap now shows a married female of 31 years, with 2 children and doesn't own a home has one of the greatest risks for churn at Boatswain Trading.



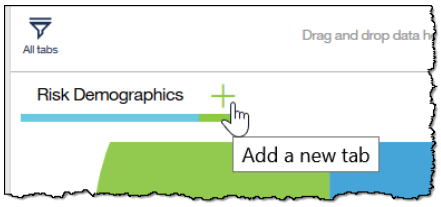
__250. Change the title of the first tab to `Risk Demographics` By selecting the tab and then the Pencil icon to edit it.



6.3.3 Analyze risk to the business

__251. Let's create another tab to see the risk to the business.

Next to the existing tab name, click `+` to add a new tab.

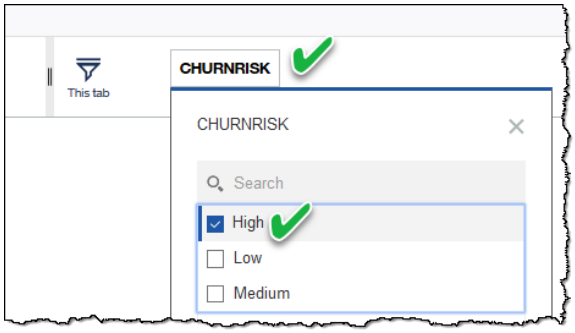


__252. Select the `Freeform` template and click `Use`.

__253. Change the tab title from `Tab 1` to `Risk to the Business`.

__254. To focus on the high risk groups, we will begin by creating a series of filters

Drag column `CHURNRISK` to the top right filter area that has the text: *Drag and drop data to filter this tab*. Click it and check `High` and click `OK`.

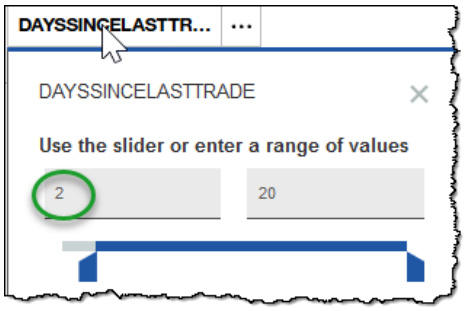


__255. Drag `GENDER` into the same filter area and check `Female`. Click `OK`.

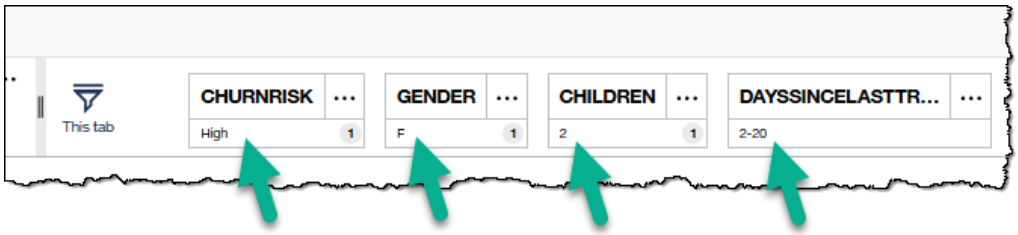
__256. Drag `CHILDREN` into the same filter area and select `2` and click `OK`.

__257. Drag DAYSSINCELASTRADE into the same filter area select 2 for the lower bound and keep the upper bound to 20.

Click OK.



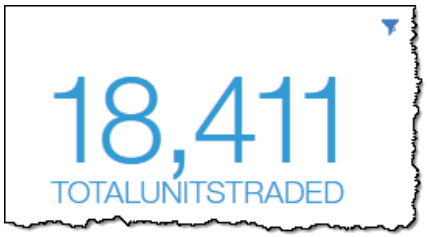
__258. The filter area should now look like this:



__259. Now we have applied filters that correspond to our previous findings, we can build the dashboard to drill into the details.

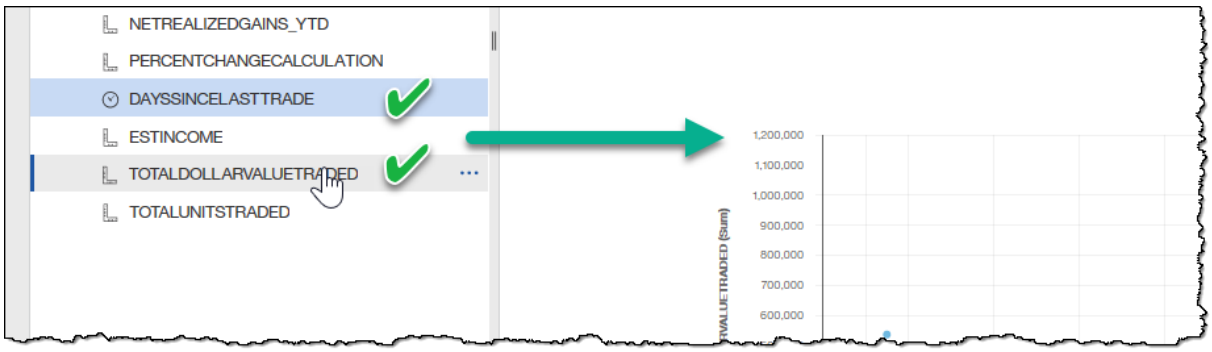
Drag TOTALDOLLARVALUETRADED to the top left area of the canvas. Resize the chart so it appears in the top left corner. This creates a single value field that is filtered by all the tab filters we created.

__260. Pull over TOTALUNITSTRADED over to the top right area of the canvas. Resize the chart so that it appears in the top right corner.

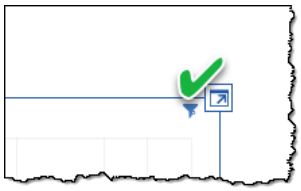


__261. Let's see the TOTALDOLLARVALUETRADED at risk by this group based upon the days since they last traded with us and the date they last logged in to our system.

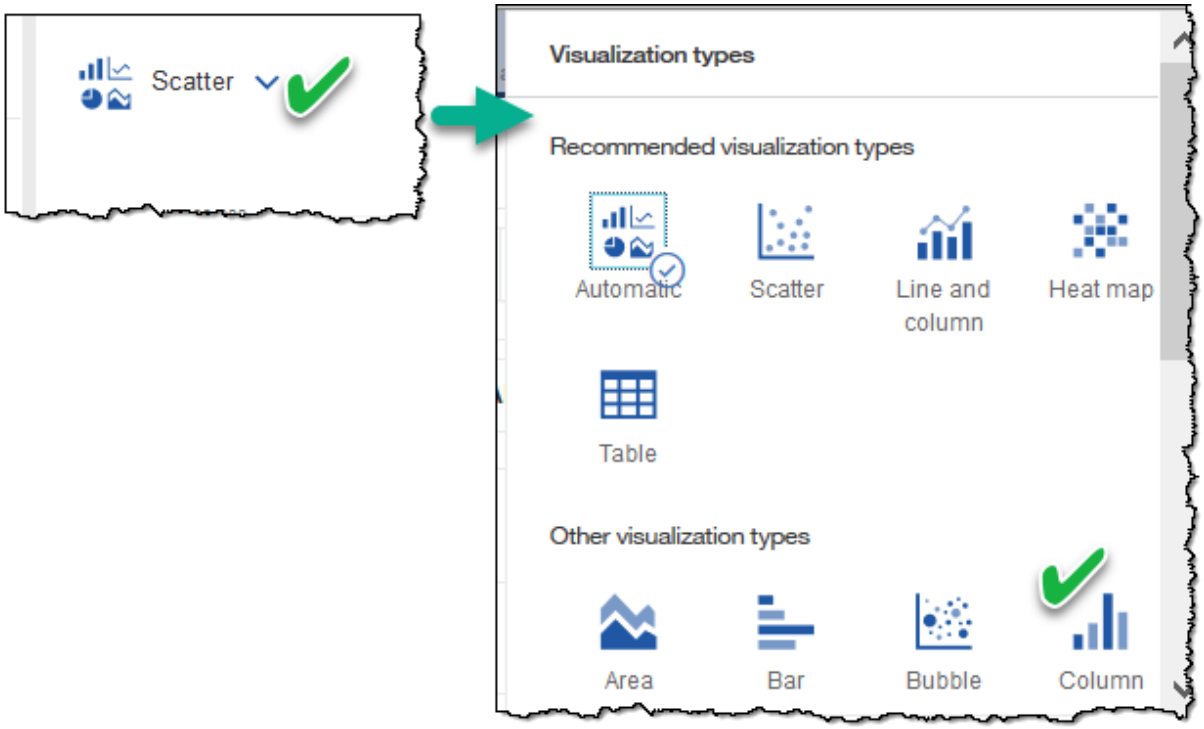
To see this, hold the control key down and click DAYSSINCELASTTRADE and TOTALDOLLARVALUETRADED and drag them both to the right.



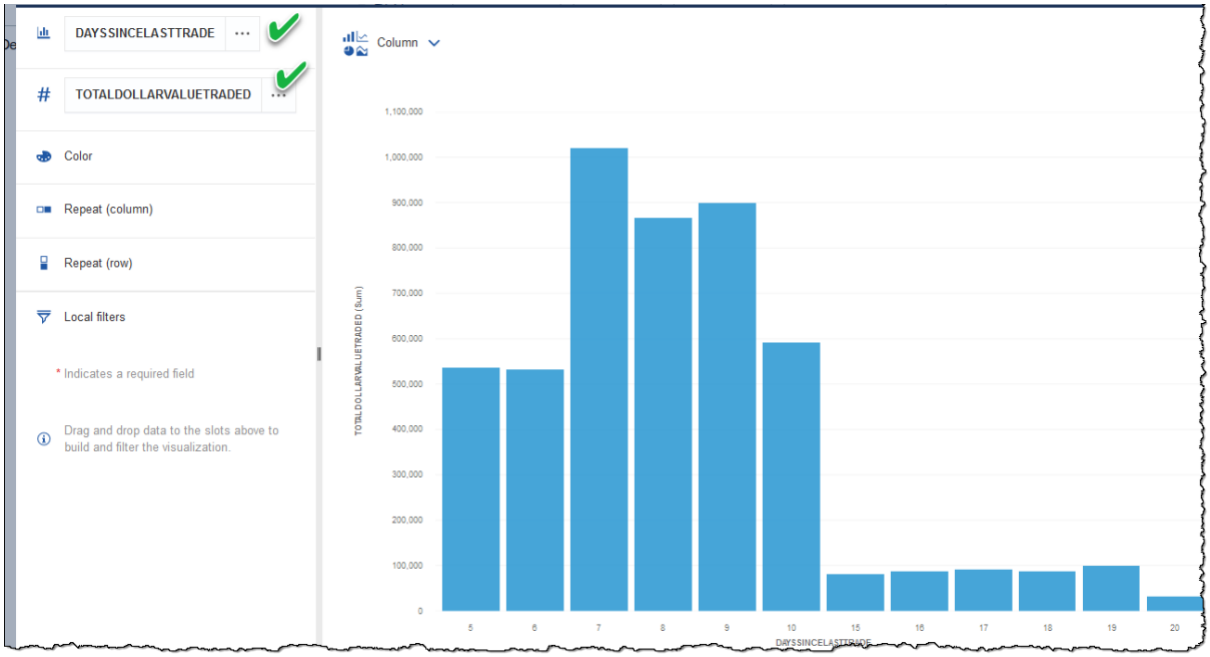
__262. Expand the visualization that was created.



__263. Change the chart type to Column. [Click and select from the chart visualization.]



__264. The visualization should look like this:

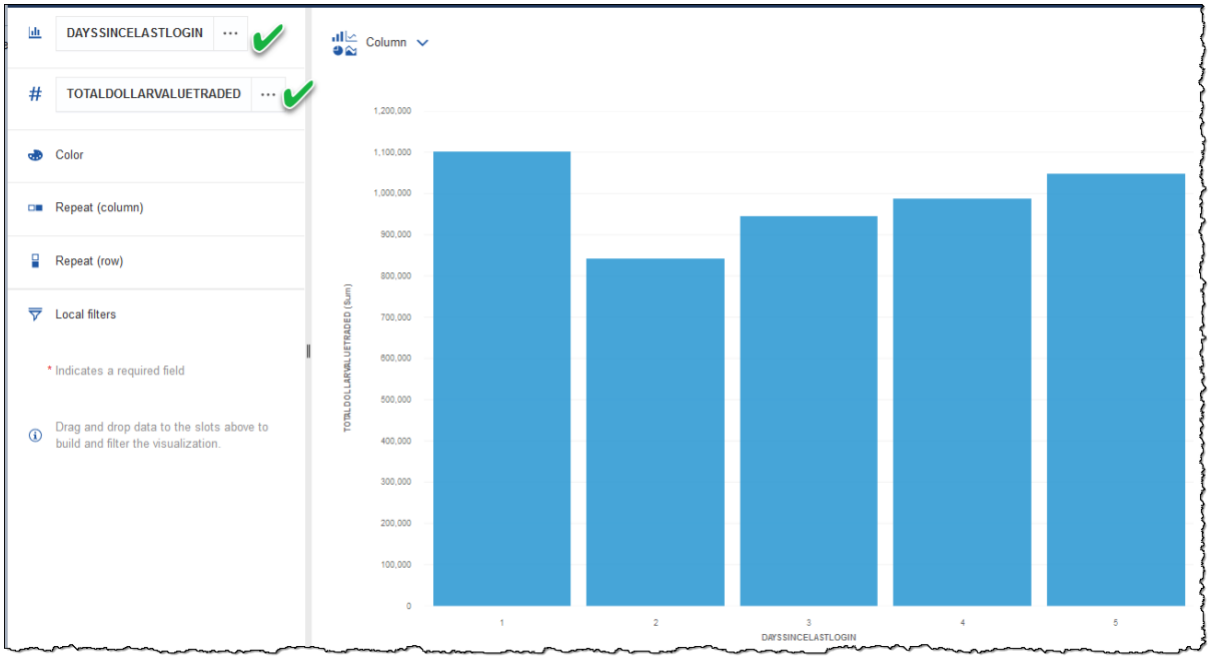


__265. Collapse the screen by clicking the collapse icon to the top right of the frame.

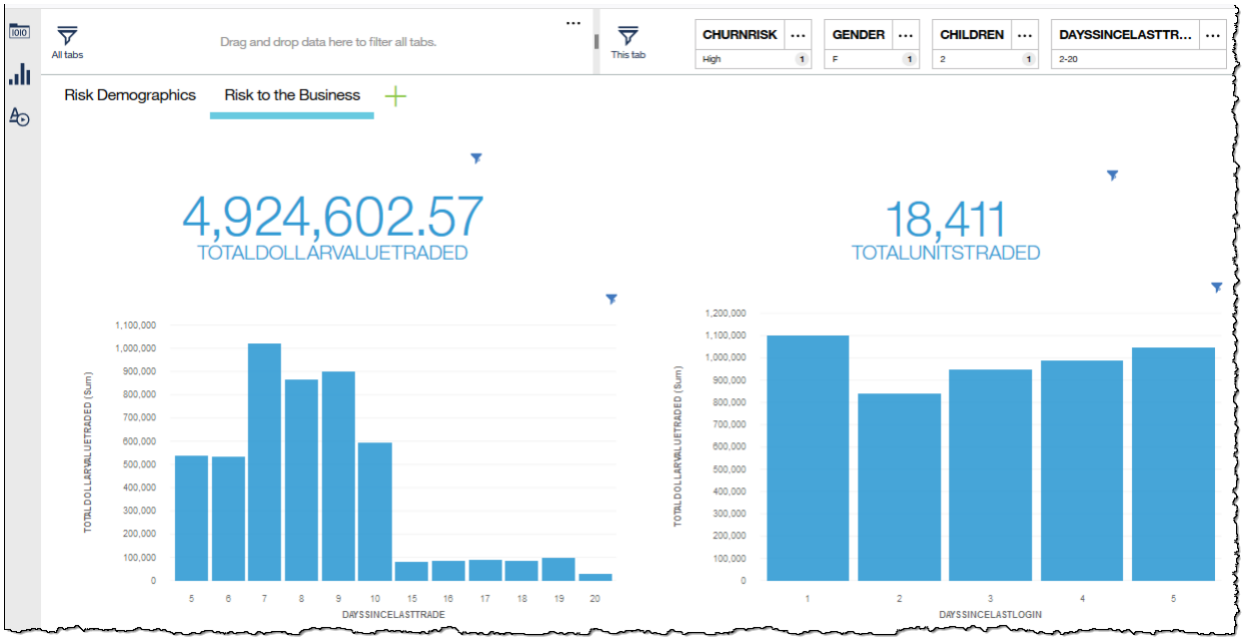
Resize this chart and fit into the bottom left of the work area. Save your work.

__266. Hold control key down and click `DAYSSINCEDLASTLOGIN` and `TOTALDOLLARVALUETRADED` and drag to the right.

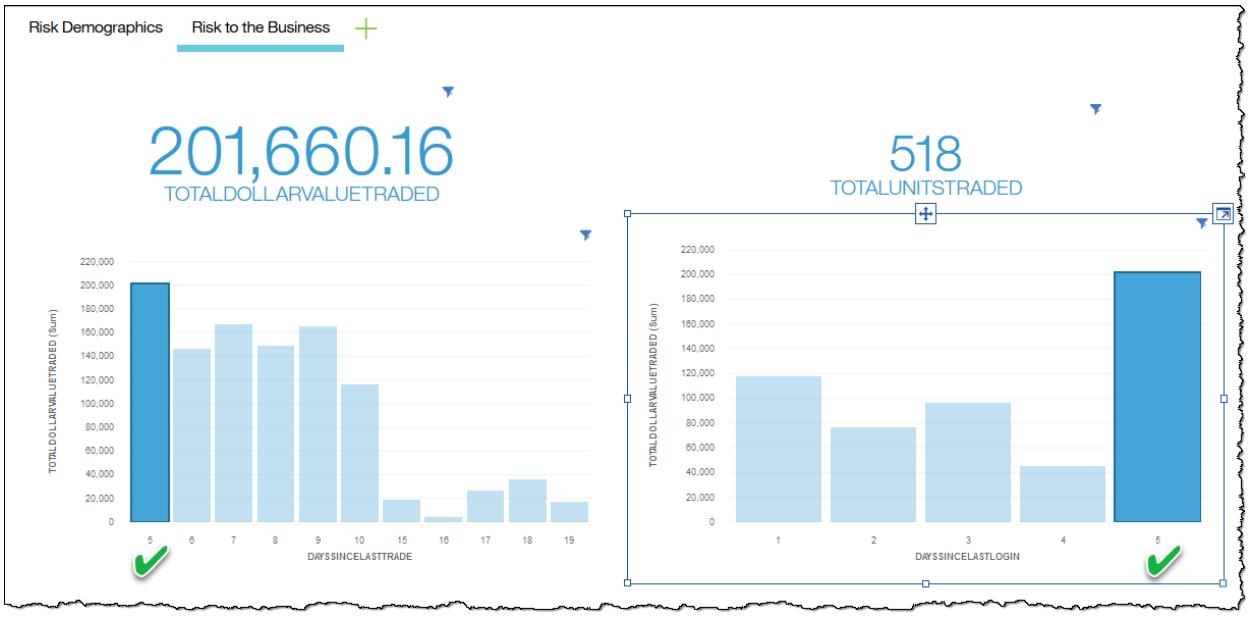
__267. Expand the chart and change the chart type to `Column.` It should display the following:



__268. Collapse this last visualization resize it and move it to make the dashboard look like this:



__269. You can click any column from each chart to show the aggregate value for total dollar traded and total units traded. For example: Click first column on first chart and last column on second chart to see the values.



Business Analyst	Playing the role of a business analyst and using these techniques, you have identified the group likely to be at risk to separate from the business. An intelligent scientific solution needs to be crafted to mitigate the risks. The data scientist personas are now engaged to move this analysis forward.

6.4 Lab conclusion

Dashboard creation and analysis is just one part of an Analyze phase, but a business analyst can do this relatively quickly and simply so that they can get information from your organization's data without relying on data scientists for everything.

However, the Data Scientist can take this information and create machine learning models from the knowledge gained by this kind of analysis.

In the next lab, you will explore how this is done.

**** End of Lab 06: Analyze Part 1 - Dashboards**

Lab 07 Analyze Part 2 – Model Creation

7.1 Lab overview

In the previous lab Analyze Part 1, the business analyst provided dashboards to provide insight into the data and the challenges in the organization is facing. A solution for these challenges now rests on the data scientist. These are the benefits that Cloud Pak for Data brings to this persona:

- Data search and preparation can consume up to 80% of a data scientist's time – CPD streamlines this and allows more time for the other necessary work.
- Data resides everywhere and in many formats, thus collecting all the data into one lake, or warehouse is not often realistic - CPD Data virtualization and Transformation makes this easy.
- Governing and cataloging the data gives the data scientist confidence that what they are using is secure and correct - again, CPD makes this easy.

7.2 Persona represented in this lab

The `Data Scientist` persona most likely one to perform the `Analyze part 2` tasks in this lab, that is, to create a machine learning model that can be consumed by the Stock Trader application.

Persona (Role)	Capabilities
 Data Scientist	Data Scientists brings expertise in statistics and the process of building ML/AI models to make predictions and answer key business questions.

7.3 Toolsets for analyzing data

The recent contributions of universities, research institutes and corporations to the open-source movement have led to the development of many innovative tools in data science. These include Jupyter and Zeppelin notebooks and the R and Python languages.

The default environment for Cloud Pak for Data that you will be using in this lab is:

- Jupyter Notebook Server 5.7.0
- Anaconda3 5.2 with the conda-forge channel
- Python 3.6
- Apache Spark 2.3.2

You can additionally install one or more of following development environments as add-ons to CPD:

- Jupyter Notebook Server with Python 2.7 or Python 3.5
- RStudio Server with R3.4.3
- Zeppelin Notebook Server 0.7.3 with Anaconda2 4.4

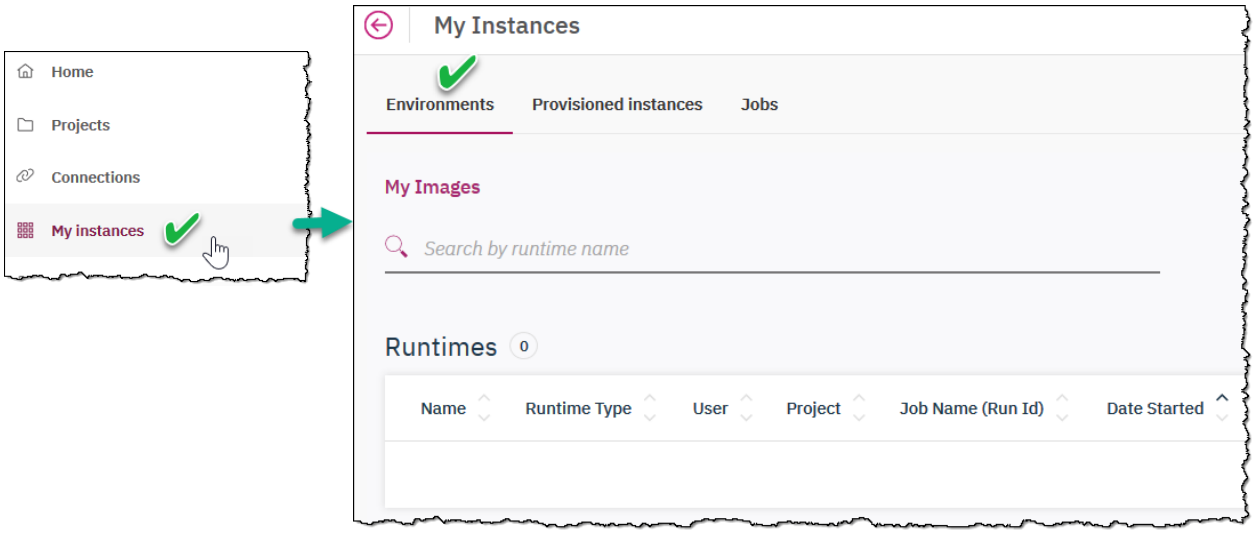
7.4 Walk through a model creation beginning steps

Developing, training, and deploying a machine learning lab from scratch is beyond the scope of this introduction to CPD, so you will walk through an existing Jupyter notebook that represents an example of the model development process. Those familiar with the process will recognize the tools and processes used in these exercises, but those who are not will get an introductory experience from it.

In this lab, you will develop, test, and score a model that addresses the business problem at hand.

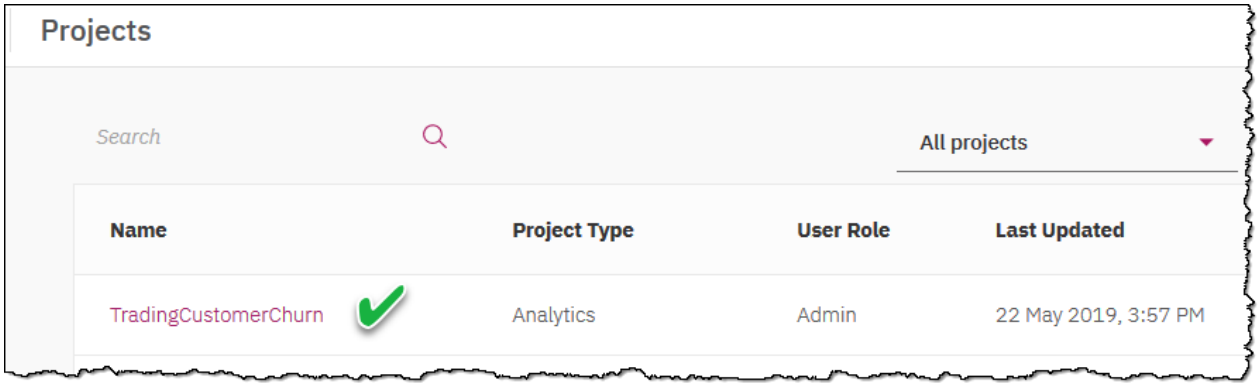
__270. Click `Navigation Menu` ⇨ `My Instances` ⇨ `Environments`

This location is where your provisioned Runtime environment will be able to be controlled. In the official workshop image, none should currently exist (this screen is blank) but if you were working in another environment you should check here first to review the Runtimes and potentially stop those that are running that are not being used. This is because each executing Runtime environment takes up compute resources on your CPD cluster.

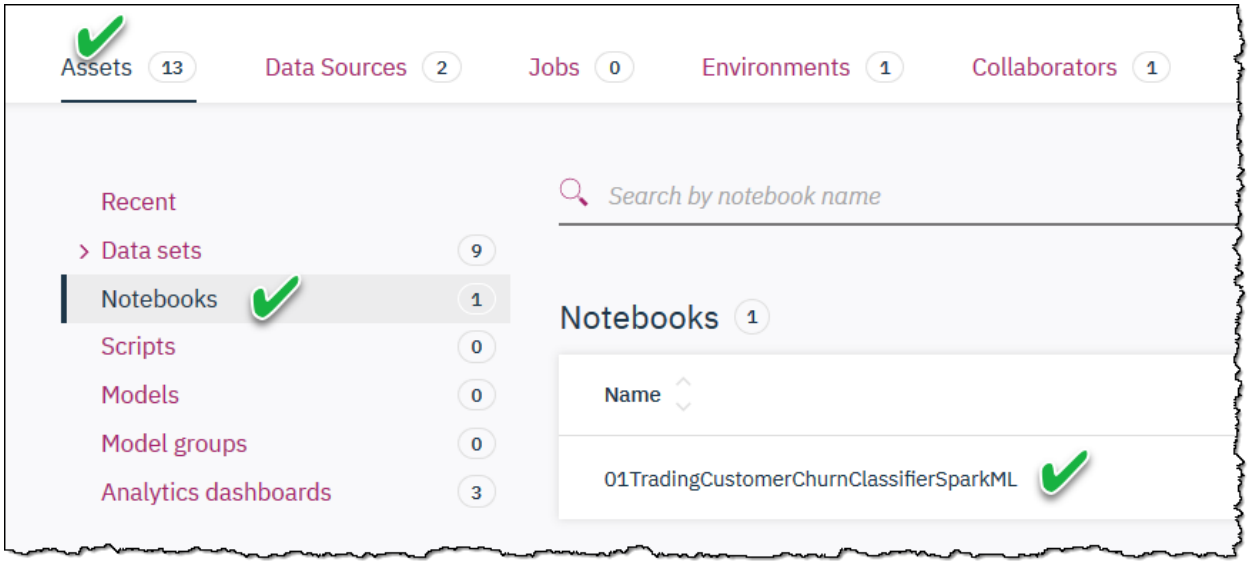


__271. From the `Navigation Menu` ⇨ `Projects`

__272. Click `TradingCustomerChurn` to open it.

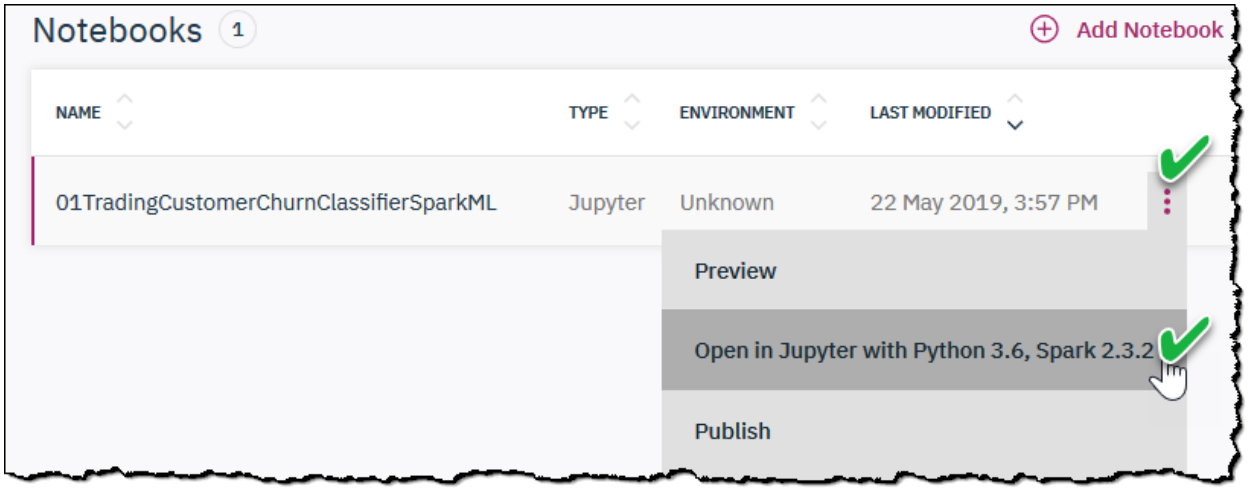


__273. Under `Assets` tab, click `Notebooks`



__274. Click the ellipsis next to Notebook `01TradingCustomerChurnClassifierSparkML`

Choose `Open in Jupyter with Python 3.6, Spark 2.3.2`

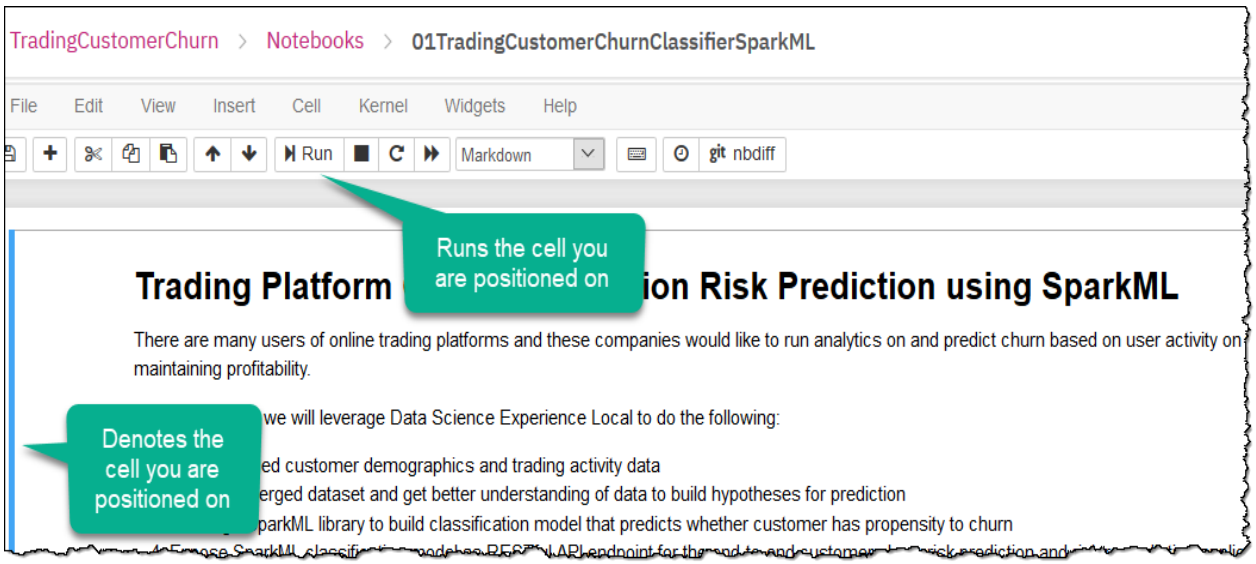


 Data Scientist	The Jupyter notebook process initiates a request to Kubernetes to launch a pod that will run on any node in the CPD cluster where sufficient resources are available. The launch may take a minute or so because it takes time to load the 7 GB image from the Docker registry and then launch it.

__275. If you see a message like this one, X out of it for now.

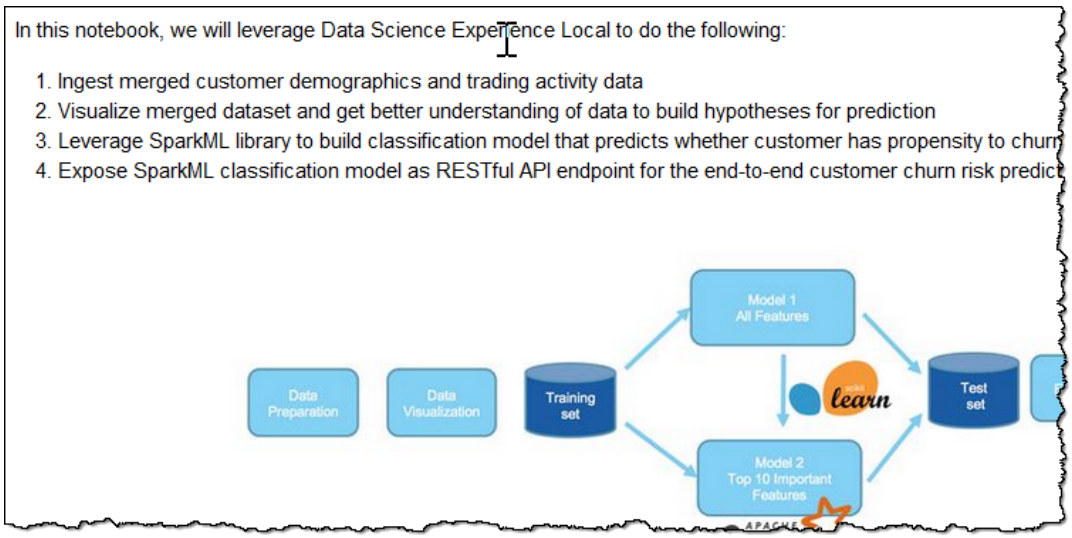
__276. A notebook is a collection of compute cells that can be run individually, or as a group. You can pause execution between steps to review its results, make changes to your code, and re-run the cell. One of the features of notebooks is that markdown cells (notes) can be interspersed with code which allows notebooks to be self-documenting.

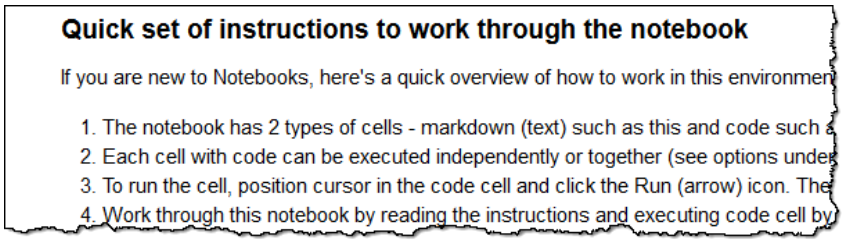
A notebook can use the environment of your choice. For example, we chose to use Python 3.6 with Spark 3.2.3 because it is the default for the current version of CPD. Others are available per the “toolsets” overview at the beginning of this lab.



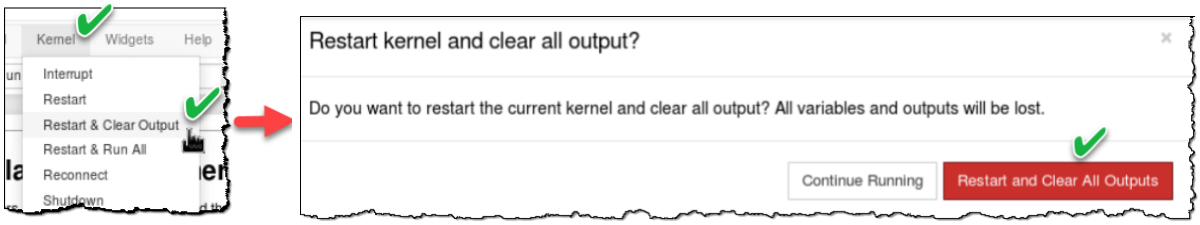
__277. Read through the introduction to learn what this notebook provides.

The notebook explains two main types of cells: `markdown` and `code`. Markdown cells are for documentation (the first cells in this notebook) and code cells that execute the Notebook processes to produce results.



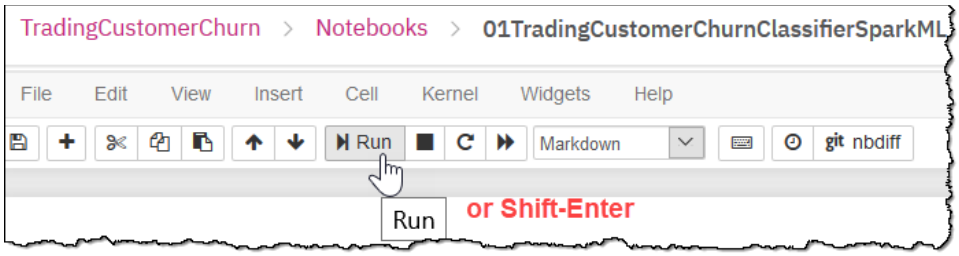


__278. Click `Kernel` ⇨ `Restart & Clear Output.` When prompted, click on the action again.



Wait for about 15 seconds after you do this for the kernel restart to take effect.

__279. Note: To execute a cells (including markdown cells) you can either click the menu icon `Run` or use keyboard `Shift-Enter` execute the cell you are positioned on.

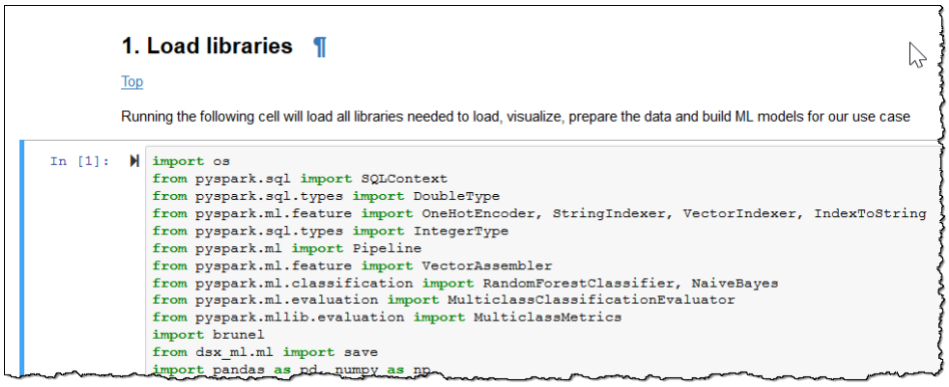


Click `Run` (or `Shift-Enter`) to execute the cell first cell you are positioned on. Note that executing a markdown cell essentially skips it because nothing happens to that cell.

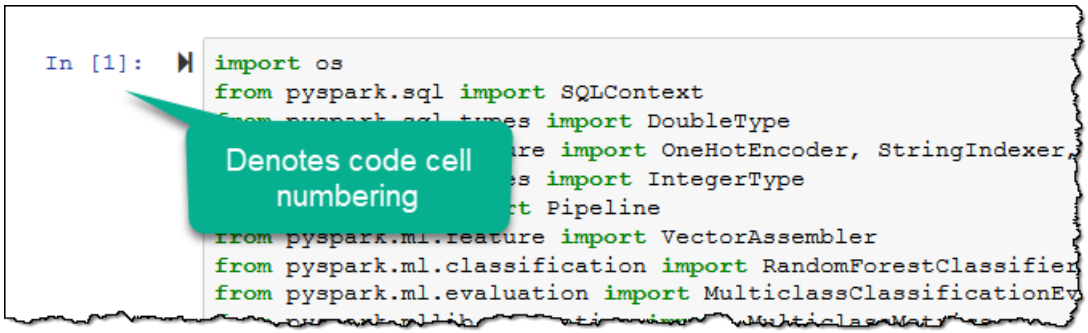
You can execute cell one at a time, in which case execution pauses and positions at the next cell.

You could also "Run All" cells which causes a serial execution from cell to cell without pause. (Do not do this because you will want to review each cell as you go through this Notebook.)

__280. Execute the first few cells in this Notebook until you reach this first code cell. Run it to load the libraries you will need for the rest of code cells in this Notebook.

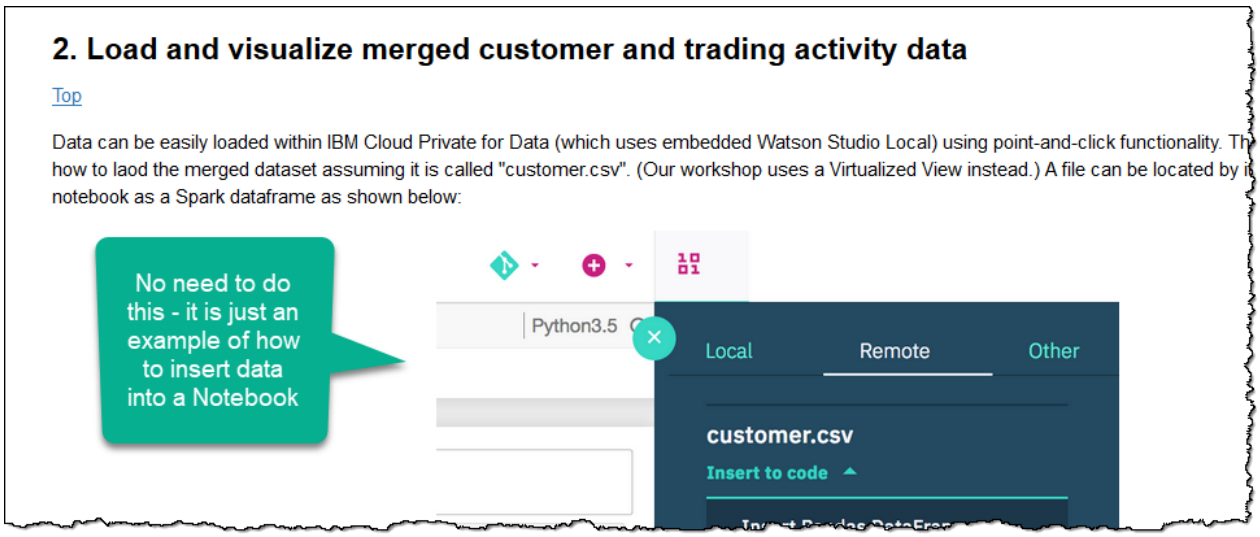


Notice that when a code cell is executed, the Notebook automatically numbers them each time this occurs. This helps tell what order the cells have been executed in.



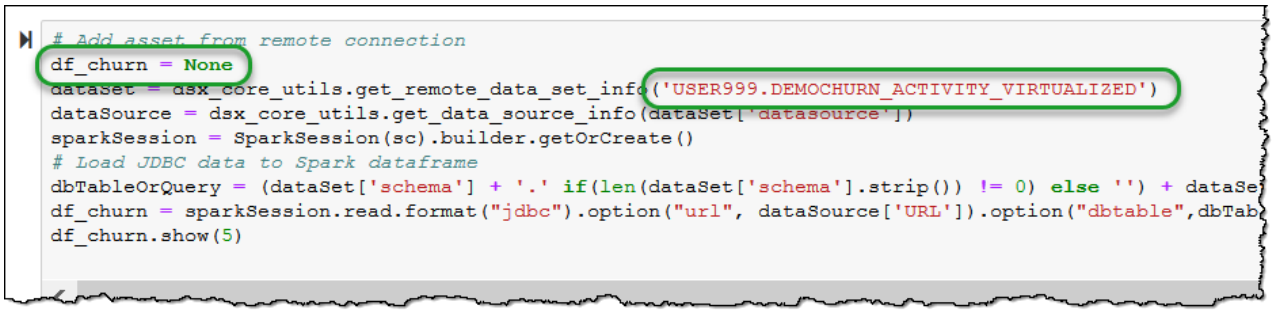
__281. The next markdown cell gives an example of the procedure to insert a Panda or Spark data frame. Ignore these instructions because we have already inserted the code to connect to the data.

Run this markdown cell to continue to the next code cell.

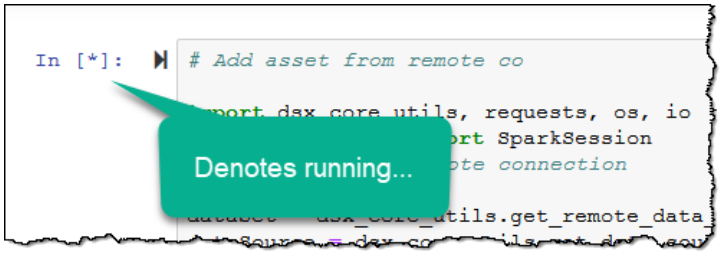


__282. The next cell is the second code cell and shows how a Notebook connects to a data source. It should look like the example below. This code has the key coding part circled that need to be correct for the rest of the notebook to run.

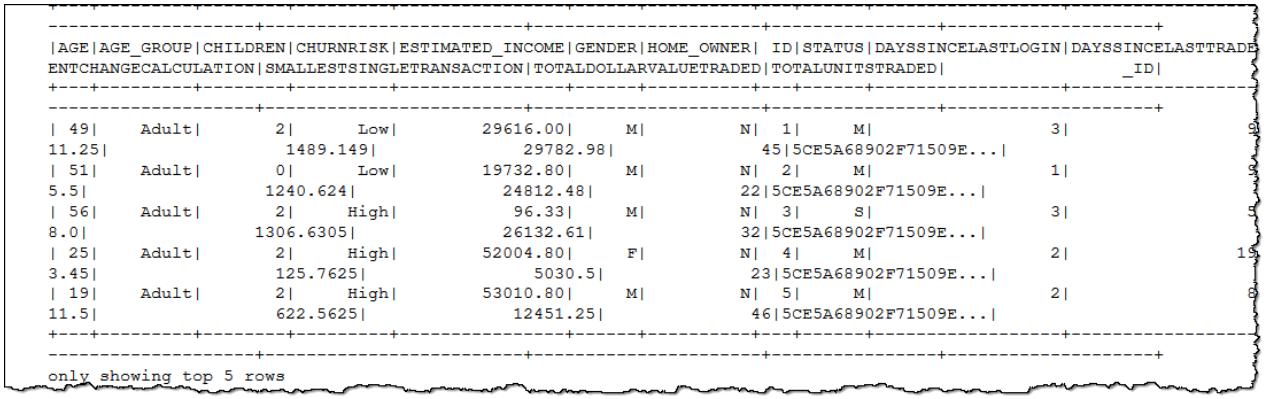
If the virtualized view you created is not the same name as the one shown here, change it in the cell before running it.



__283. Click `Run` (or `Shift-Enter`). While the cell is running, you will see the `[*]` indicator (meaning it is running.)



__284. The output will look like this which should be recognizable as a select of the first 5 rows from the virtualized view you created earlier: `USER999.DEMOCHURN_ACTIVITY_VIRTUALIZED.`

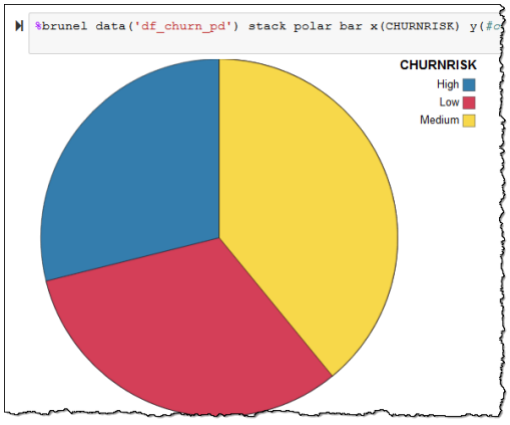


```
+---+---------+--------+---------+----------------+------+----------+---+------+-----------------+-----------------+
|AGE|AGE_GROUP|CHILDREN|CHURNRISK|ESTIMATED_INCOME|GENDER|HOME_OWNER| ID|STATUS|DAYSSINCELASTLOGIN|DAYSSINCELASTTRADE
ENTCHANGECALCULATION|SMALLESTSINGLETRANSACTION|TOTALDOLLARVALUETRADED|TOTALUNITSTRADED|                 _ID|
+---+---------+--------+---------+----------------+------+----------+---+------+-----------------+-----------------+
| 49|    Adult|       2|      Low|        29616.00|     M|         N|  1|     M|                3|                9
11.25|                1489.149|             29782.98|               45|5CE5A68902F71509E...|
| 51|    Adult|       0|      Low|        19732.80|     M|         N|  2|     M|                1|                3
5.5|               1240.624|               24812.48|               22|5CE5A68902F71509E...|
| 56|    Adult|       2|     High|           96.33|     M|         N|  3|     S|                3|                5
8.0|              1306.6305|               26132.61|               32|5CE5A68902F71509E...|
| 25|    Adult|       2|     High|        52004.80|     F|         N|  4|     M|                2|               15
3.45|               125.7625|                5030.5|               23|5CE5A68902F71509E...|
| 19|    Adult|       2|     High|        53010.80|     M|         N|  5|     M|                2|                8
11.5|               622.5625|               12451.25|               46|5CE5A68902F71509E...|
+---+---------+--------+---------+----------------+------+----------+---+------+-----------------+-----------------+
only showing top 5 rows
```

Data Scientist	You do not have to write the code to use data in a Notebook. The `Insert to code` capability shown in the previous markdown cell can do this for you.

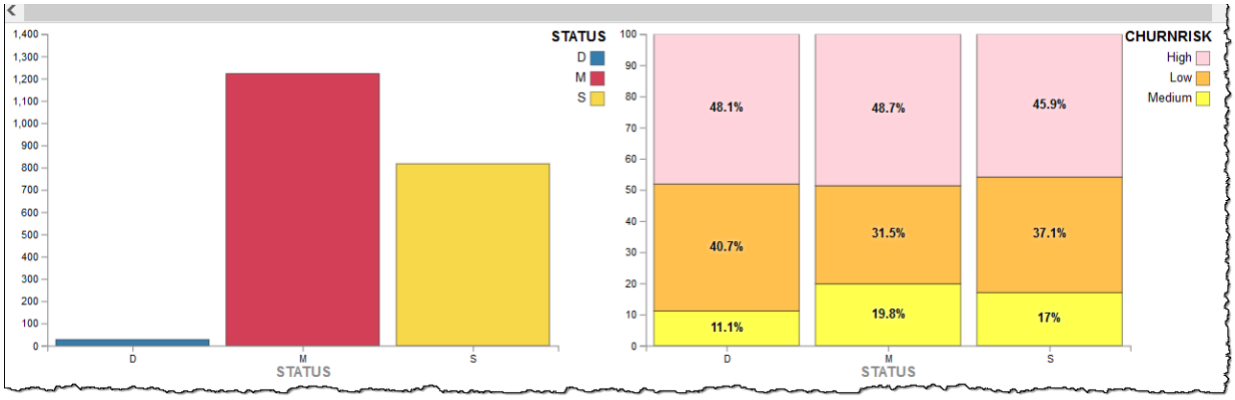
__285. The next cell contains code to visualize the data using the Brunel library.

`Run` the cell.

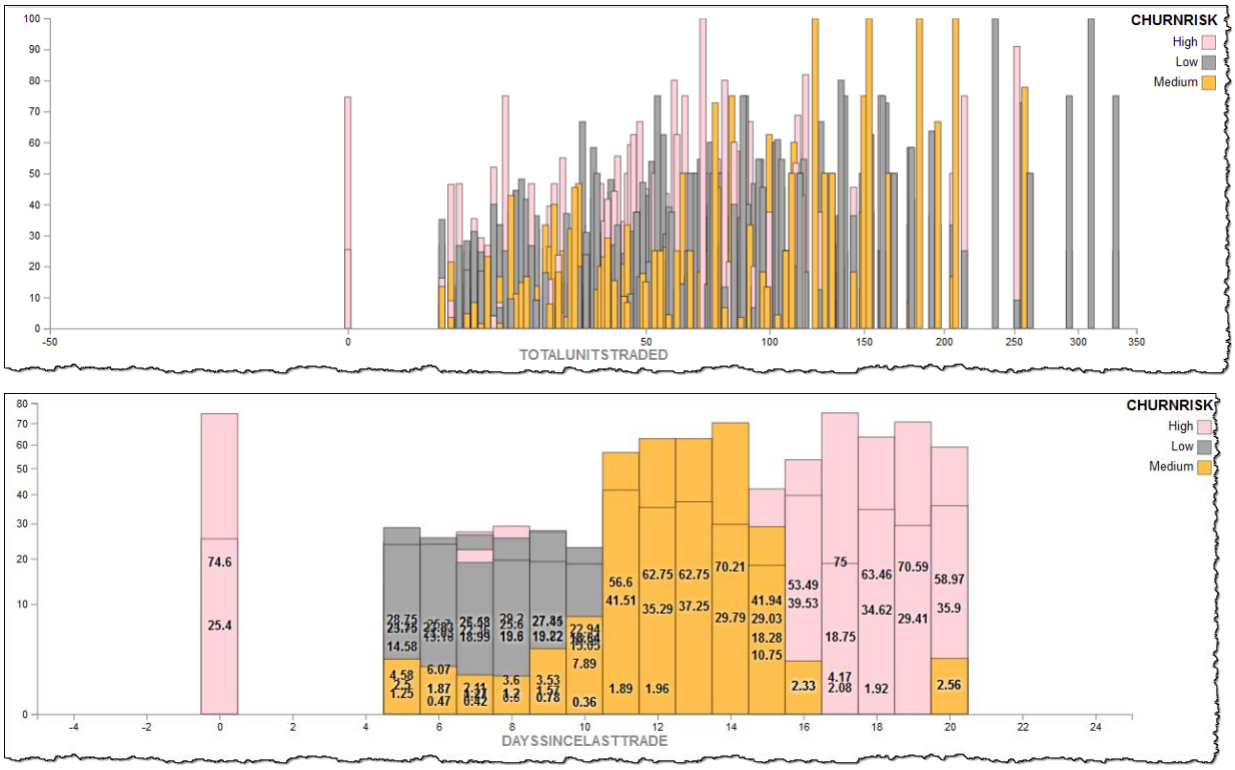


Brunel is just one of many open source visualization libraries available for Python and Jupyter. There are others like matplotlib, plotly, and Pixiedust.

__286. Run the next cell to show a visualization of CHURNRISK by different categories and status.

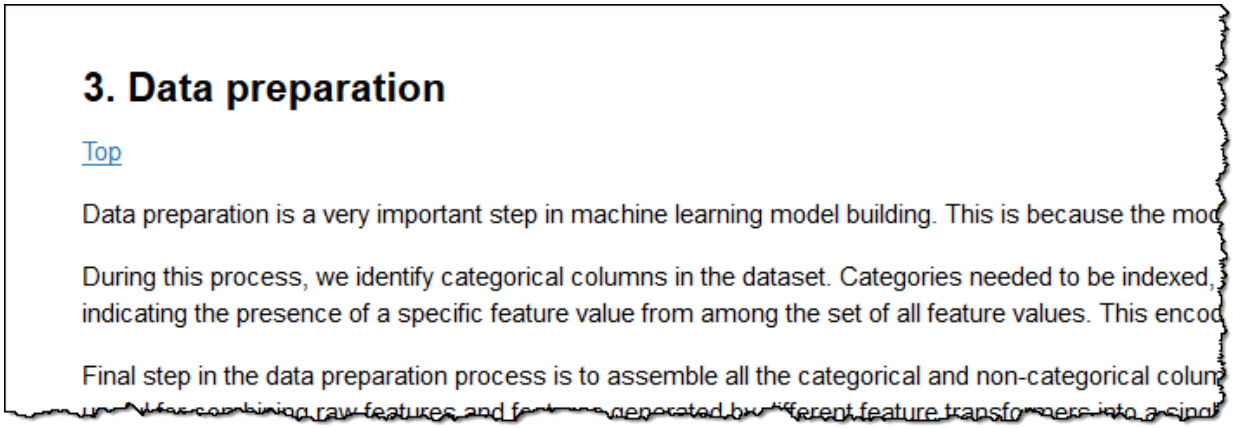


__287. Run the two next cells to see more visualizations breaking down CHURNRISK.

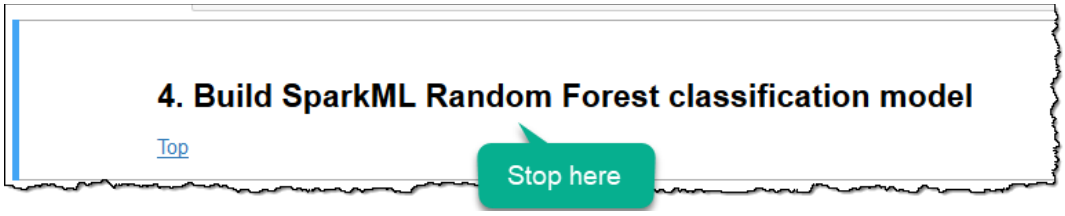


__288. Run the next cells to prepare the data for the model development.

Read through the first markdown cell in this series to get an understanding of what these next steps involve.

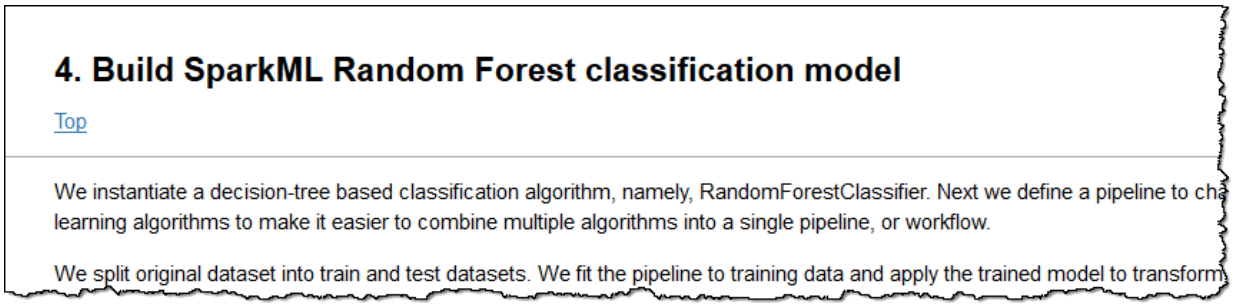


__289. Stop at the markdown cell that says `Build SparkML Random Forest classification model`



7.4.1 Build a Spark ML model using Random Forest Classification

__290. Read through the markdown cell to understand what you are doing next.



__291. Run the next two code cells that define the classifier and index labels and define the train and test datasets for a pipeline chain.

```
# instantiate a random forest classifier, take the default settings
rf=RandomForestClassifier(labelCol="label", featuresCol="features")

# Convert indexed labels back to original labels.
labelConverter = IndexToString(inputCol="prediction", outputCol="predictedLabel", labels=labelIndexer.label

stages += [labelIndexer, assembler, rf, labelConverter]

pipeline = Pipeline(stages = stages)
```

```
# Split data into train and test datasets
train, test = df_churn.randomSplit([0.7,0.3], seed=100)
train.cache()
test.cache()
```

```
]: DataFrame[AGE: int, AGE_GROUP: int, CHILDREN: int, CHURNRISK: string, ESTIMATED_INCOME: int, GENDER: string
STTRADE: int, LARGESTSINGLETRANSACTION: int, NETREALIZEDGAINS_YTD: int, NETREALIZEDLOSSES_YTD: int, PERCENT
, TOTALUNITSTRADED: int, _ID: int]
```

__292. Run the next few cells to build the model using the pipeline, then transform and show the results of the likelihood of churn for a particular customer (by ID).

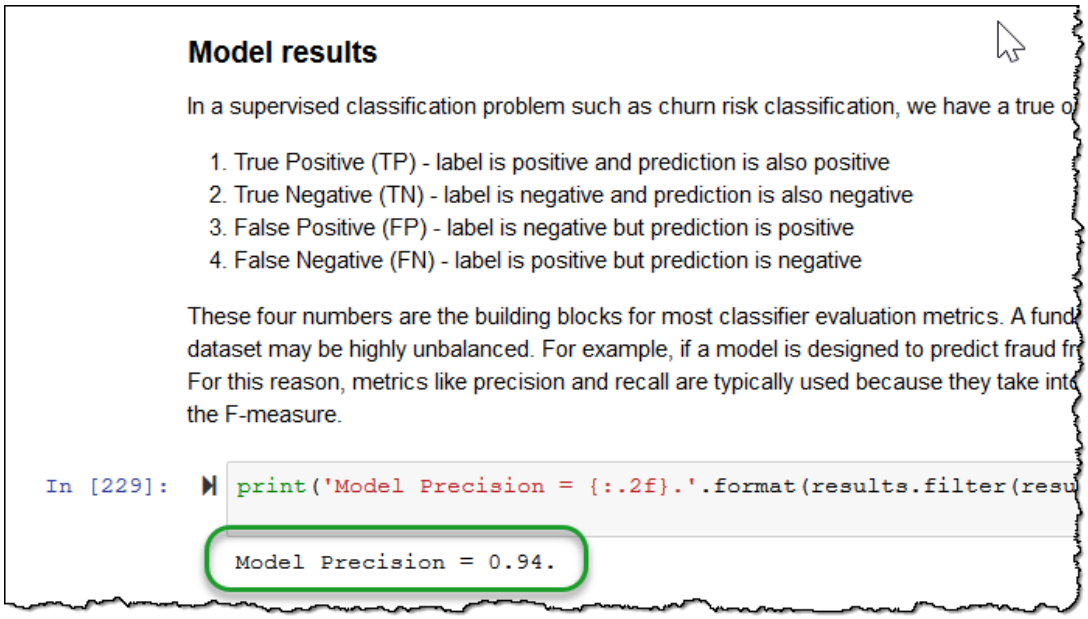
:

	ID	CHURNRISK	label	predictedLabel	prediction	probability
0	1044	Medium	2.0	Medium	2.0	[0.18256265483446013, 0.0, 0.8174373451655399]
1	1631	High	0.0	High	0.0	[0.961111111111111, 0.0, 0.03888888888888888]
2	1754	High	0.0	High	0.0	[0.961111111111111, 0.0, 0.03888888888888888]
3	1715	High	0.0	High	0.0	[0.8960250741768074, 0.021179039301310047, 0.0...
4	1310	Low	1.0	Low	1.0	[0.0, 0.9921115821527794, 0.007888417847220615]
5	1072	Low	1.0	Low	1.0	[0.02620087336244542, 0.9636144206985578, 0.01...

__293. Run the cells through the Model results markdown cell

__294. The “Model results” markdown cell describes in detail how a model like this is classified.

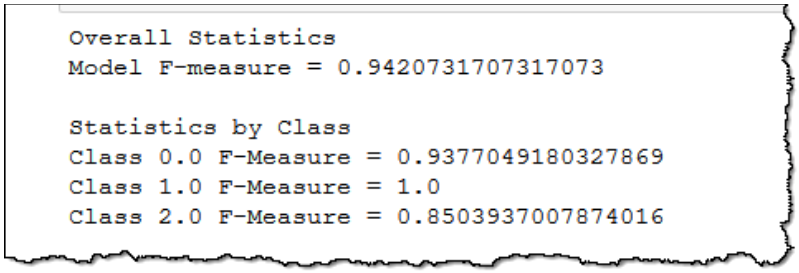
Run the next cell to see the Model Precision. (Note: your results may vary.)



7.4.2 Evaluate different models

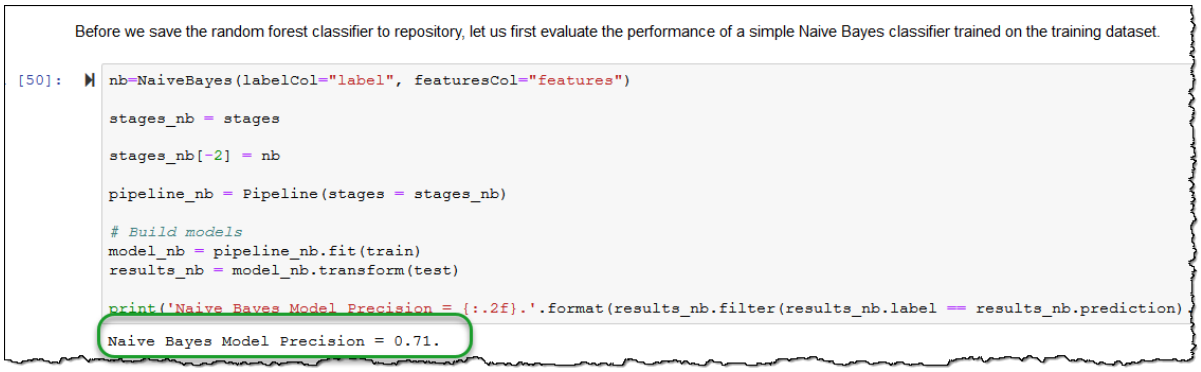
__295. The process of building model is incomplete unless it can be evaluated with a test data to see its accuracy.

Run until you see output similar to this:



__296. Next is Naïve Bayes classifier trained on the training data set. Notice this is not as good as the Random Forest classifier that shows a high F-measure.

Run until you see output similar to this:



7.4.3 Save the model in the repository

__297. After a comparison of two models, the Random Forest model is saved into the repository.

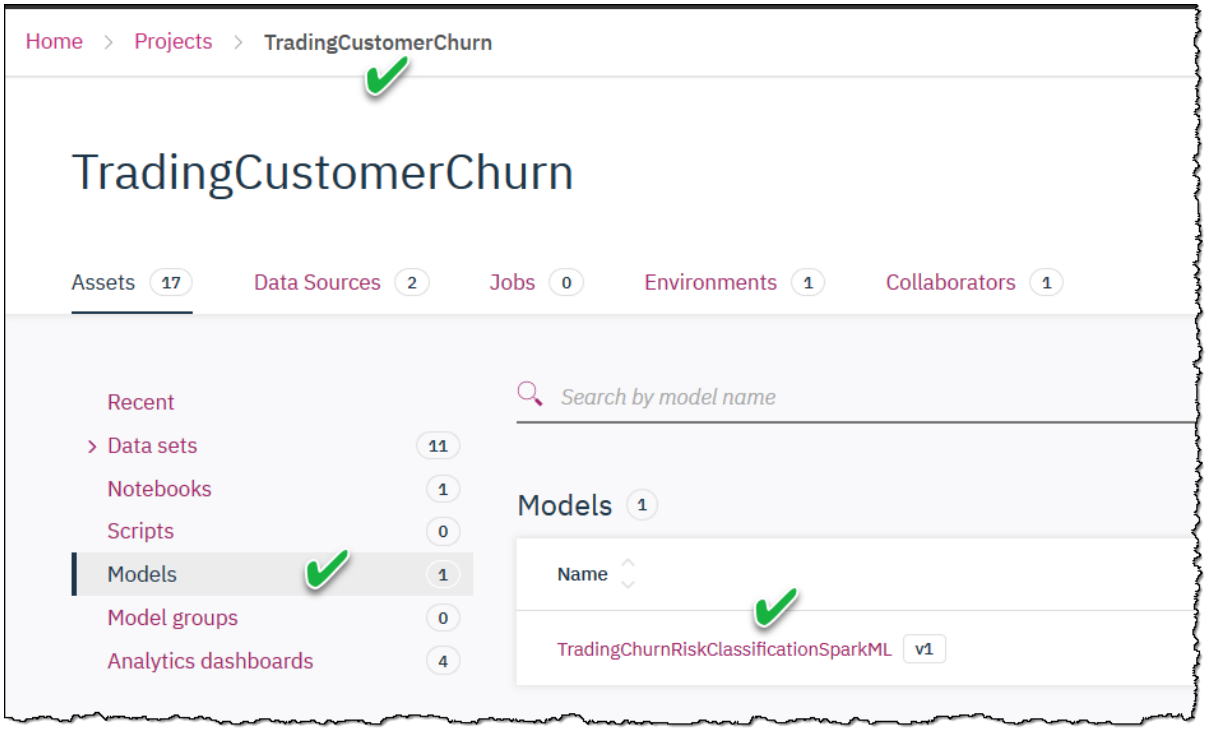
 Data Scientist	The model is saved in the CPD repository, which can provide governance to the data and the model itself.

7.5 Real-time and batch scoring

Now that the model is created, you will score and evaluate it.

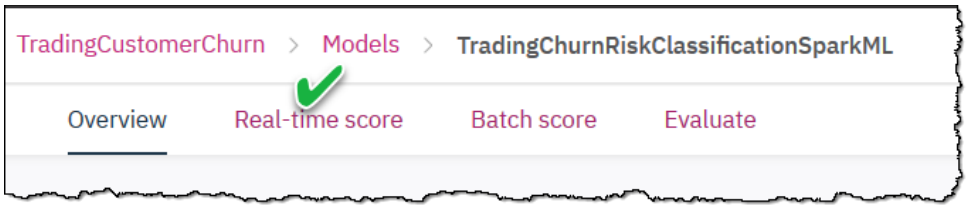
__298. Click project `TradingCustomerChurn` ⇨ `Models`

Click on the model to open it.

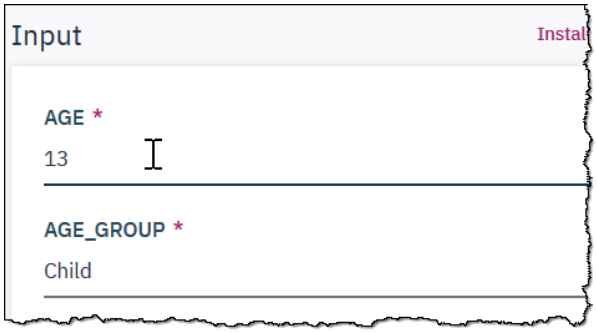


__299. The first test on this data will be a real-time score of the model using sample input values.

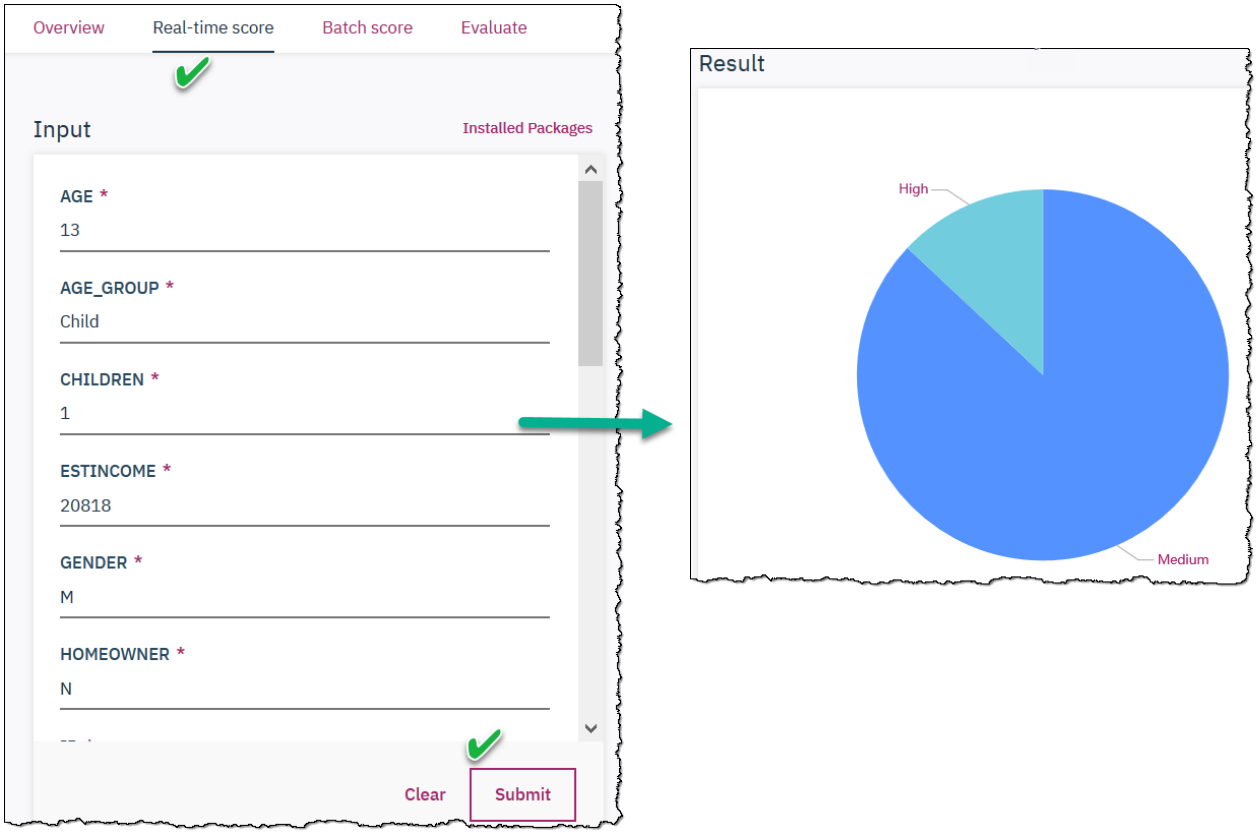
To do this, click `Real-time score`



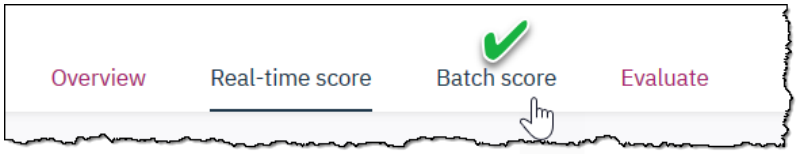
__300. Notice the sample input values can be changed - use what is provided by default for this first score.



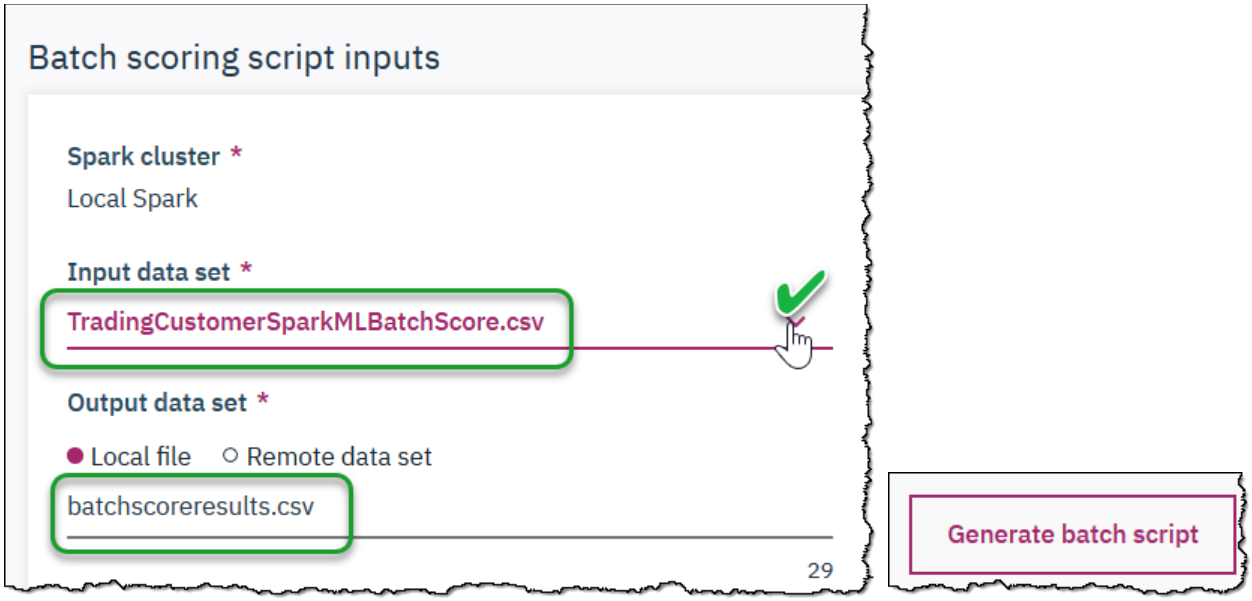
__301. Click `Submit`. The model Real-time scoring shows the percentage of High vs. Medium likelihood of churn for this particular set of input values. (Your results may vary if your default Input values are different.)



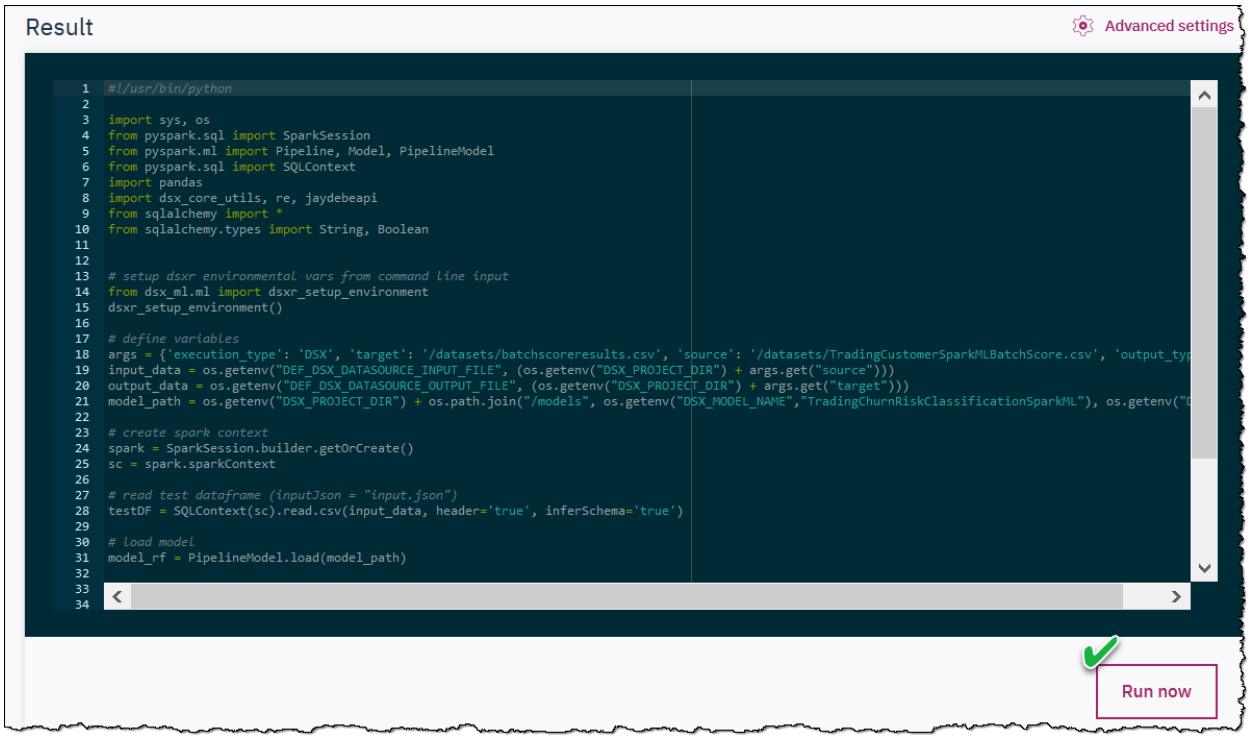
__302. Click `Batch Score`.



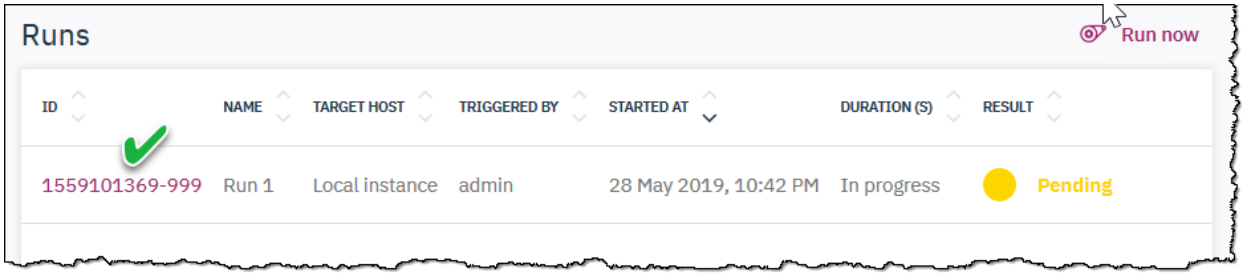
__303. Select Input data set: `TradingCustomerSparkMLBatchScore.csv` and type the output file name `batchscoreresults.csv`. Click `Generate batch script`.



__304. Click `Run now`.



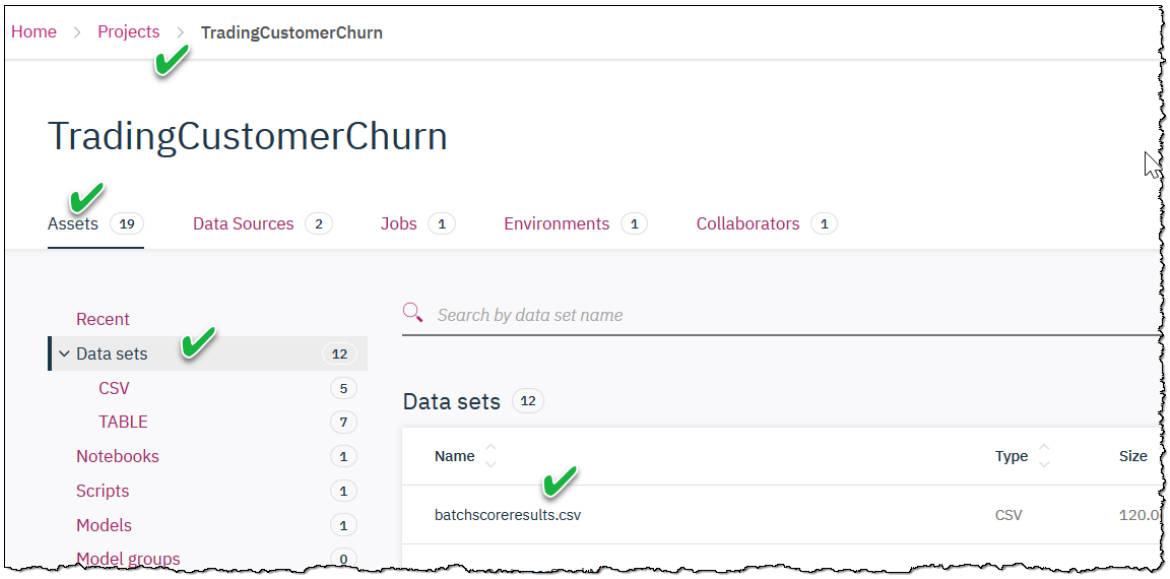
__305. Scroll down to `Runs` and click on the `ID`.



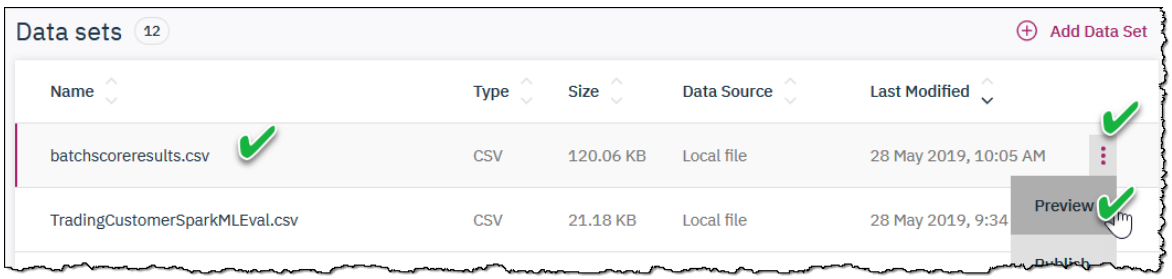
__306. When the `DURATION` has a number, it means the job is finished.

__307. Return to the Project `TradingCustomerChurn` ⇨ `Assets` ⇨ `Data sets`

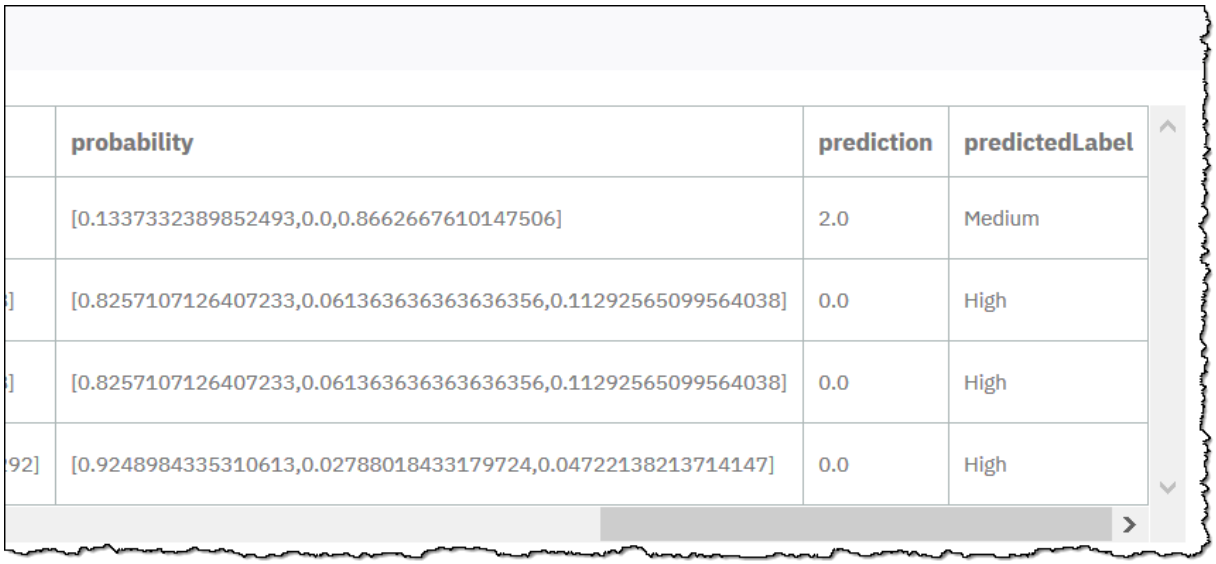
Find `batchscoreresults.csv`



__308. `Preview` this dataset



__309. Scroll to the right and check the prediction for the sample data for different input data combinations.



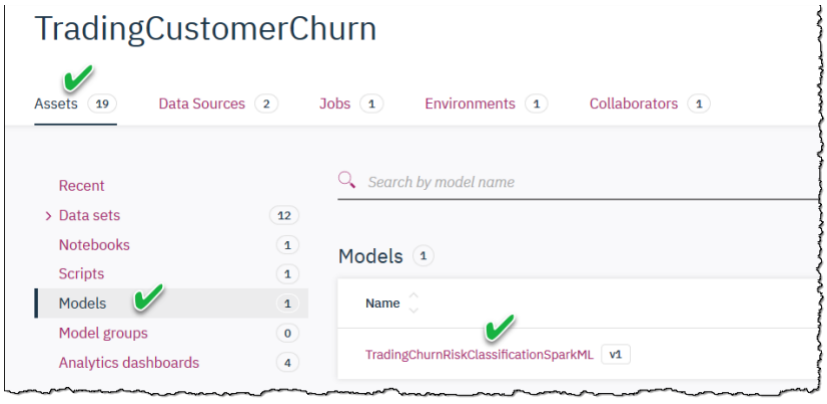
	probability	prediction	predictedLabel
	[0.1337332389852493,0.0,0.8662667610147506]	2.0	Medium
]	[0.8257107126407233,0.061363636363636356,0.11292565099564038]	0.0	High
]	[0.8257107126407233,0.061363636363636356,0.11292565099564038]	0.0	High
92]	[0.9248984335310613,0.02788018433179724,0.04722138213714147]	0.0	High

__310. Close the batch results preview

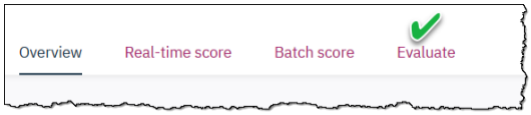
7.6 Create an Evaluation

__311. Return to the Project TradingCustomerChurn ⇨ Assets ⇨ Models

Open model TradingChurnRiskClassificationSparkML

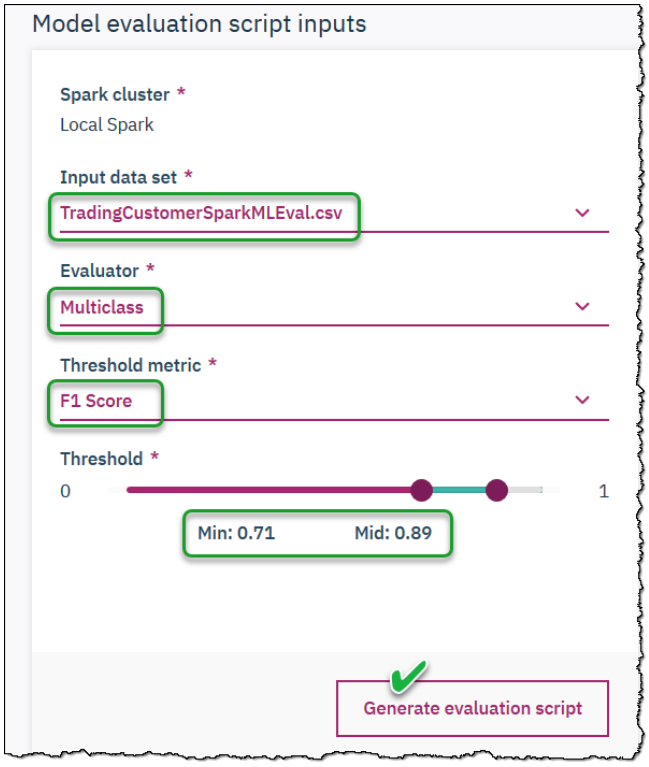


__312. Click `Evaluate`

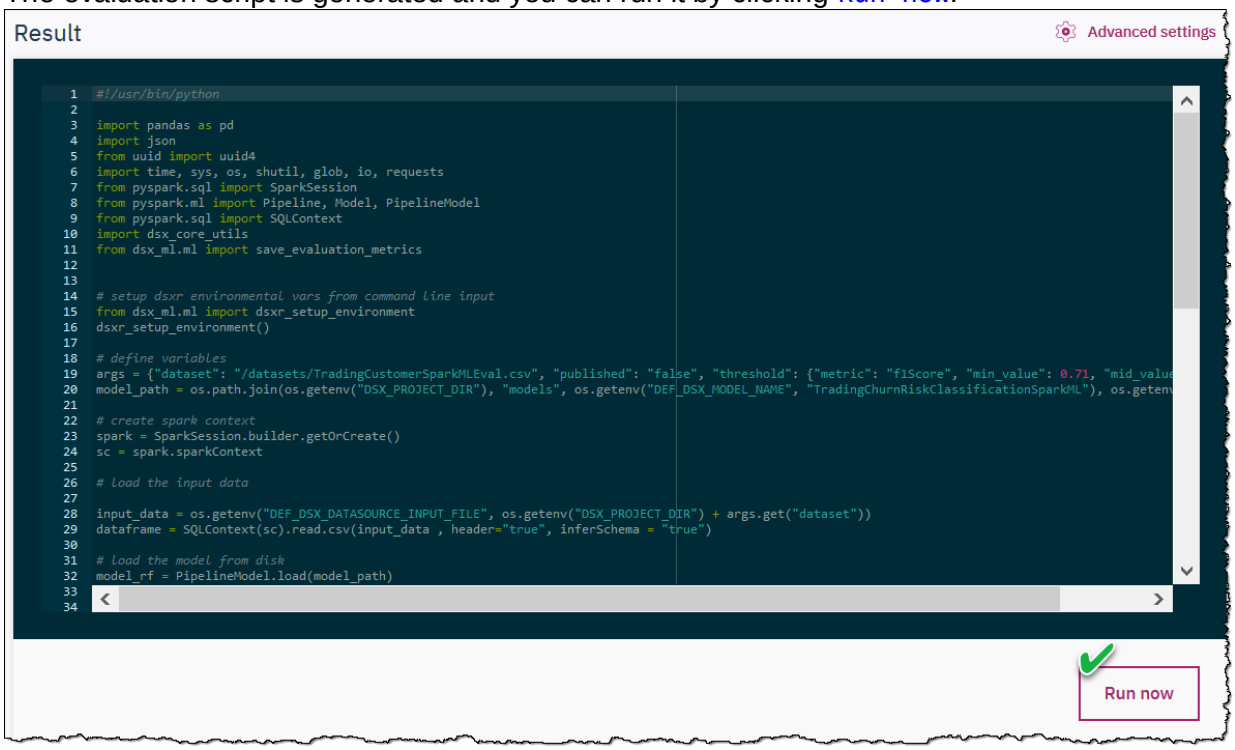


__313. Select from first drop-down Input data set `TradingCustomerSparkMLEval.csv`

Select Evaluator `Multiclass`
Select Threshold metric `F1 score`
Set Threshold criteria to be between `0.71` to `0.89`.
Click `Generate evaluation script`.



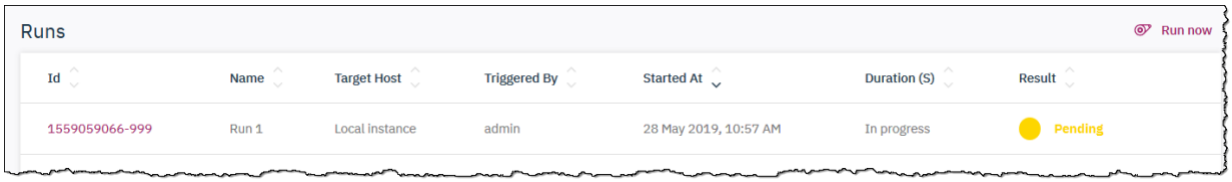
__314. The evaluation script is generated and you can run it by clicking Run now.



__315. The evaluation job is scheduled and at first is Pending ⇨ Running

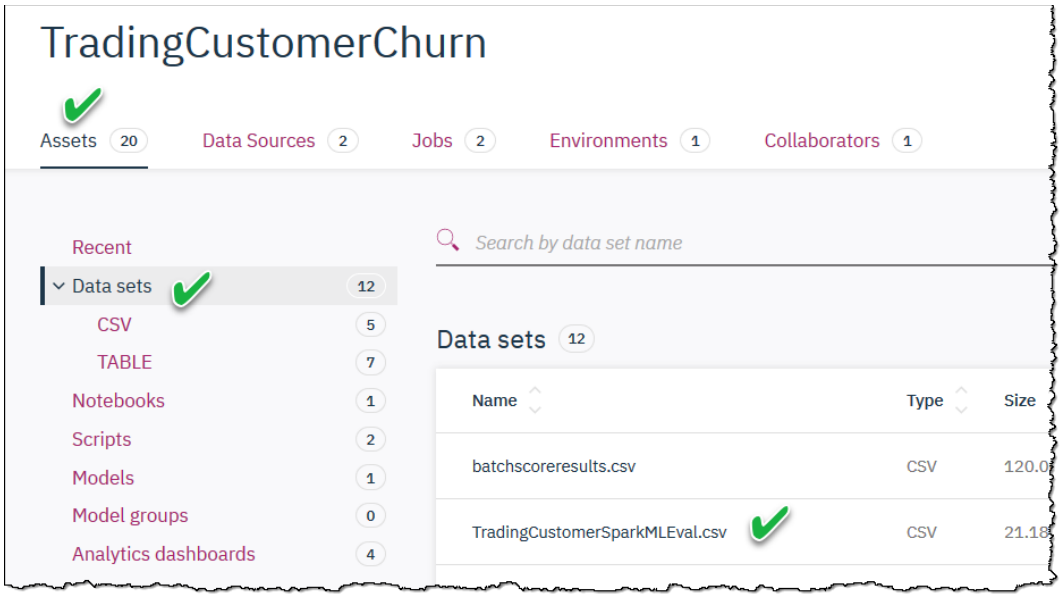
Click on the Id

DURATION shows the job completed

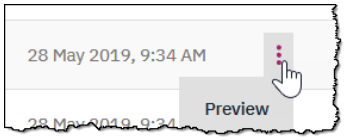


__316. Return to the Project `TradingCustomerChurn` ⇨ `Assets` ⇨ `Data sets`

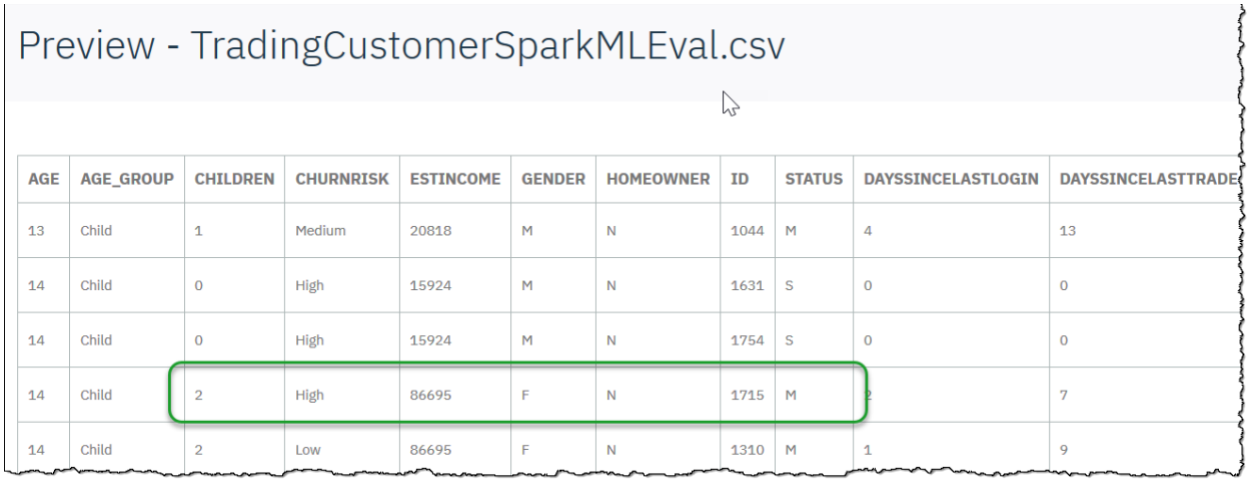
Find `TradingCustomerSparkMLEval`



⇨ `Ellipsis` ⇨ `Preview`



__317. Notice one of the high risk for churn candidates in this evaluation matches the same criteria of what the business analyst found in the heatmap of his dashboard.

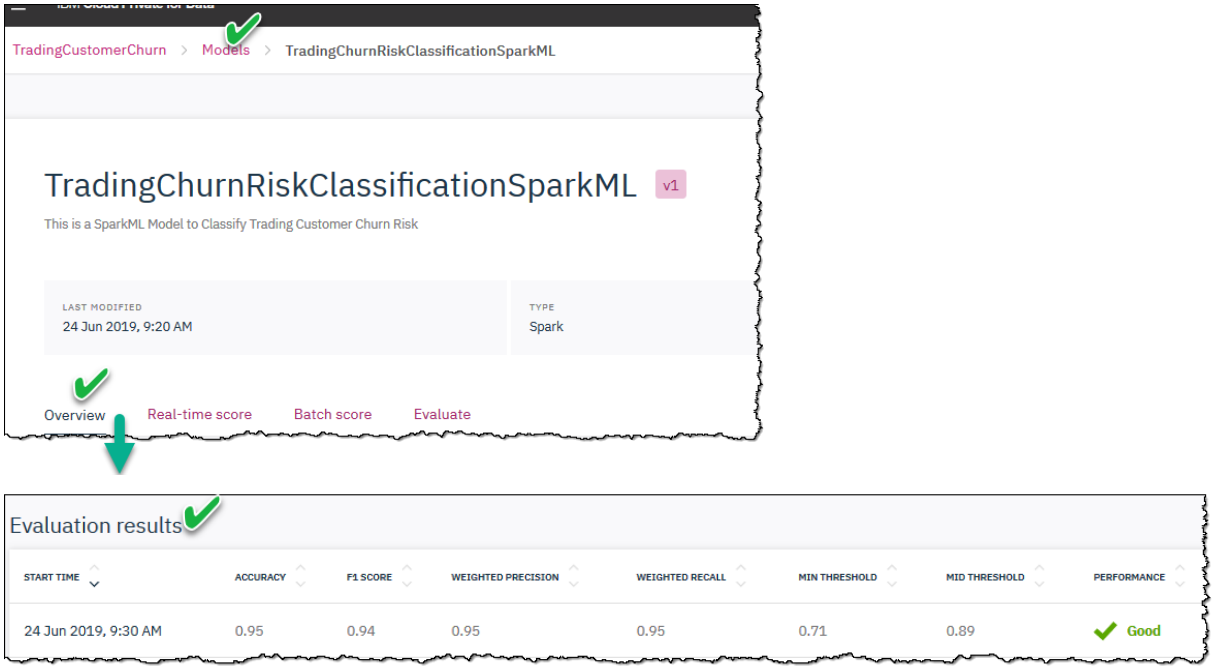


AGE	AGE_GROUP	CHILDREN	CHURNRISK	ESTINCOME	GENDER	HOMEOWNER	ID	STATUS	DAYSSINCELASTLOGIN	DAYSSINCELASTTRADE
13	Child	1	Medium	20818	M	N	1044	M	4	13
14	Child	0	High	15924	M	N	1631	S	0	0
14	Child	0	High	15924	M	N	1754	S	0	0
14	Child	2	High	86695	F	N	1715	M	2	7
14	Child	2	Low	86695	F	N	1310	M	1	9

Data Scientist

Models must be periodically tested, trained and redeployed. This is where you will see even more value of CPD as an integrated data platform.

__318. Check out results under `Models` ⇨ `Overview` ⇨ `Evaluation results`



7.7 Lab conclusion

You have seen how to create, score and evaluate models. But continued success for your organization will be by creating repeatable processes to prepare, test, evaluate and deploy of the models that have been infused into the applications.

The CPD platform provides the agility required for this.

The next lab exercise will show the deployment and the repeatable process that can fit into your CICD pipe line of DevOps.

**** End of Lab 07: Analyze Part 2 – Model Creation**

Lab 08 Deploy and Infuse

8.1 Lab overview

In our previous lab exercise, we created a model, scored and evaluated it using the CPD console. Business value is achieved when a successful model is deployed, infused into an application and subsequently updated. In this `Deploy and Infuse` lab you will learn how CPD assists you in this regard.

In the case of Boatswain Trading, it is their Stock Trader application that will benefit from this particular exercise.

 Developer	We are calling this lab `Deploy and Infuse` even though you will see some materials on Cloud Pak for Data refer to this stage as `Deploy`, while others call it `Infuse`. The thing is, since deployed models allow for the infusion of machine learning into applications, this stage can be described as both.

8.2 Persona represented in this lab

The `Developer` persona is the likely one to perform the various `Deploy and Infuse` tasks in this lab. However, the Data Scientist persona could perform the CPD specific tasks as well.

Persona (Role)	Capabilities
 Developer	Developers create and maintain the end-user applications that utilize the output from all the other personas on the CPD platform.

8.3 Deploy the model

Think of model deployment as the equivalent of writing a self-service application that takes the model and makes it available through a REST API interface. Application developers access and consume the model through the same interface. While this is a manual process in most organizations, Cloud Pak for Data can automate deploying and maintain models without writing a single line of code.

CPD eliminates the need for to do the following:

- Write code to perform the above capability and use a runtime to deploy it.
- Create a runtime on bare metal machines that require OS installation, network, storage, etc.
- Create a runtime on a virtual machine in a VMware on Intel, or IBM POWER VM® on a POWER platform.
- Create a runtime in Docker that requires someone to build the image and deploy it on one of the above platforms.

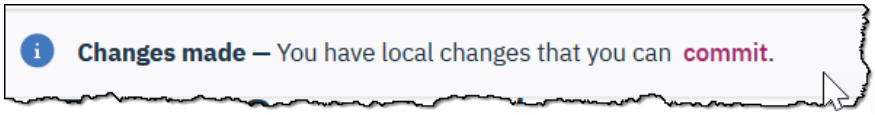
Each of the above requires manpower and machine resources. Using CPD, you can bypass this and quickly harvest insight from your data in a repetitive fashion by integrating it with your CICD pipeline.

8.3.1 Commit changes

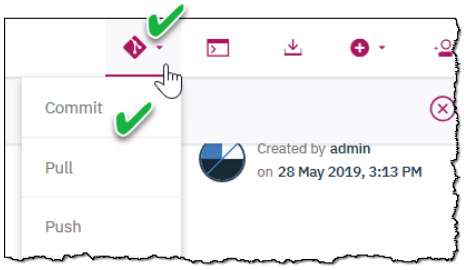
You will now examine the automated delivery and node deployment which you can integrate with a DevOps implementation.

__319. From the `Navigation menu` ⇨ `Projects` ⇨ `TradingCustomerChurn`

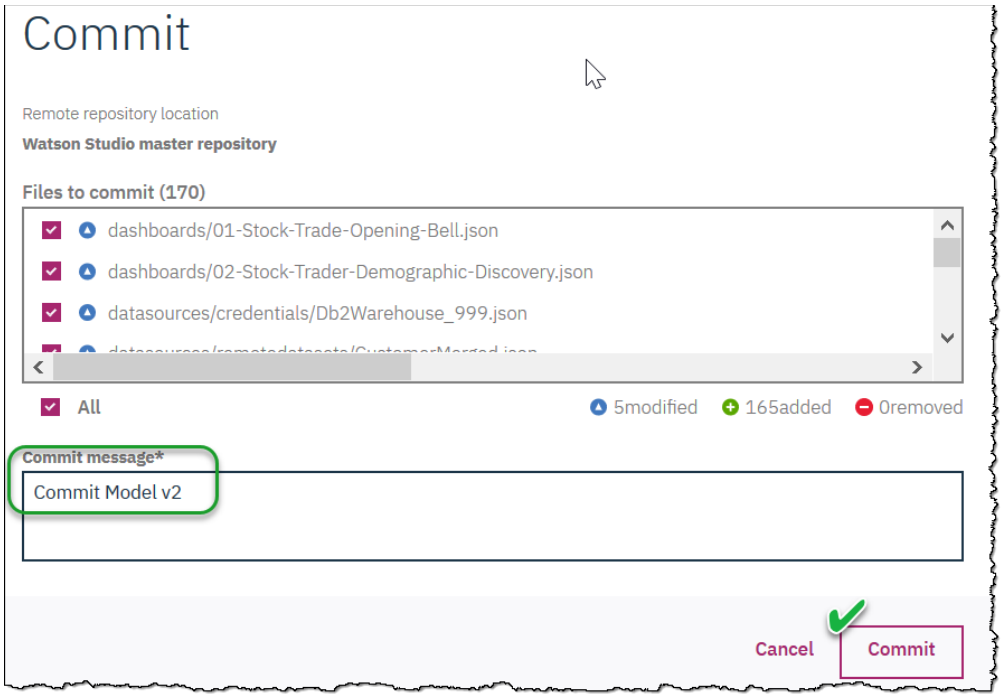
You should see this message because you have made changes to your project.



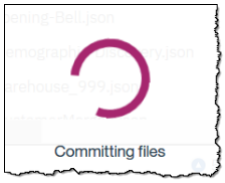
__320. Click `Git` icon and click `Commit`.



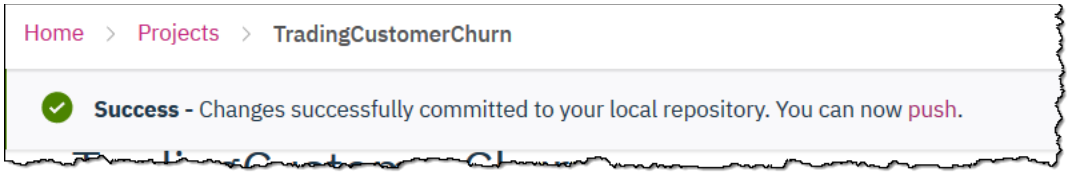
__321. Type: *Commit message* `Commit Model v2` ⇨ `Commit`.



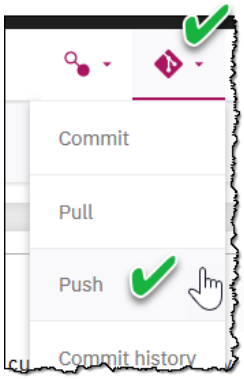
__322. Wait while the files are committed



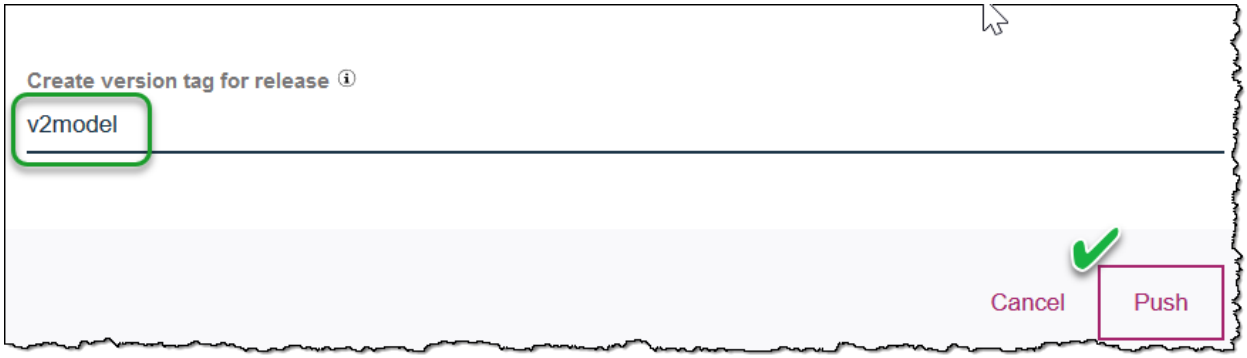
__323. When completed you will see the following message.



__324. Now `push` your changes.



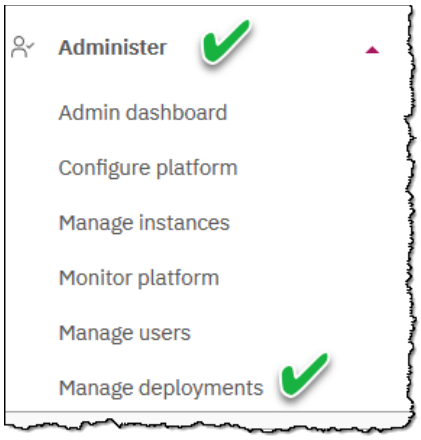
__325. Use tag `v2model` ⇨ `Push` (Note: If v2model is already there choose v3model)



__326. The changes have been pushed to the master repository. `X` out to delete this message.

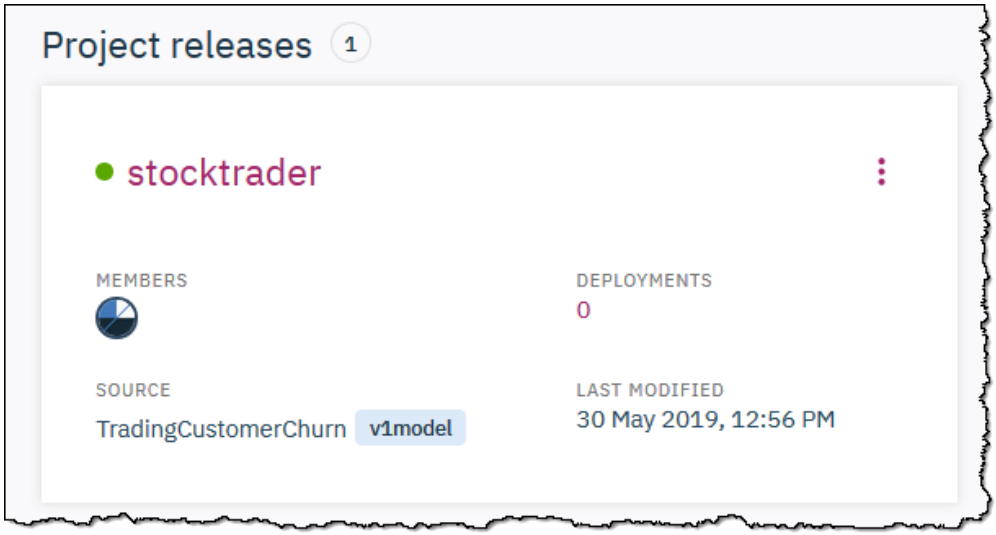
8.3.2 Model management and deployment

__327. Click `Navigation menu` ⇨ `Administer` ⇨ `Manage deployments`



__328. A `Project Release` called `stocktrader` has already been created. This was used to deploy the model used in the Executive Demo you ran through earlier.

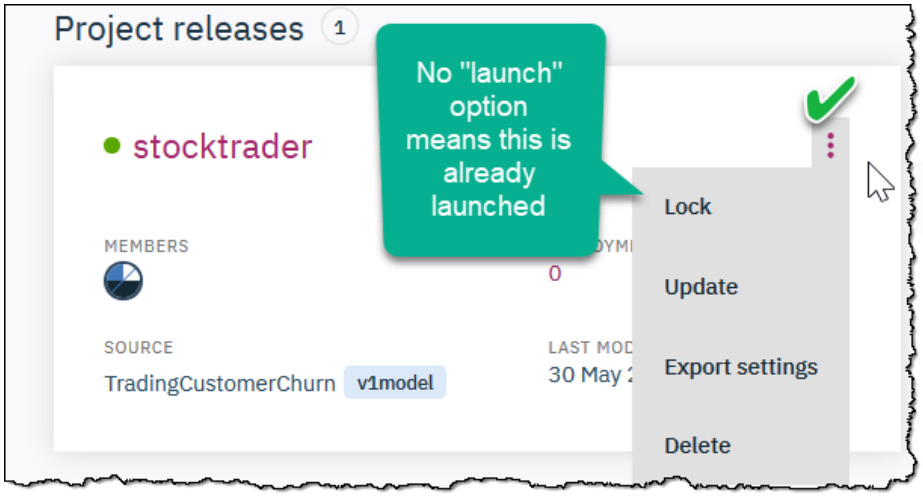
You will now use this same `Project Release` to deploy another version of the model, the one you created in the previous lab.



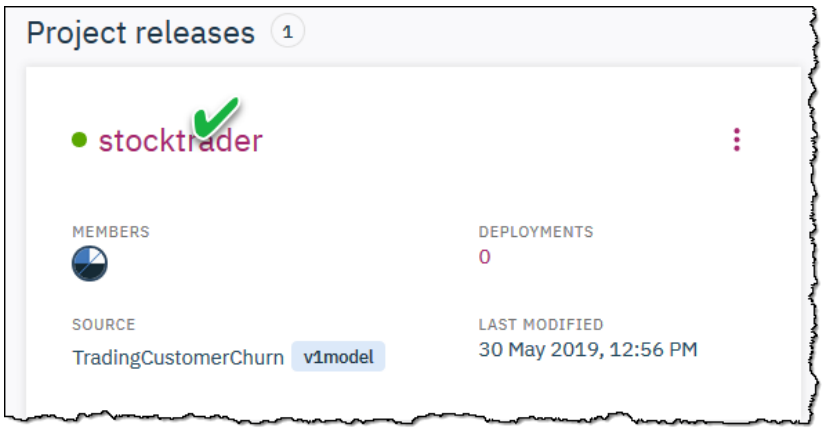
 Developer	Even though this was done for you, creating a Project Release is relatively simple: Click `+ Add Project Release` and fill in the screen with a Name, Route, Source Project, and Tag.

__329. Click the `Ellipsis` to view the options for this Project release.

If `Launch` is available, then `Launch` the Project release. (Note: it should already be launched and look like the image below.)

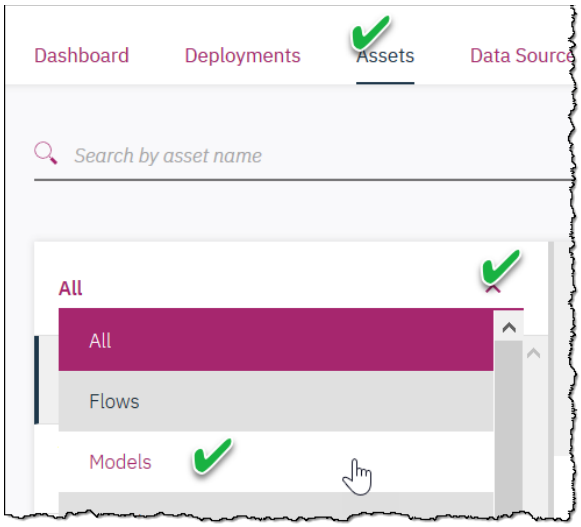


__330. In the Project release tile for `stocktrader`, click on the actual text "`stocktrader`" to open it.

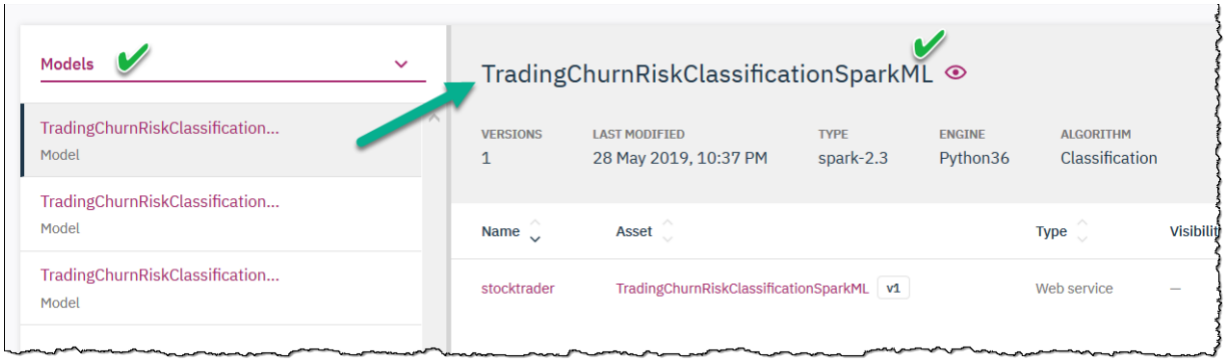


__331. Click tab `Assets`

Filter with drop down and choose `Models`



__332. Find the model with the name: `TradingChurnRiskClassificationSparkML`



__333. On the right of display of that model, click `+ web service`

__334. Name: `stocktrader` ⇨ Model version: `1` ⇨ Web service environment: `Python 3.6 – Script as a Service` ⇨ `Reserve CPU cores` ⇨ `Reserve GB Memory` ⇨ `Replicas:1` ⇨ `Create`

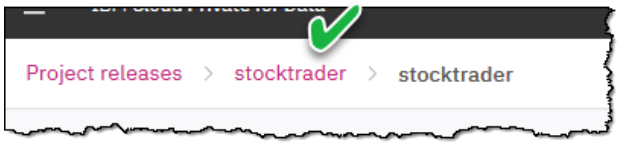
__335. The Kubernetes pod deployment is being created. This will take a few minutes to complete.

In the meantime, review the scoring `Endpoint` that can be embedded into your Stock Trader application. The Developer can easily click on the copy icon to paste it into his application code.

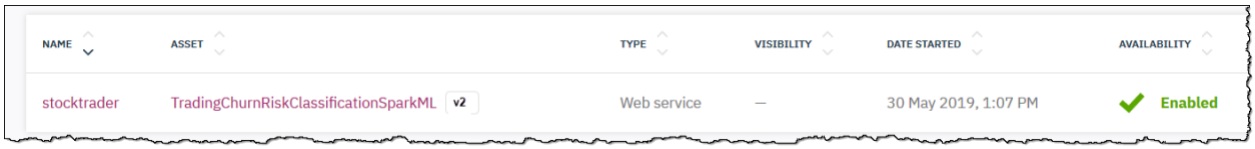
__336. In the same manner, a `Deployment Token` was created for the Developer to use.

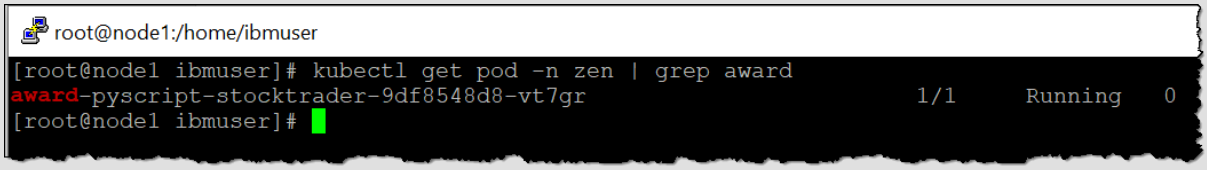
 Developer	The model deployment through CPD is secured with an authentication token. Any applications using the model need to authenticate themselves by providing this token.

__337. Click `stocktrader` from the path bar at the top of your screen to go back to that screen

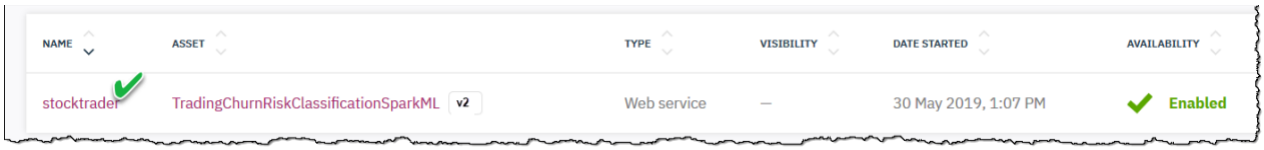


__338. Click tabs `Deployments`. It should be Enabled by this time. If is not, refresh your screen until it is enabled.



 Developer	The CPD platform builds a Docker container using the underlying private cloud platform. (In this case it is IBM Cloud Private.) The container and its Kubernetes pod are deployed onto the cluster and once deployed, they are ready to service the API requests to provide results back from the model. This is what the enabled (Running) pod for this deployment looks like on the Kubernetes cluster:  This is an example of the automation that the CPD platform provides in support of integrating data and application development pipelines.

__339. Click the deployment `stocktrader` to enter it again.

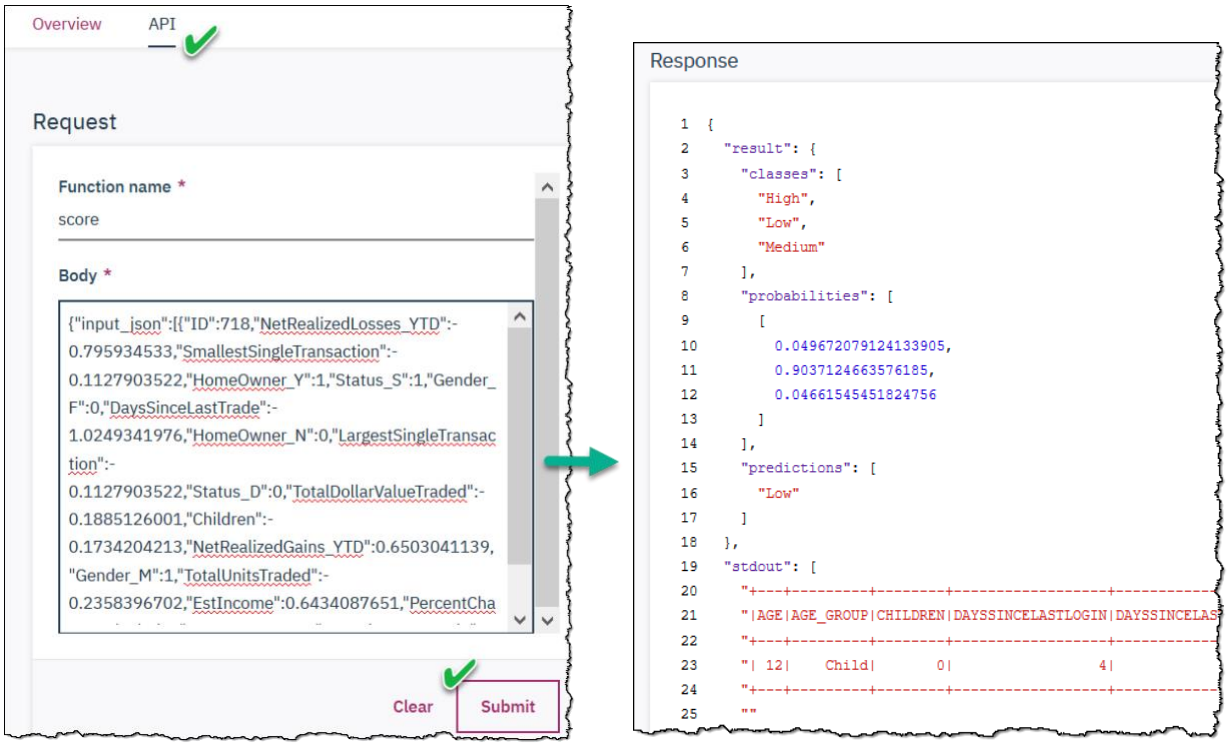


8.4 Test the deployment

__340. In the stocktrader deployment, click tab `API`.

Review the Function name and Body, and then click `Submit`

Review the Response output.

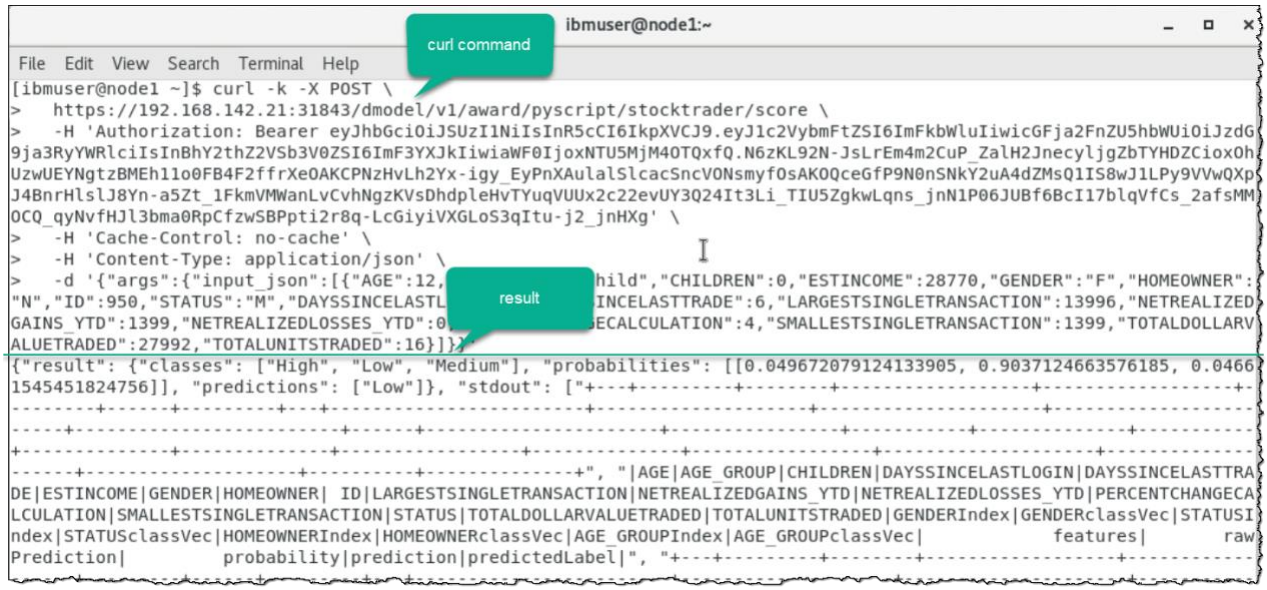


__341. Review `Generate code`

A `curl` command is generated that can be used to return values from the REST API, which the Developer can easily implement in an application.

__342. Below is an example of running the curl command in a Terminal window.

(You can try this for yourself if you want to. From the desktop open a `Terminal` ⇨ Type: `root` ⇨ Copy the generated curl command from the CPD console and paste it into the terminal ⇨ `Enter`.)



8.5 The deployment infused in the application Stock Trader After

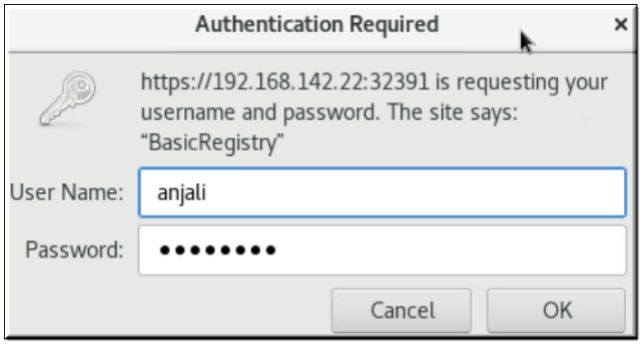
The machine learning model was created by data scientists and deployed using CPD platform. You will now see how a modern microservices-based application can consume it.

You reviewed the `Stock Trader After` application in the Executive Demo lab, but you will now review it in more detail.

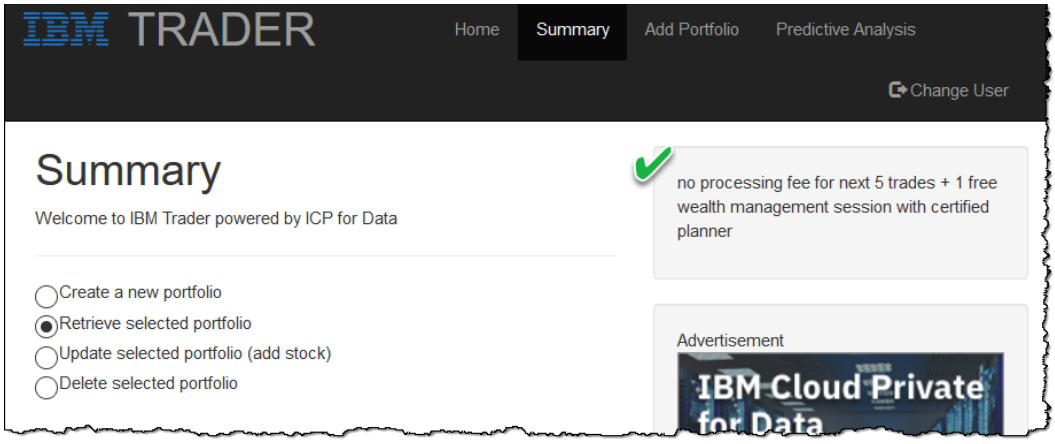
__343. Find and double-click desktop icon: `Stock Trader After.`

__344. Log in as *Username*: `anjali` and *Password*: `password`.

Click `OK`.



__345. Notice the offer box that shows after logging in to the application.

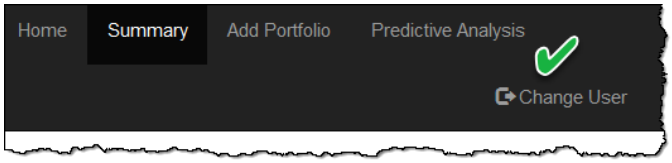


Based on user ID of the login, you get the demographics information about that user from the database which then provides that information to a REST API call from the backend of the microservice.

The demographics information is given to the model running on the CPD platform. The model returns a response (the churn prediction value) which is based on the business logic to present a retention offer to the user.

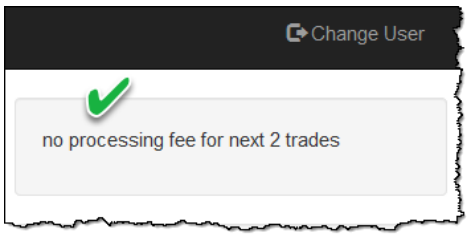
All of this is performed in real time as the user logs into the application.

__346. Click `Change User`.



__347. Type User: `dan` and Password: `password` and see the offer.

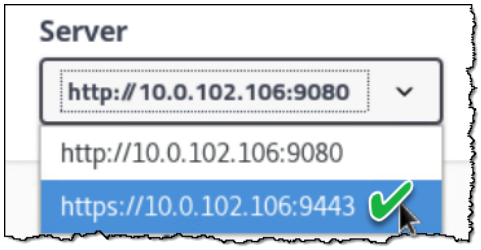
The offer is based on the risk assessment of the User ID.



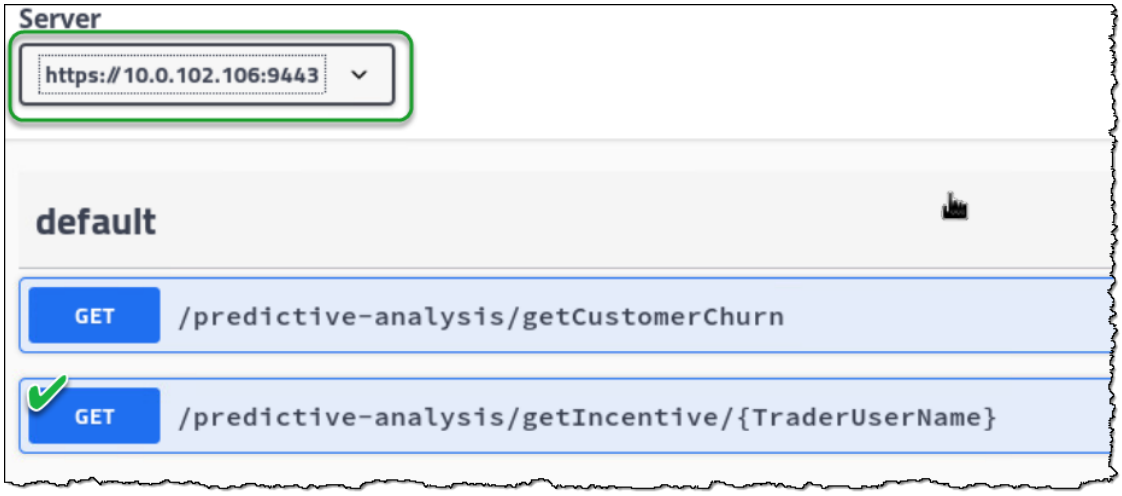
__348. Let's go further into the REST API call.

Click `API Explorer` desktop icon.

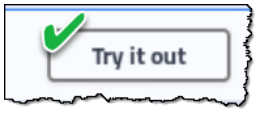
__349. Under `Server`, choose the `https` option.



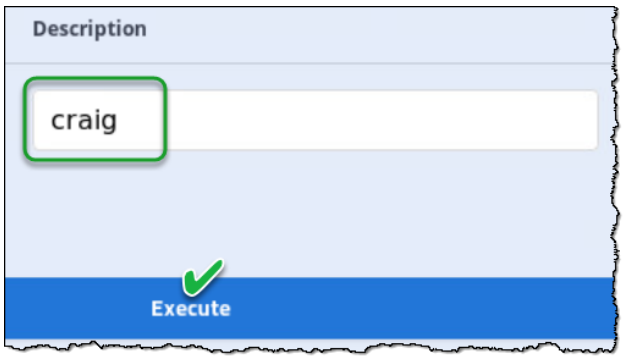
__350. Click `GET` on the second option: `/predictive-analysis/getIncentive/{TraderuserName}`



__351. Click `Try it out`.

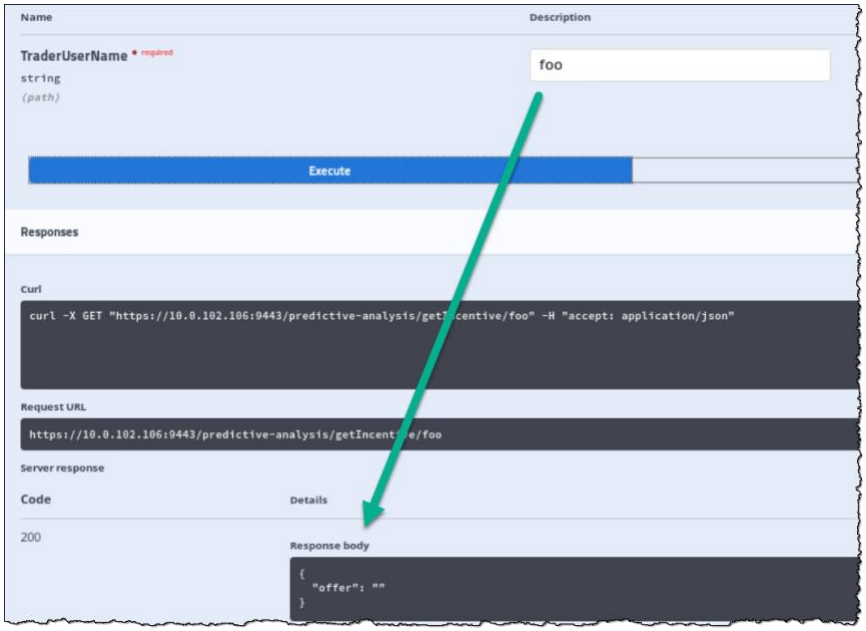


__352. Type `craig` and click `Execute`.



__353. Notice the responses: `curl` command with GET verb, 200 response code, and the response body which is consumed by the application.

__354. Try user `foo`. You will notice no response as we have no demographic information for this user.



8.6 Lab conclusion

As an organization, we collect and have access to more data than we can possibly analyze. Data is spread all over, in many silos, and is hard to find. Locating that one person in an organization who has the breadth and depth of knowledge of all our data is unlikely.

Cloud Pak for Data is the remedy. It provides an integrated data platform where data engineers and data stewards can prepare the data, data scientists can shop for data and build analytics models, business analysts can provide insights into the data and developers can infuse the results into the applications that run the business.

**** End of Lab 08: Deploy and Infuse**

Lab 09 Wrap-up

9.1 Lab overview

Let's do some wrap up tasks the various personas might do after completing the project.

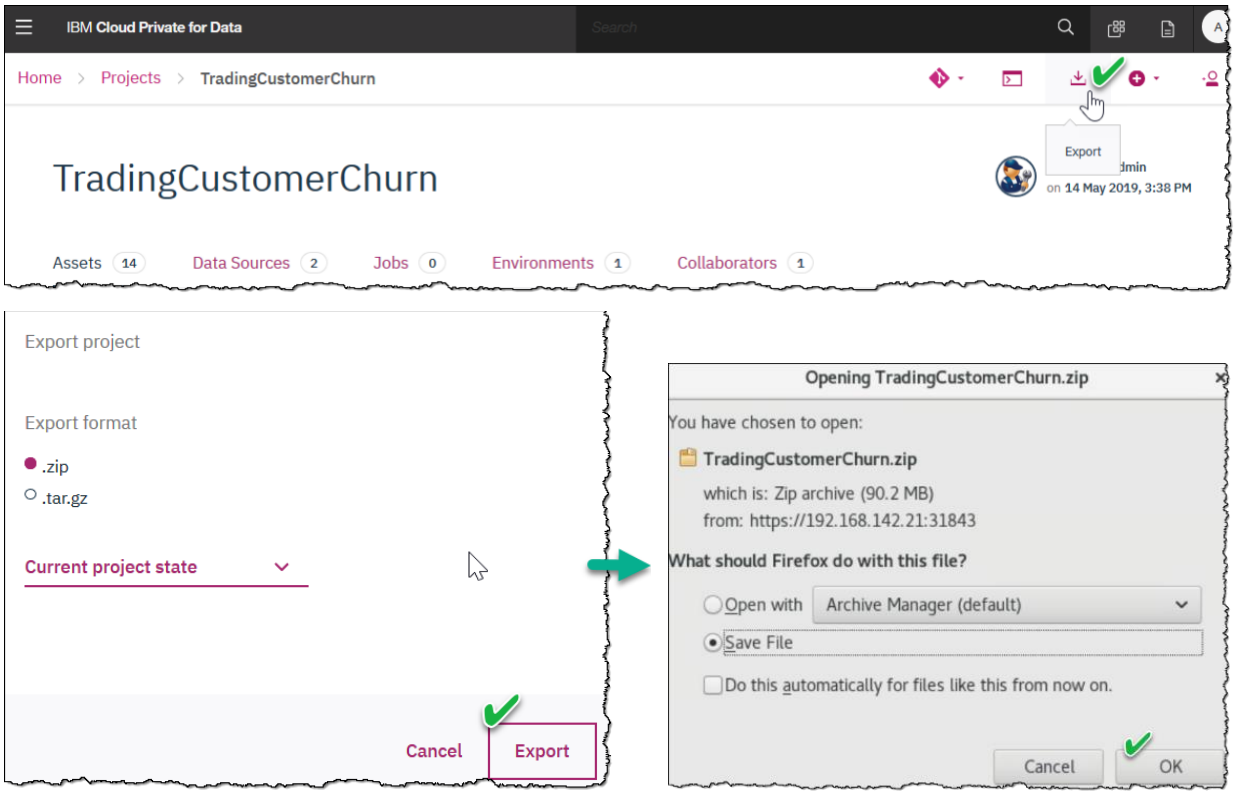
9.2 Data Scientist wrap-up

Data Scientist

9.2.1 Save the project

The Data Scientist saves her work by `Exporting` the project she just completed to a file on the server in a location that the OS administrators consistently backup.

__355. In the project screen, click on the `Export` icon at the top right of the screen

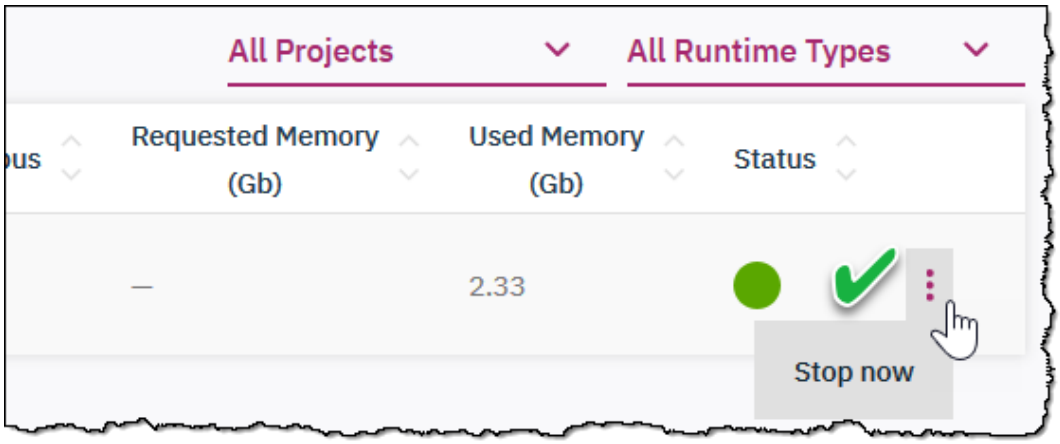


9.2.2 Stop the environment

She also saves resources on the CPD cluster by making sure to stop her environment after she is done with it.

__356. Click `Navigation Menu` ⇨ `My Instances` ⇨ `Environments`

__357. She clicks on the `ellipsis` on the Environment ⇨ `Stop now`



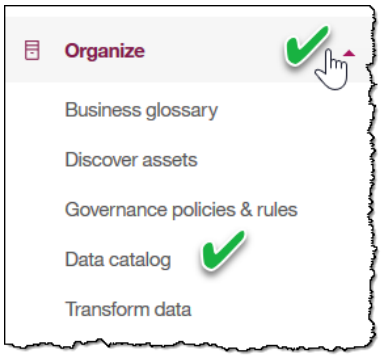
9.3 Data Steward wrap-up

Data Steward

9.3.1 Review the Data catalog

The Data Steward works to maintain the Data catalog to keep it accurate and up-to-date.

__358. From `Navigation Menu` ⇨ `Organize` ⇨ `Data catalog`



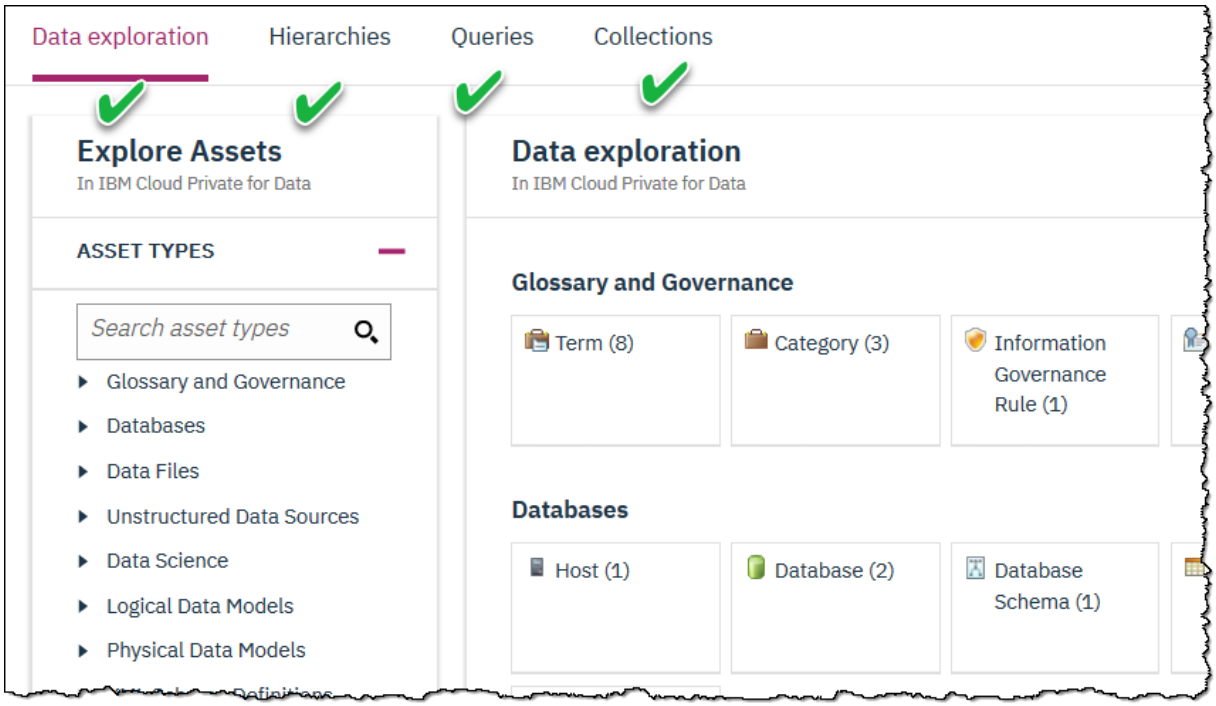
__359. From here, he can use the various Data Catalog features:

`Data Exploration:` to help him find data in the catalog.

`Hierarchies`: to display various asset groups in a hierarchical tree that helps him understand the meaning of the asset and its relationship with other assets.

`Queries`: helps him find and list assets, their properties and their relationships.

`Collections`: group together assets related to a specific subject.



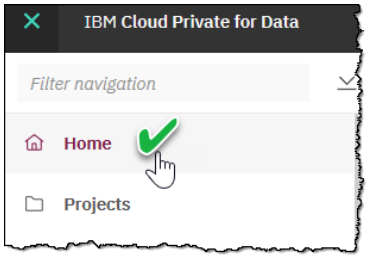
9.4 Data Engineer wrap-up

Data Engineer

9.4.1 Publish virtualized tables and views to the catalog

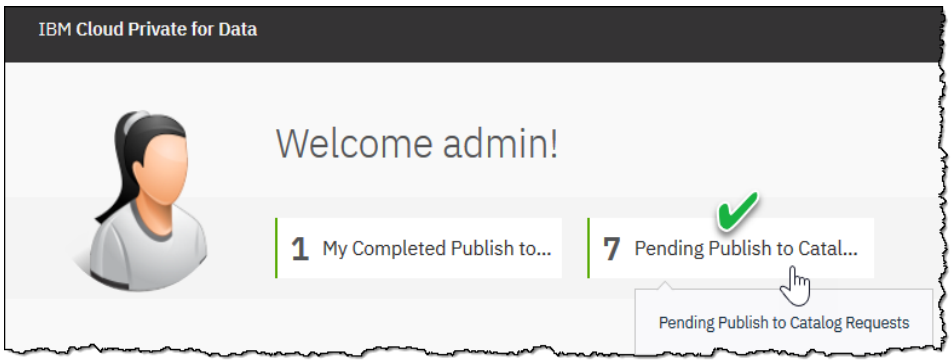
The Data Engineer wants to make the virtualized views she created available to any who wish to shop for them.

__360. From `Navigation Menu` ⇨ `Home`

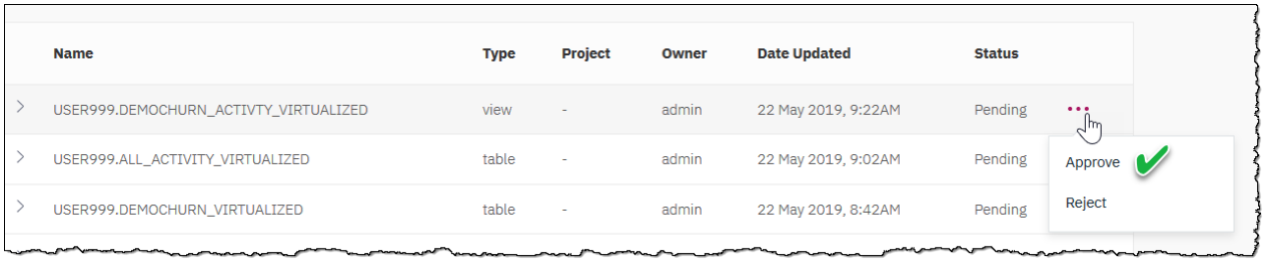


__361. After reviewing her home page first, she then…

Selects `Pending Publish to Catalog Requests`



__362. Selects `ellipsis` next to the virtualized object she wishes to publish ⇨ `Approve`



9.5 Administrator wrap-up

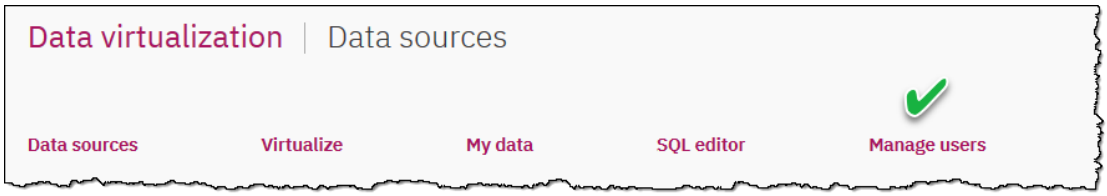
9.5.1 Grant Data Virtualization Access

The Administrator has been asked to open up Data Virtualization capabilities to a trusted user called `ibmuser` which allows the user to create them himself.

__363. The `Navigation Menu` ⇨ `Collect` ⇨ `Virtualized data`

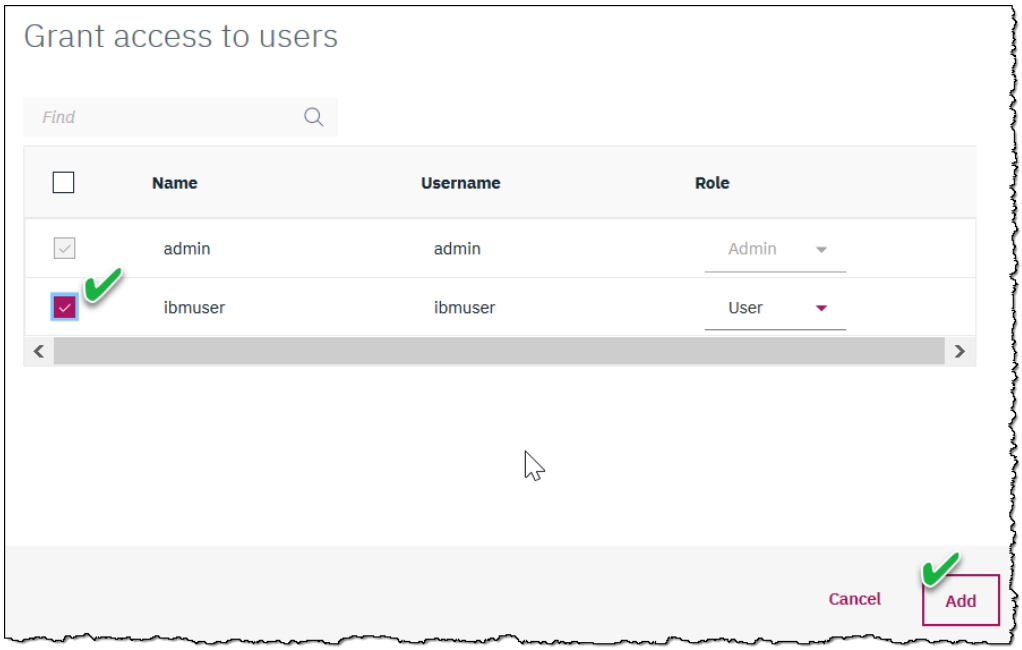
__364. Click `Menu`

__365. Click `Manage users`



__366. Click `+ Add users`

__367. Select user `ibmuser` ⇨ `Add`



9.6 Workshop conclusion

With this workshop, you can now see how Cloud Pak for Data turns your organization's data into a critical corporate asset with end-to-end data integration and collaboration on a modern, cloud-native platform.

You have now completed your IBM Journey to Cloud and AI: Analytics Modernization using Cloud Pak for Data.

**** End of Lab 09: Wrap-up**

Appendix A. Stock Trader Opening Bell Dashboard

In this lab, you will build the Stock Trader Opening Bell dashboard from scratch.

We begin by analyzing current trends of customer visits and daily trades in our Stock Trader application. We requested data engineers to provide a file with historical totals of visits and trades for the past year. This file was provided and deposited in our project where we all collaborate.

Build the dashboard

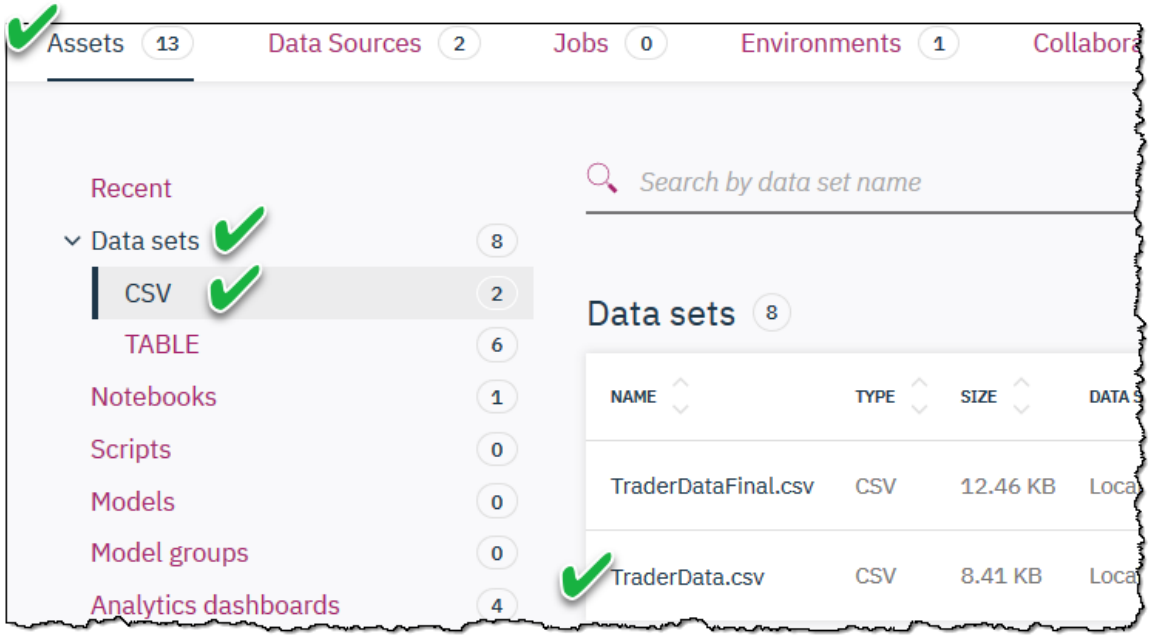
Let's build a dashboard with this data to see if we see any trends.

__368. From the top left Navigation Menu dropdown click `Projects.`

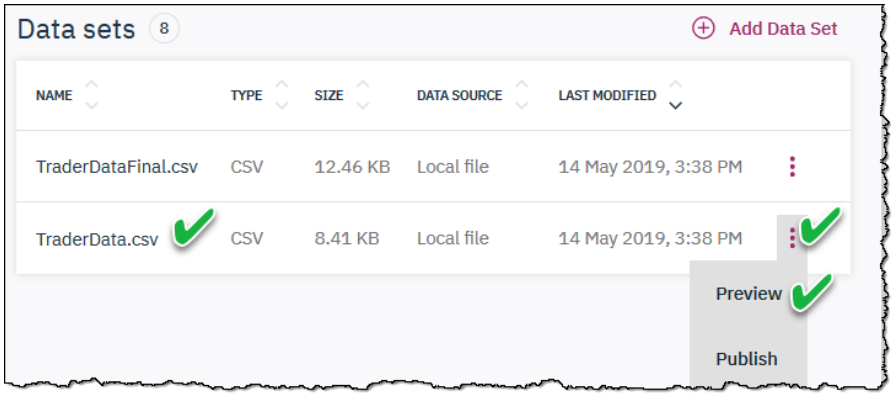
__369. Click project `TradingCustomerChurn`.

__370. Under `Assets` click `Data Sets` ⇨ `CSV`

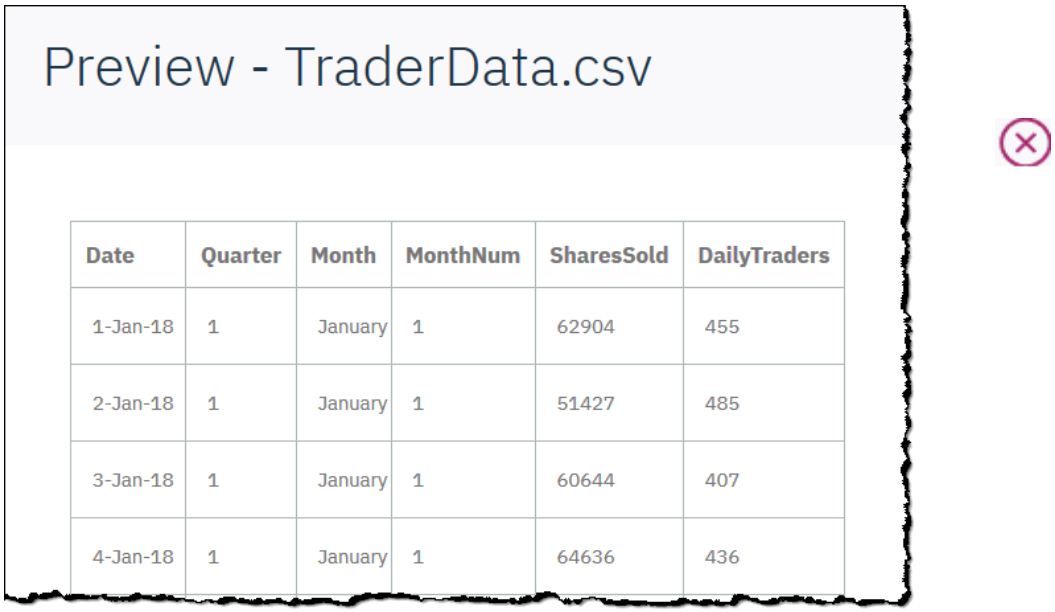
Notice `TraderData.csv` available in our project. We will use this data file to build the `Stock Trader – Opening Bell` dashboard.



__371. Click the ellipsis next to `TraderData.csv` then `Preview`

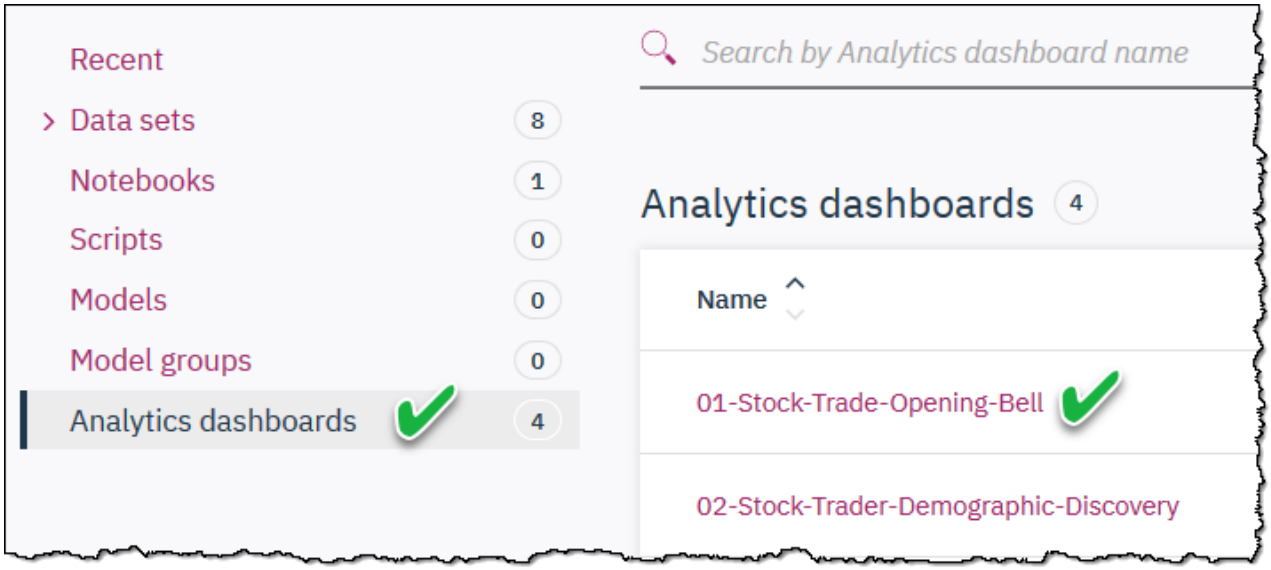


__372. Peruse the data, then the `X` to `Close` ⊗ this window.



Date	Quarter	Month	MonthNum	SharesSold	DailyTraders
1-Jan-18	1	January	1	62904	455
2-Jan-18	1	January	1	51427	485
3-Jan-18	1	January	1	60644	407
4-Jan-18	1	January	1	64636	436

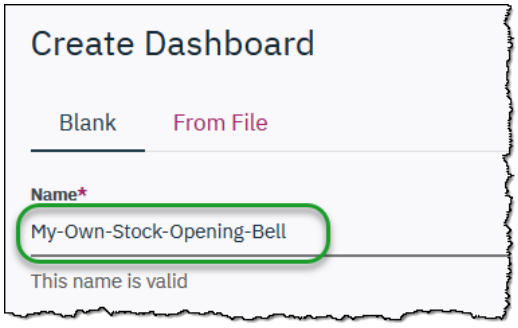
__373. Click `Analytics dashboards.`



We used the `01-Stock-Trade-Opening-Bell` dashboard in the Executive Demo lab and we will build this exact same dashboard here.

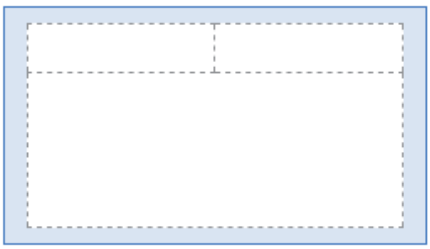
__374. Click `+ Add Analytics Dashboard`.

__375. In the Blank tab, type the Name: My-Own-Stock-Opening-Bell



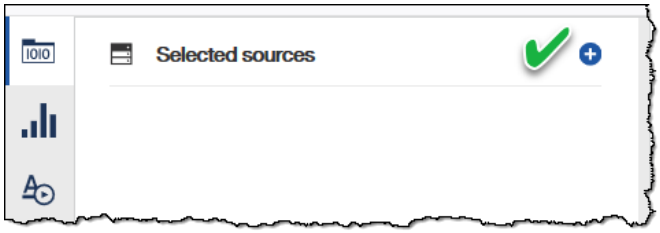
__376. Click Create.

__377. You are now presented with a choice of canvas templates. Select the one that looks as shown:

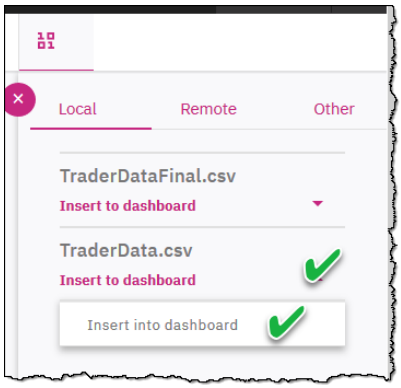


__378. Click OK.

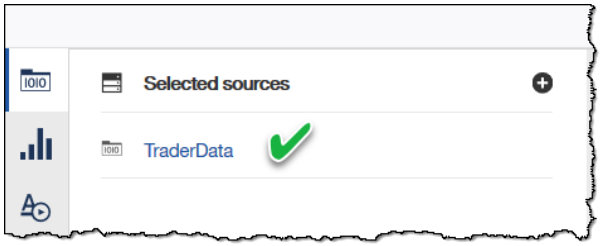
__379. From the Selected sources area near the top left of your screen, click 



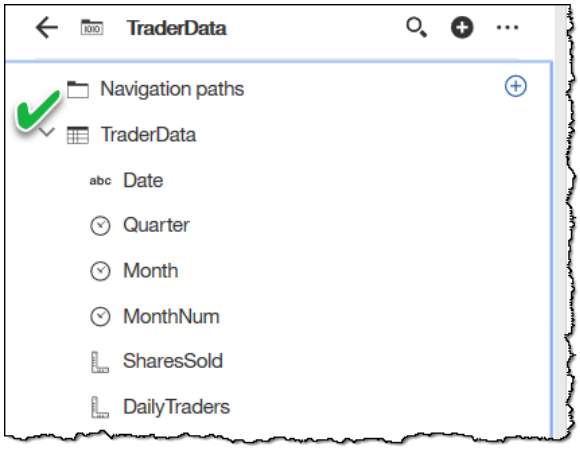
__380. The data pane will slide into the right side of the canvas and open to the Local tab by default. Click the down arrow of TraderData.csv and click Insert into dashboard.



__381. You will now see `TraderData` in the `Selected sources` area on the left of the screen.

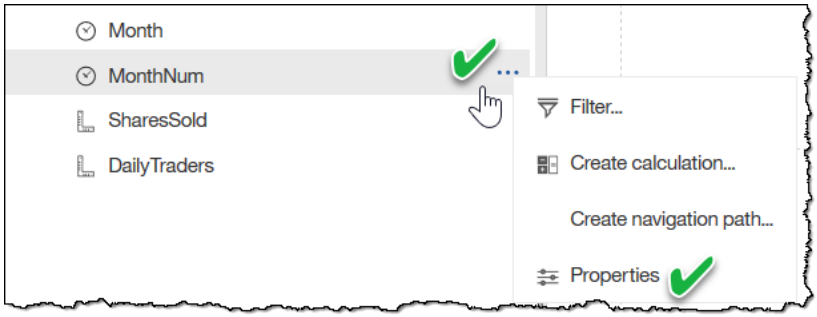


__382. Click `TraderData` and expand it to show all the data items in that data source file.



__383. Next we need to make sure the properties of the column data is what we want to be represented correctly within our dashboard.

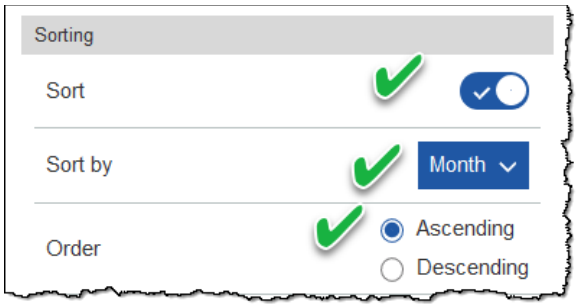
Click the `MonthNum` ellipsis, then select `Properties` from the flyout menu.



__384. `MonthNum` is an attribute on which we sort our months. Change `MonthNum` usage to be an `Attribute` and Aggregate to be `None`, then click `Close.`

__385. Select the Properties of Month by selecting the ellipse next to Month.

__386. Slide the Sort icon to turn it on and sort by MonthNum in Ascending order, then click Close.

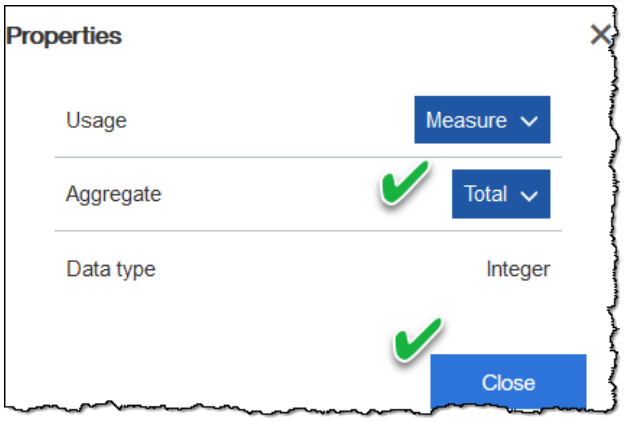


__387. Next, SET the aggregation of our measures for totaling.

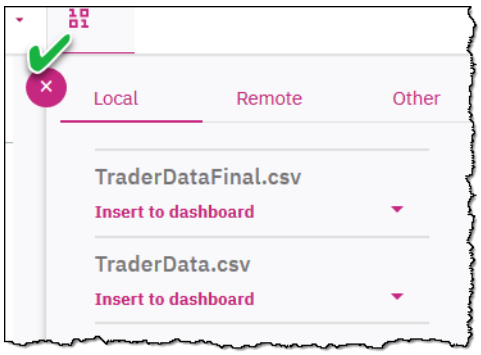
Hold the shift key down and click SharesSold and DailyTraders to select both.

Click the ellipsis and then click Properties.

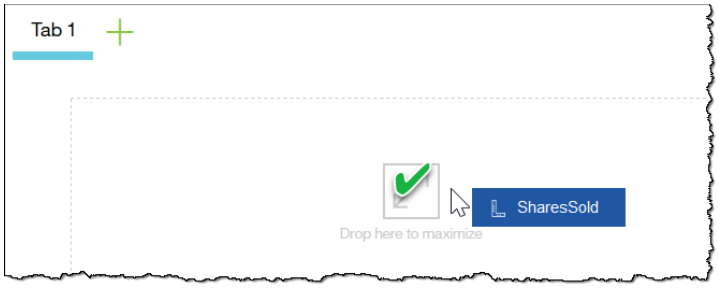
__388. Change *Aggregate* to Total from the drop-down menu and click Close.



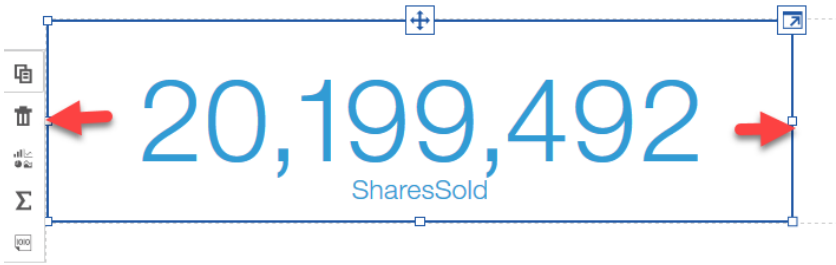
__389. Close the Data window at the top right of the screen by clicking the X.



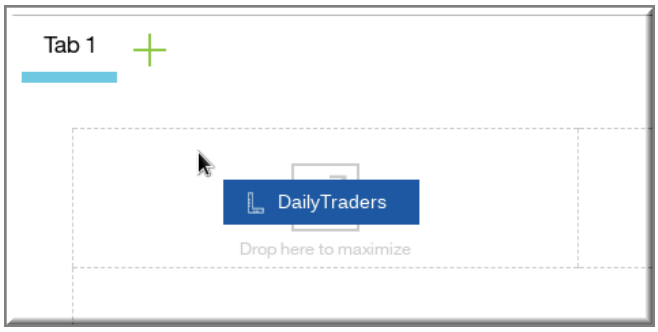
__390. Drag `SharesSold` to the top left box, hovering over the *Drop here to maximize* area when it turns blue. This gives us a total of `Shares sold` over all time.



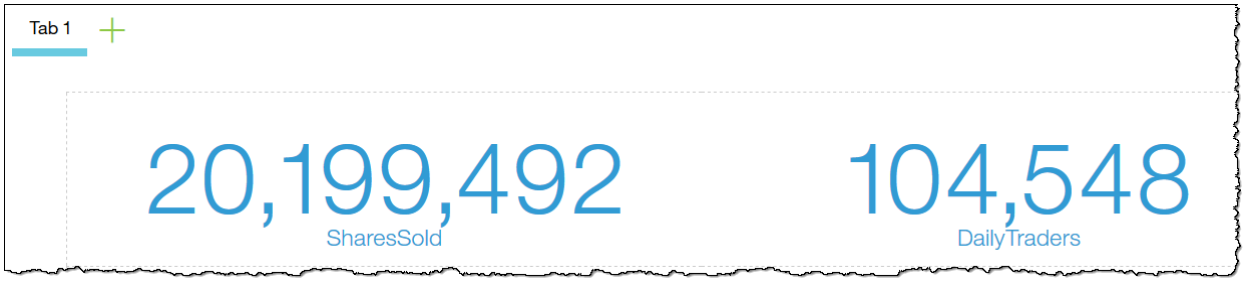
__391. If you did not drop at the right place it won't fill the template area. You can still adjust the guide to match with the outline of the left box.



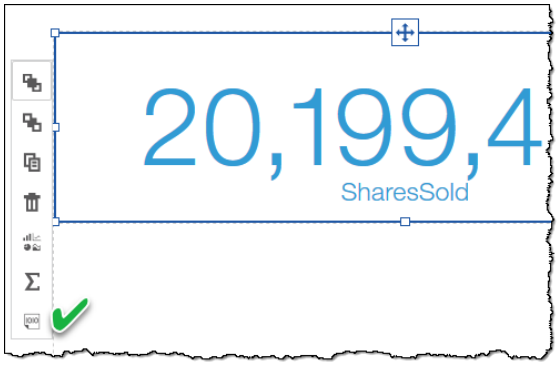
__392. Drag `DailyTraders` to the top right box, hovering over the *Drop here to maximize* area when it turns blue as well. This maximizes this metric in this box. This gives us a total of trades over all time.



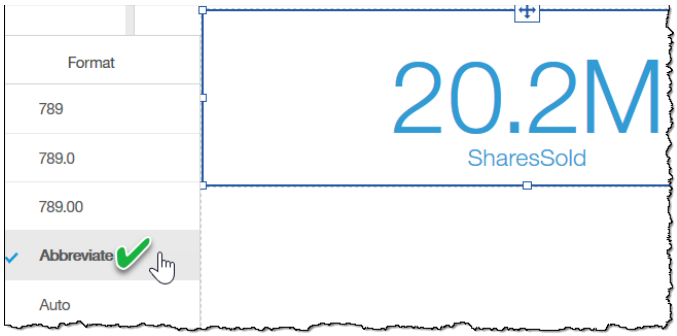
__393. After you complete both top boxes, the dashboard should display as shown:



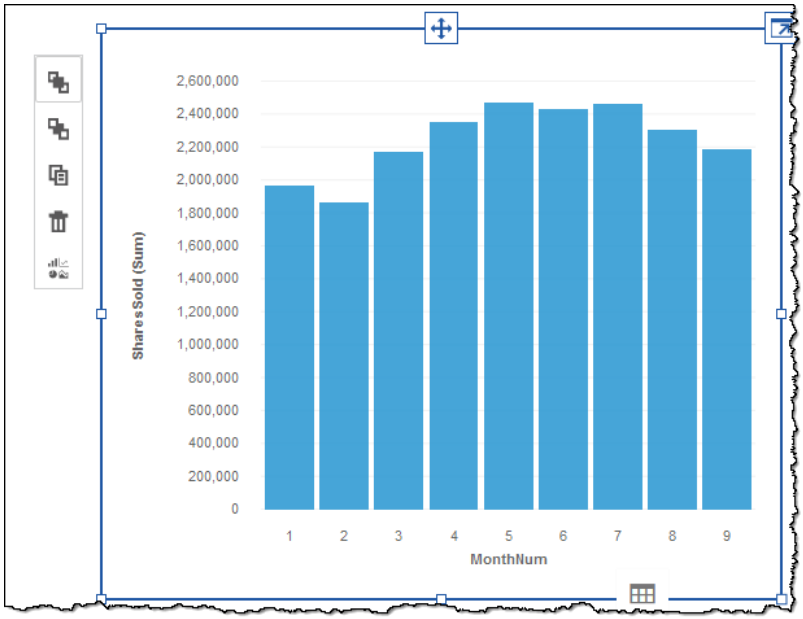
__394. Click the SharesSold visualization and select the format from the flyout menu.



__395. Click Abbreviate. Notice the number in the visualization changes format.

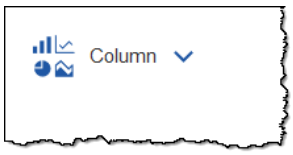


__396. Hold the shift key down, click Month and SharesSold and drag the two onto the lower canvas area. This time do not drop in the *Drop here to maximize* area.

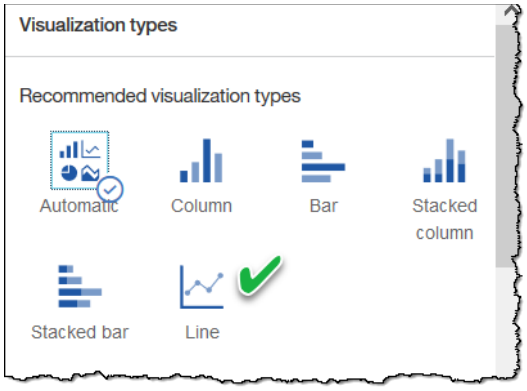


__397. Click the expand icon on top right of the visualization to allow you to make edits to it.

__398. Click the drop-down menu for visualizations which is now set to `Column`.

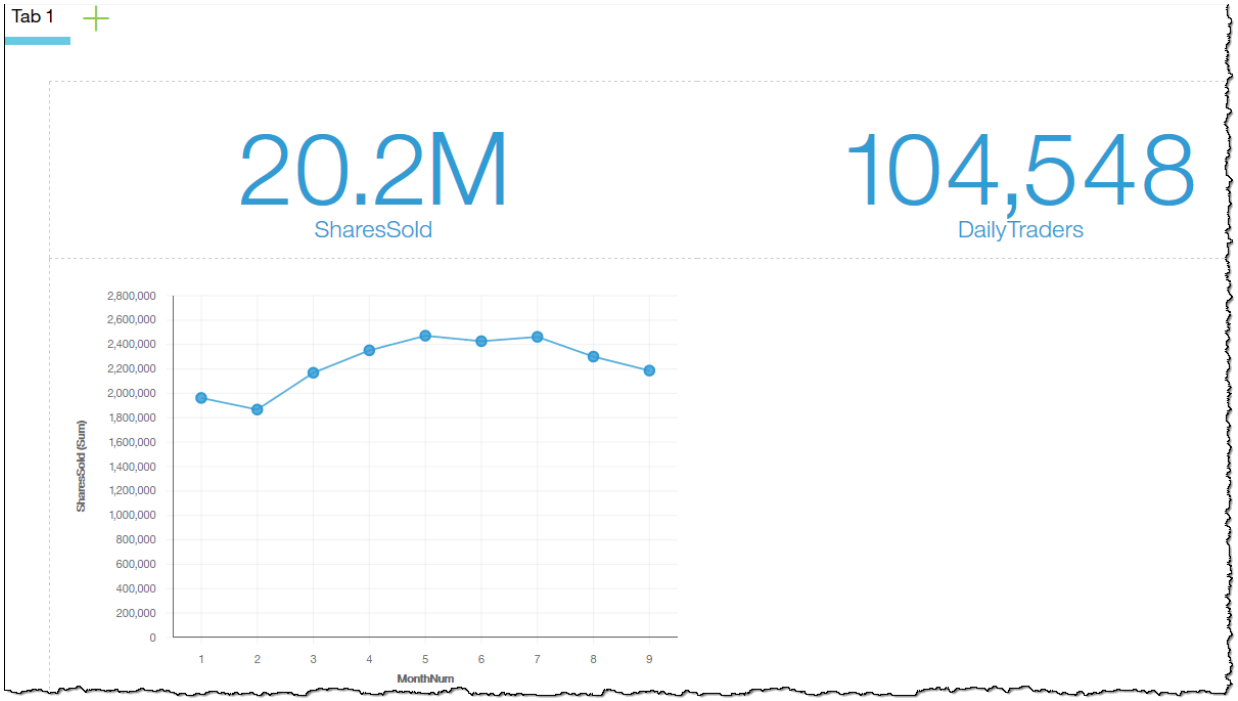


Select `Line` visualization type to change the visualization.



__399. Now select the `Collapse` icon to return the visualization to it's original size.

__400. Drag the chart so it aligns with the left side of the work area to look like this:

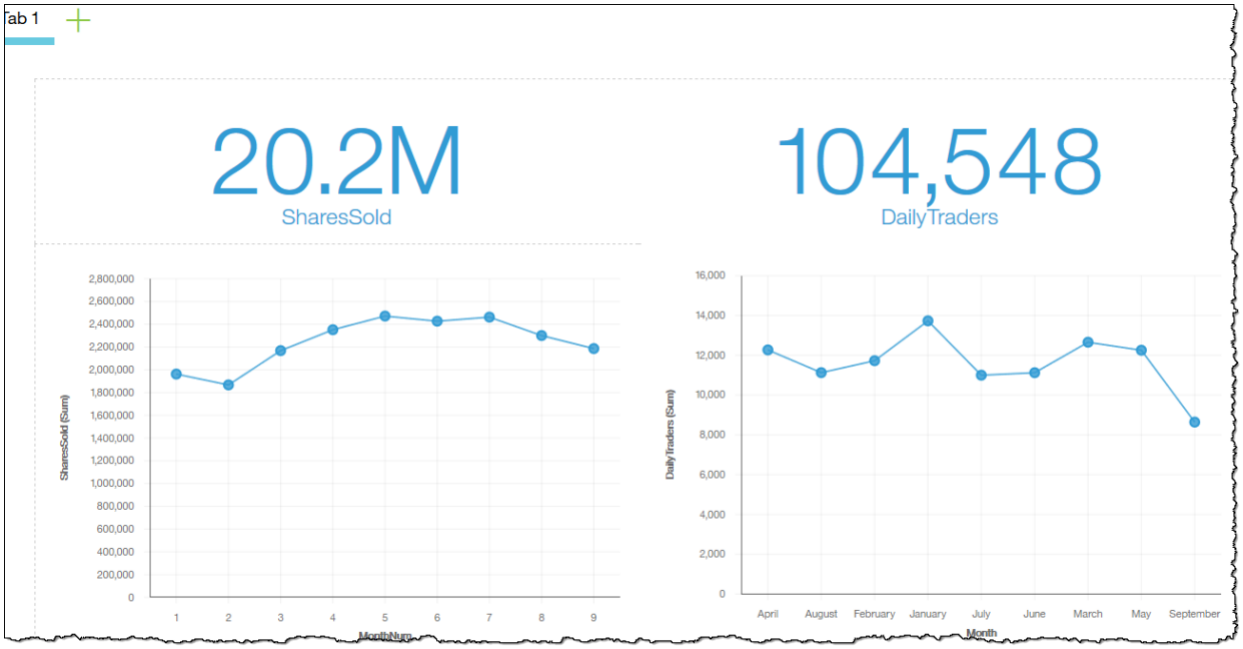


__401. Hold the shift key down, click `Month` and `DailyTraders` and drag the two onto the lower canvas area. Again do not drop in the *Drop here to maximize* area.

__402. Change the visualization from `Column` to `Line` as we did with the first graph. [Click **Expand**, change the type and click the same button to bring its in original size.]

__403. Adjust the top, bottom, left and right of the chart boundaries so that all the boxes are aligned.

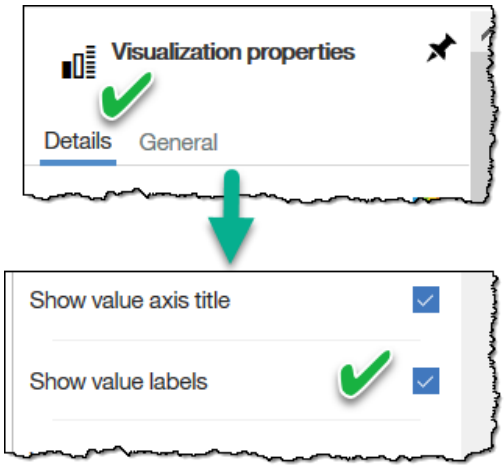
__404. The dashboard should display as below.



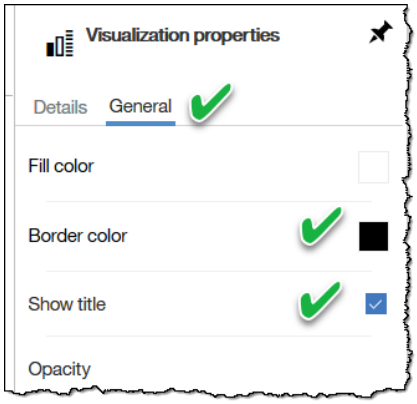
__405. Click the `Save` icon at the top of the screen to save your work

__406. Select (by clicking on) the bottom left line chart visualization, then select the `Properties` button at the top of the screen to format that visualization.

__407. From the `Details` tab, check `Show Value Labels`.

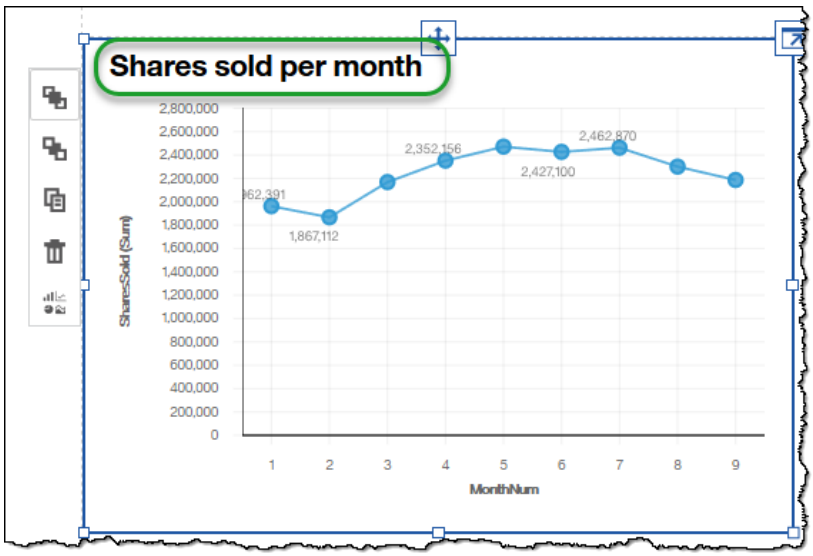


__408. Now select the `General` tab and check to `Show title` and change `Border color` to black

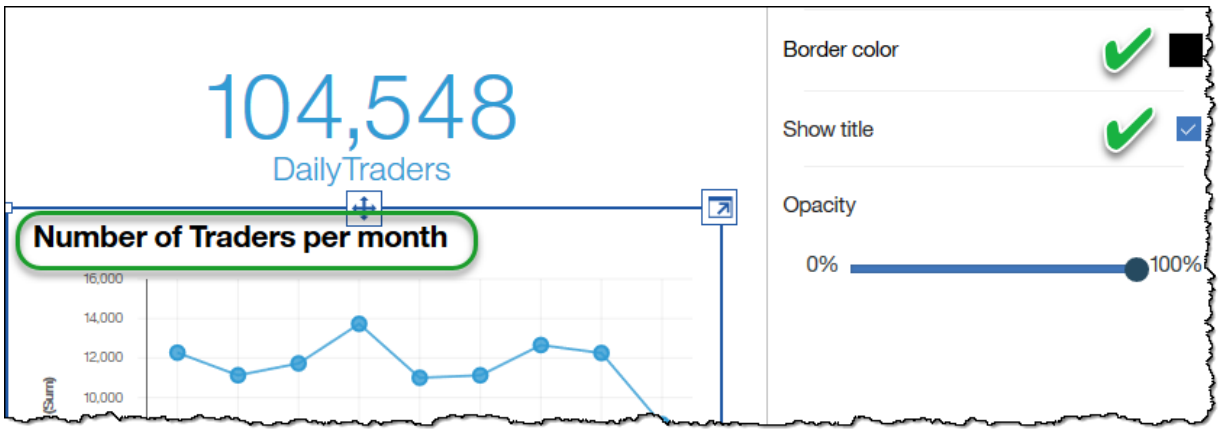


__409. In the bottom left visualization itself you can now type in a title for it.

Enter `Shares sold per month`.

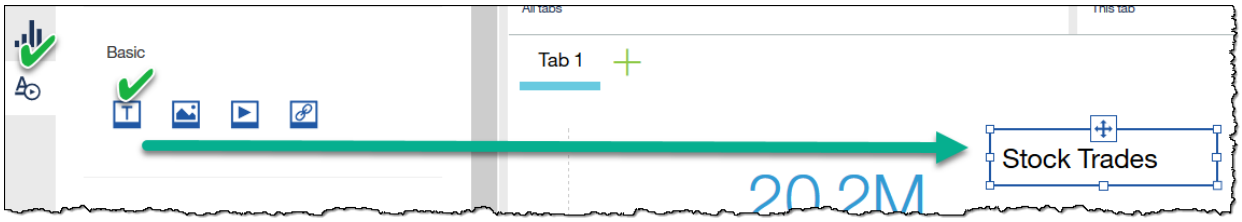


__410. Click the bottom right visualization and enter `Number of Traders per month` for the `Show title` option and change the `Border color` to black.

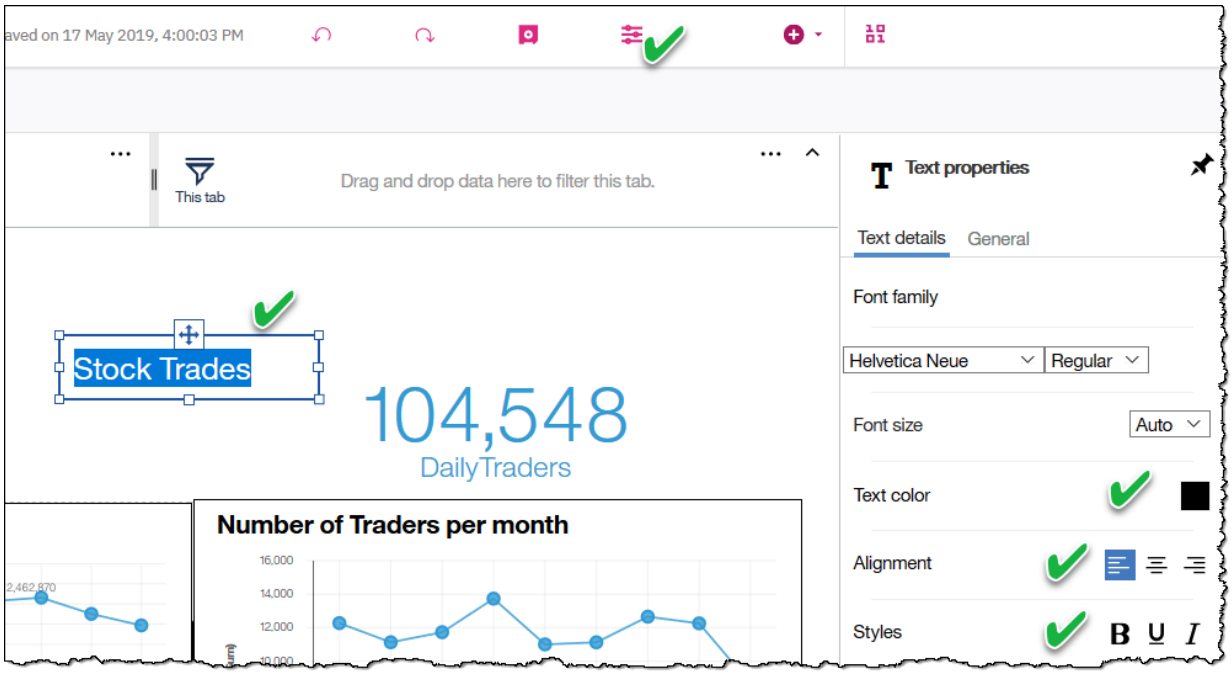


__411. On the left side of the canvas, click the `Widgets` menu and drag a `Text box` between the top two charts.

Title it **Stock Trades**. Adjust the box so the title stands out.



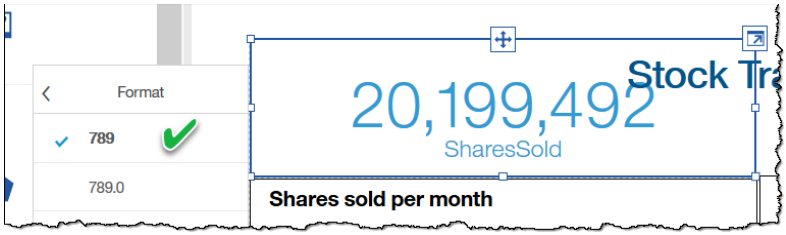
__412. Format the properties of the title `Stock Trades.` Make the text font size 46, bold, and dark blue in col. Then center it in the text box.



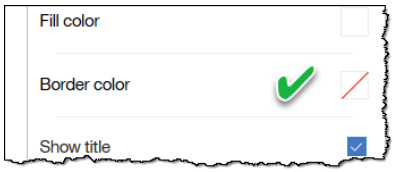
__413. The final results should look something like this:

__414. Click `Save` again to mark your progress here.

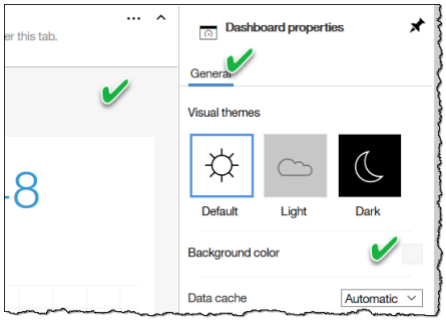
__415. Change the first visualization back from abbreviation to full integer number.



__416. Remove the borders from the bottom visualization by changing the color to nothing.

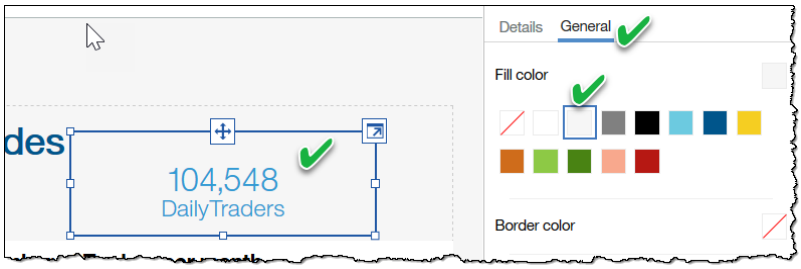


__417. Make the background of the entire dashboard grey by clicking on the work area and not on any visualization. Then from the `Dashboard properties` menu under `General`, change `Background color` to grey.



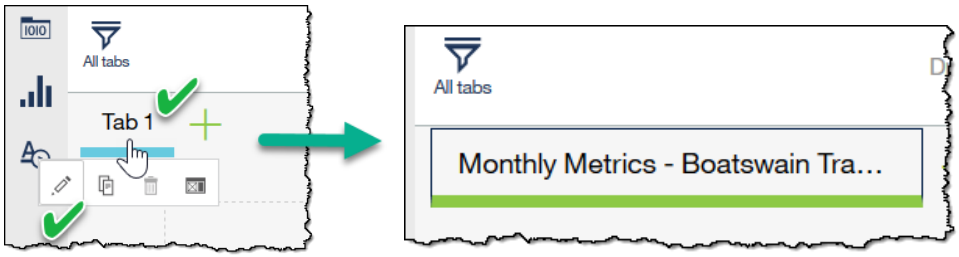
__418. Adjust the size of the top visualizations to be a bit smaller and centered over each of the bottom visualizations, leaving room for the title to be in full display and not crowded.

__419. Change the fill color of each of the top visualization to be grey to match the background of the dashboard itself.



__420. Edit the tab name of this set of visualizations for the dashboard.

Name it: `Monthly Metrics – Boatswain Trading`



__421. Do a final `Save` on this dashboard, then exit the dashboard and come back into in so you will not be in Edit mode.

The final dashboard should look like this:

You will notice from examining the charts that `Shares sold` is relatively flat and daily trades are falling off. We need to use Cloud Pak for Data to discover the WHY behind this trend.

Appendix B. Lab fixes

The following are fixes for lab issues that can come up.

Lab 02 – Executive Demo

9.7.1 Test Connection Fails

1. Test connection can fail if the Db2 user was not created in the pod.

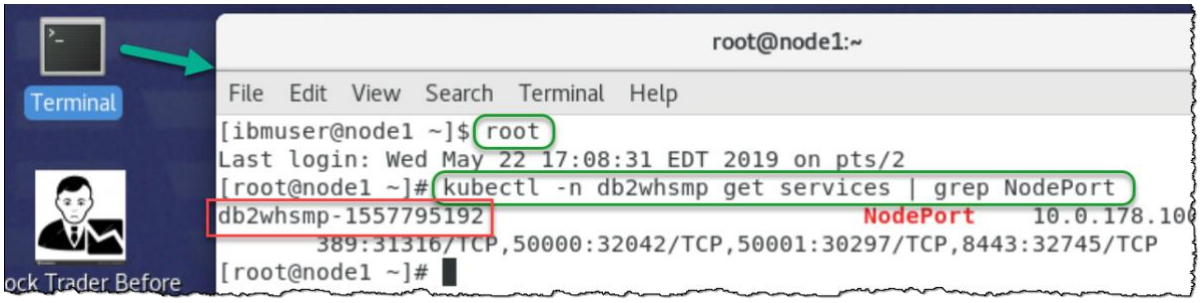
   Fix this by clicking on the desktop icon:

2. Test connection can also fail with it not being able to find the host. This means the host service name changed since the last time this workshop cluster was deployed.

   You can fix this by doing the following:

   - From the Desktop, open a Terminal window
   - Type: **`root`** (this makes you root user)
   - Type: **`kubectl -n db2whsmp get services | grep NodePort`**
   - The service name will be something like this: db2whsmp-nnnnnnnnn

   - Make sure this service name is the beginning part of the hostname in the JDBC URL string.

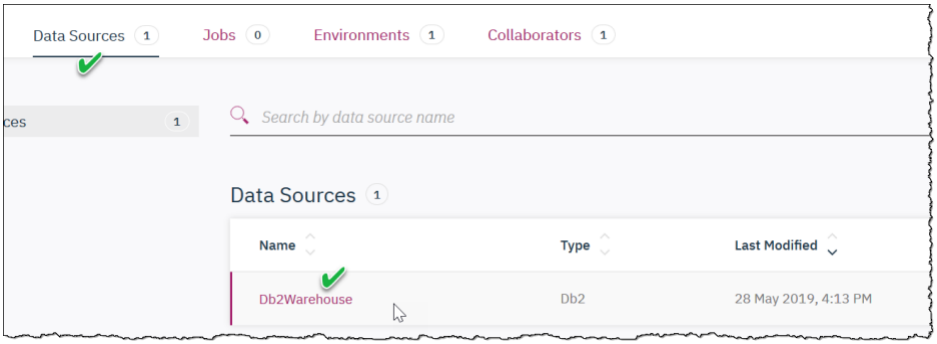
   

   - Retry the Test connection again.

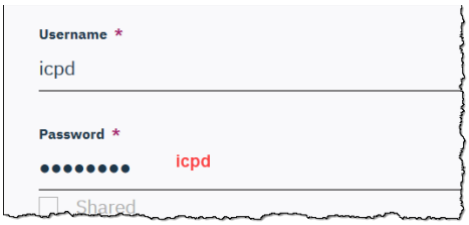
9.7.2 Dashboard doesn't render

Sometimes the dashboard doesn't render because the Data Source credentials needs to be re-initialized.

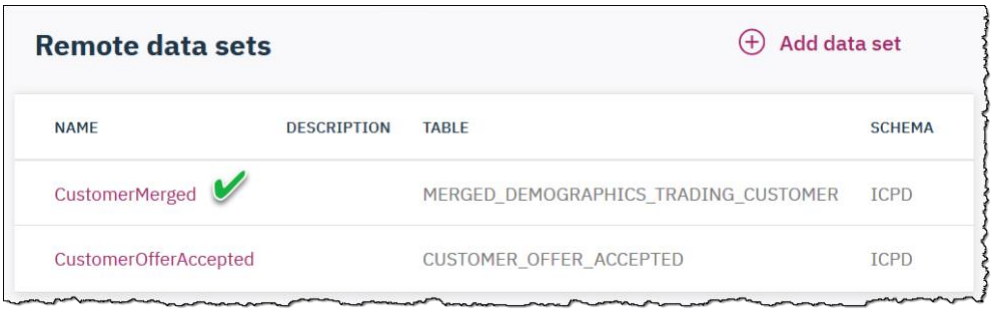
- In the Project go to Data Sources, then find Db2Warehouse.

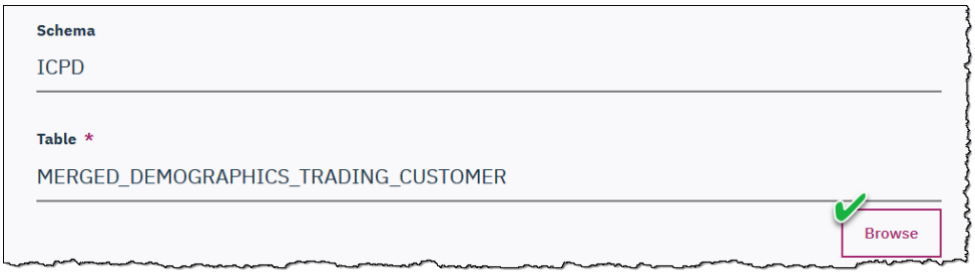


- Retype the Username/Password as below



- For each data set in this screen, click on it and browse to find the table name in the schema (as shown below.) This will refresh the schema/table for this screen.





- Click Save for each data set you do this for.
- Click Save for the Data Source itself to save all your work.
- Try the dashboard again.

Lab 04 – Organize

9.7.3 Runaway Discover

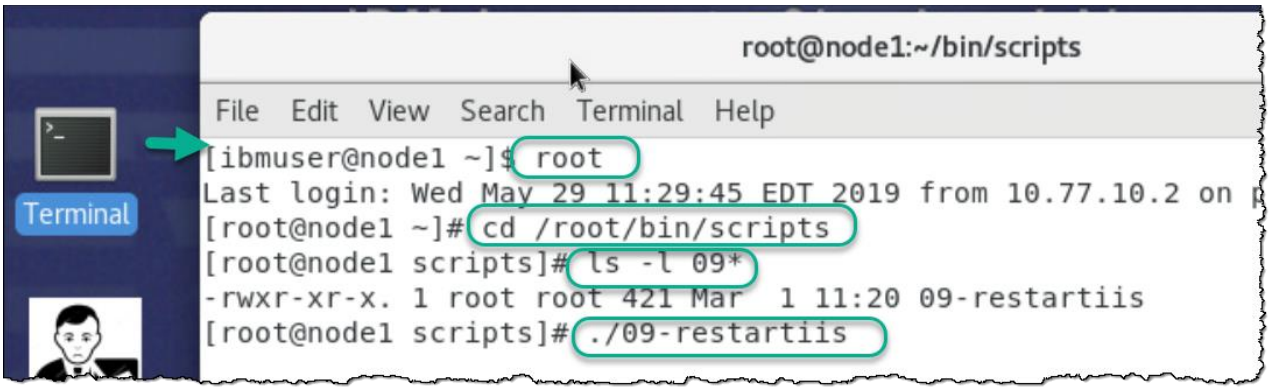
If the Discover process doesn't complete after 6 or 7 minutes, it may be runaway. You can fix this by doing the following:

- From the Desktop, open a Terminal window, then type:

  `root` (this makes you root user)

  `cd /root/bin/scripts`

  `./09-restartiis`



- After this script completes, wait 2 minutes for everything to initialize again in the pod.
- Try the Discover process again.

**** End of Appendix:**

Appendix C. Notices

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Appendix D. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	Db2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	System i	
System z	Tivoli	WebSphere	Workplace	System p	

Adobe, Acrobat, Portable Document Format (PDF), and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. See Java Guidelines

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark and a registered community trademark of the Office of Government Commerce and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Other company, product and service names may be trademarks or service marks of others.

The information contained in these materials is provided for informational purposes only and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors and are not intended to be a commitment to future product or feature availability in any way.