SCP3C05 – Predictive Analytics on the SAP Cloud System

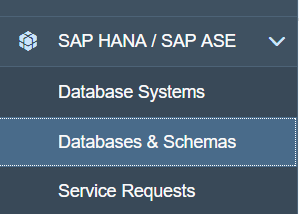
|  |  |
| --- | --- |
| **Product**  SAP oud Platform  **Level**  Undergraduate/Graduate  Beginner  **Focus**  Predictive Analytics  **Author** Ross Hightower | MOTIVATION  This case illustrates the predictive analytics algorithms on the SAP Cloud Platform  **PREREQUISITES**  SCP3C01 |
|  | |

This case illustrates two of the algorithms in the predictive analytics library (PAL) available on the SAP Cloud Plaltform. You can find help on these algorithms at <https://help.sap.com/pa>.

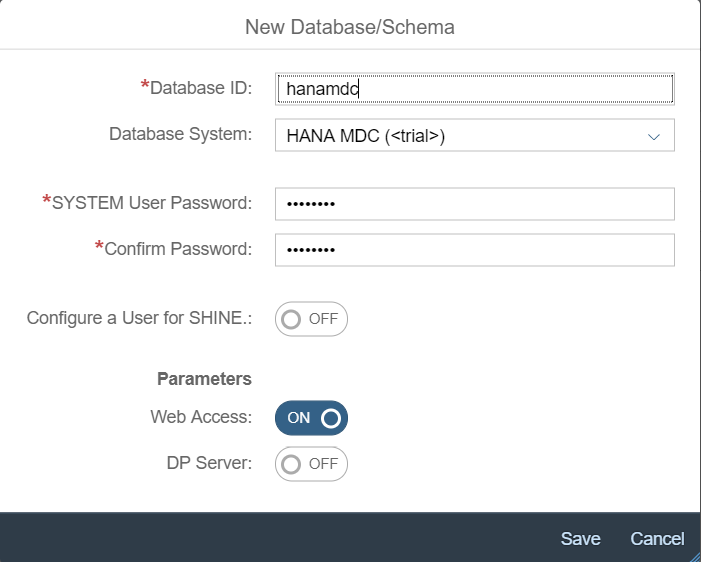
The algorithms use data stored in a HANA database and use HANA database tables for inputs and outputs. So the first thing we have to do is create a HANA database instance.

## Create the HANA MDC Instance

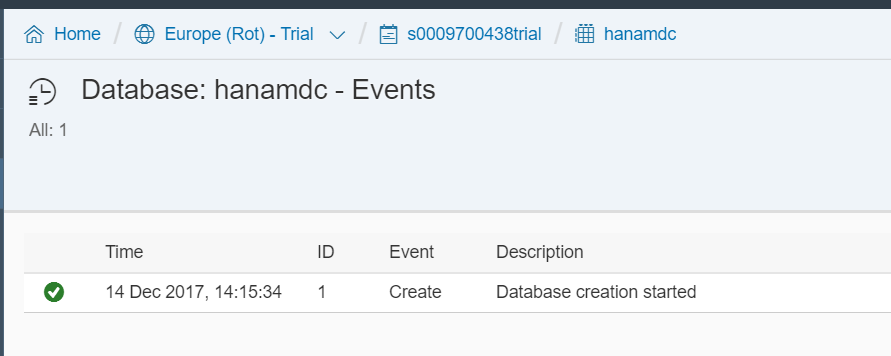
Logon to the SAP Cloud Platform Cockpit and navigate to SAP HANA / SAP ASE → Databases & Schemas.



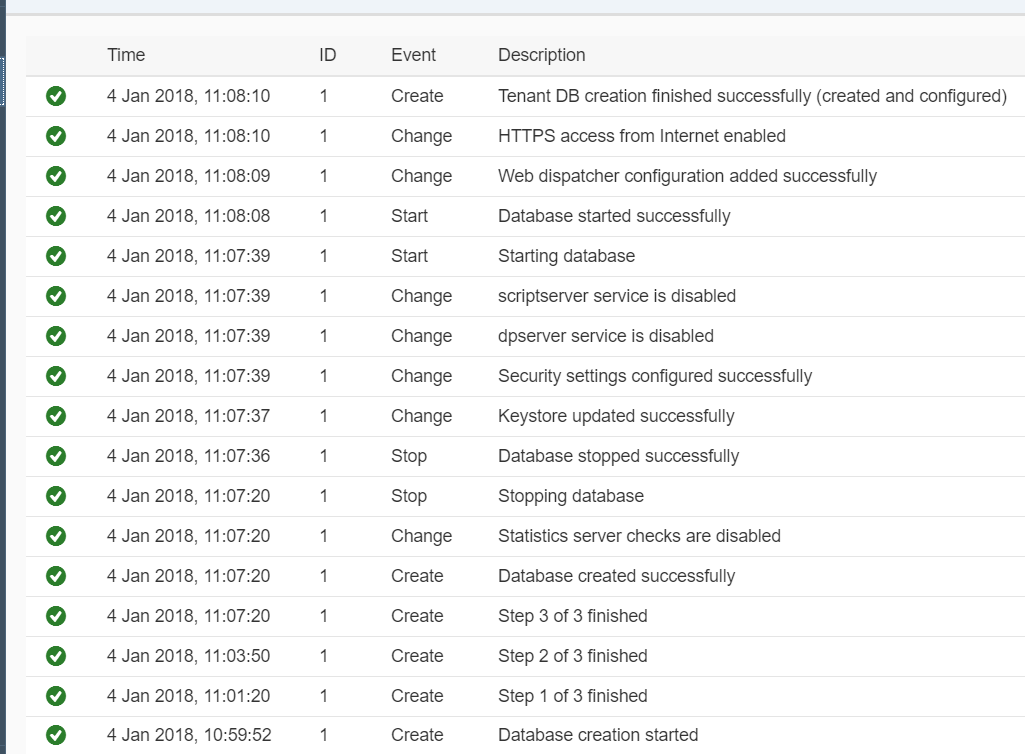
Click New to create a database instance. Enter data as shown below. The password must be a least 15 characters.



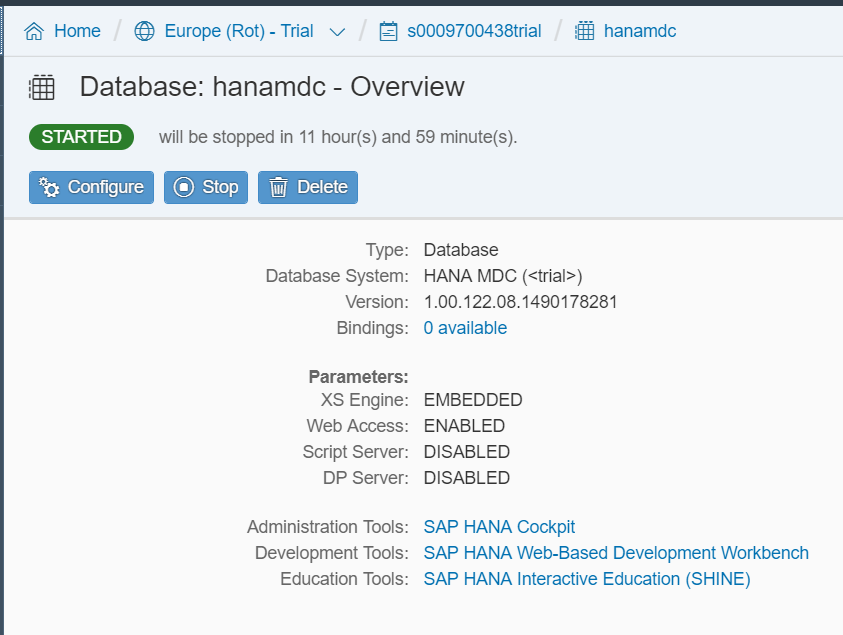
The database creation process will take several minutes. The Events page (shown below) will keep you advised of the status.



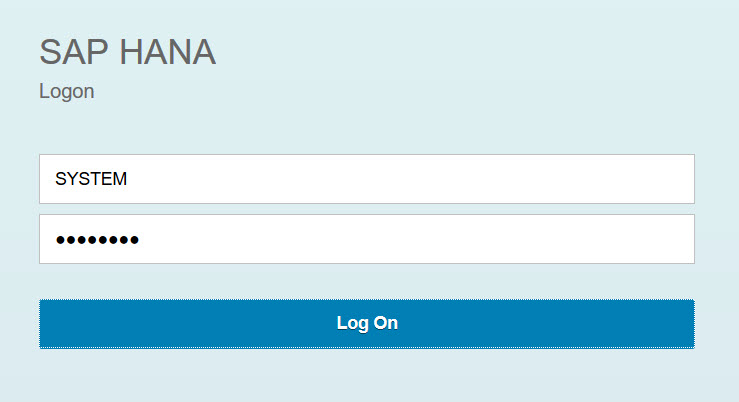
Once the creation process is done, the Events page will look like this:



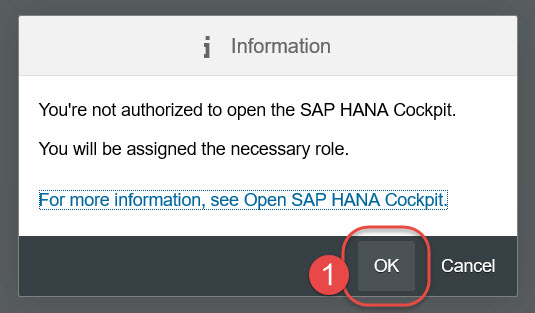
Click Overview on the menu on the left. Click Administration Tools: SAP HANA Cockpit.



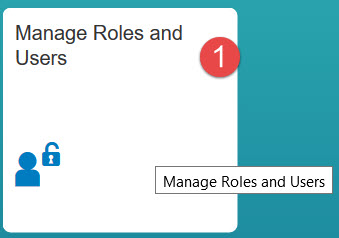
Login with the SYSTEM id and password you created for the database instance.



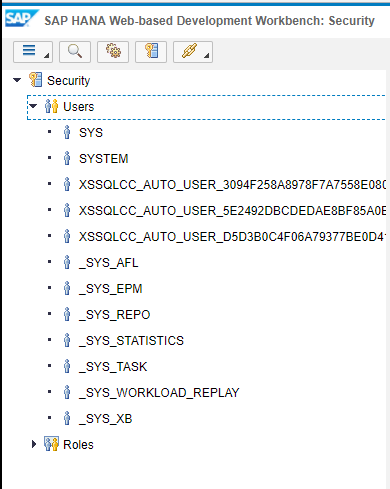
The first time you log on, you will see this dialog. Click OK and then Continue.



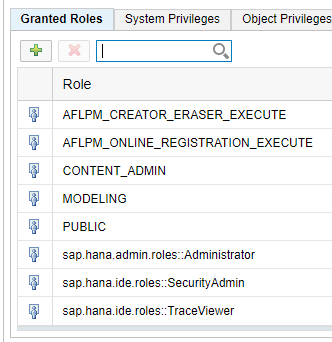
Once you are in the SAP HANA Cockpit, click the Manage Roles and Uses tile.



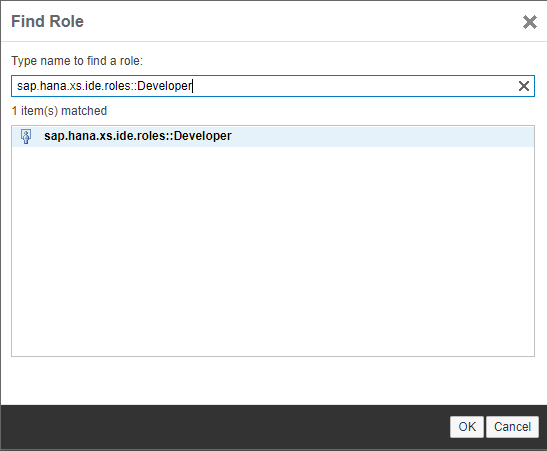
Once SAP HANA Web-based Development Workbench: Security opens expand the Uses node and click the SYSTEM user.



Click the green plus on the Granted Roles tab.

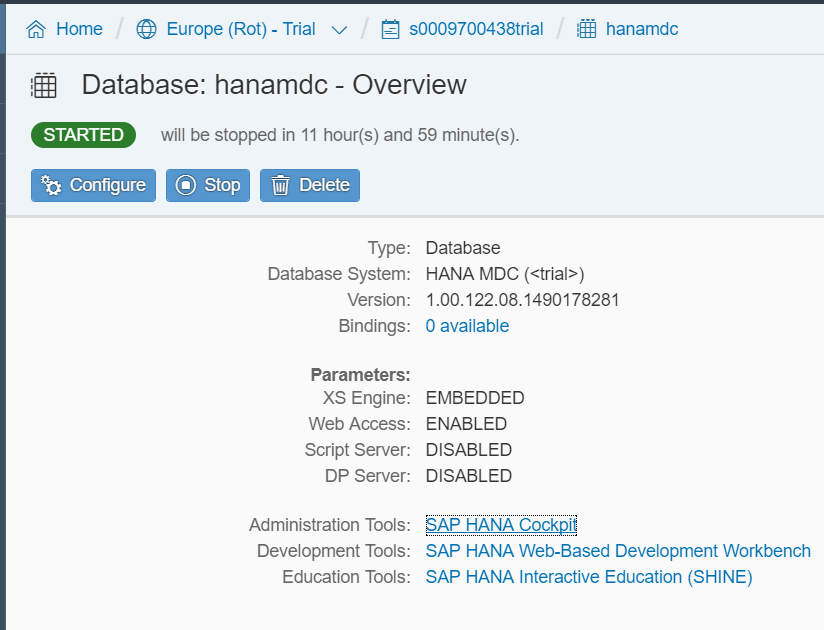


Search for the role sap.hana.xs.ide.roles::Developer, select it and click OK. Click  or use ctrl-S to save the user.

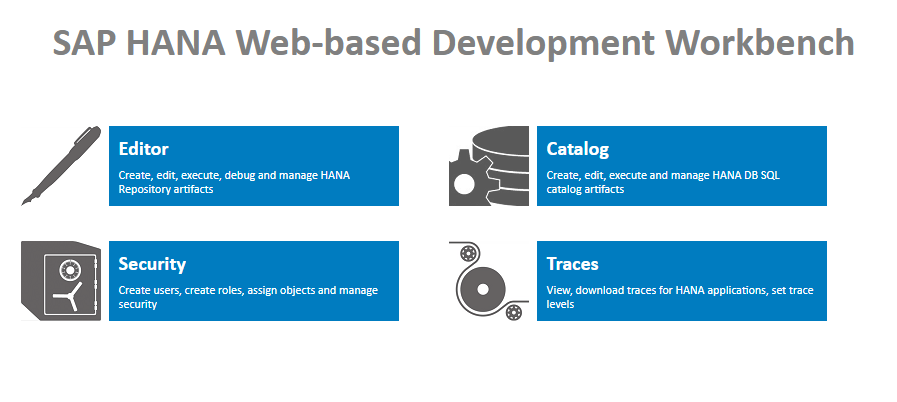


(Note this is normally a very bad idea. System administrators should not have developer roles).

You can close the Security Workbench and return to the SAP Cloud Cockpit. Click Development Tools: SAP HANA Web-Based Development Workbench.



Click Catalog.

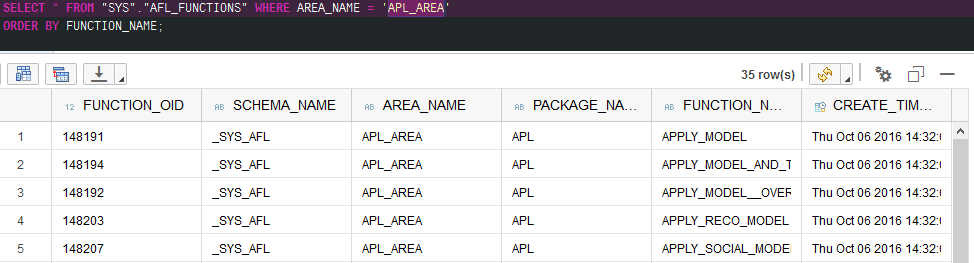


Once in the Catalog Workbench, click the SQL icon near the top of the screen. Enter the following code and click Run .

|  |
| --- |
| SELECT \* FROM "SYS"."AFL\_FUNCTIONS" WHERE AREA\_NAME = 'APL\_AREA'  ORDER BY FUNCTION\_NAME; |

Listing

If the [Automated Predictive Library](https://help.sap.com/viewer/7223667230cb471ea916200712a9c682/3.2.0.8/en-US) (APL) is installed on the SAP HANA instance, then the query will return the list of APL functions available in the library. The SAP Help page for the APL can be found at <https://help.sap.com/viewer/7223667230cb471ea916200712a9c682/3.2.0.8/en-US>



We will be using the FORECAST and APL functions. In order to use these functions you need to add a user to the AFL\_SYS\_AFL\_APL\_AREA\_EXECUTE role. Replace the SQL in the SQL Console with the code shown below and click Run. This code creates a user called APL\_USER and assigns the appropriate roles.

|  |
| --- |
| DROP USER APL\_USER CASCADE;  CREATE USER APL\_USER PASSWORD Welcome17Welcome17 NO FORCE\_FIRST\_PASSWORD\_CHANGE;  ALTER USER APL\_USER DISABLE PASSWORD LIFETIME;  call \_SYS\_REPO.GRANT\_ACTIVATED\_ROLE ('sap.hana.ide.roles::Developer','APL\_USER');  call \_SYS\_REPO.GRANT\_ACTIVATED\_ROLE ('sap.pa.apl.base.roles::APL\_EXECUTE','APL\_USER');  GRANT AFL\_\_SYS\_AFL\_APL\_AREA\_EXECUTE TO APL\_USER;  GRANT AFLPM\_CREATOR\_ERASER\_EXECUTE TO APL\_USER;  GRANT "CREATE SCHEMA" TO APL\_USER; |

Listing

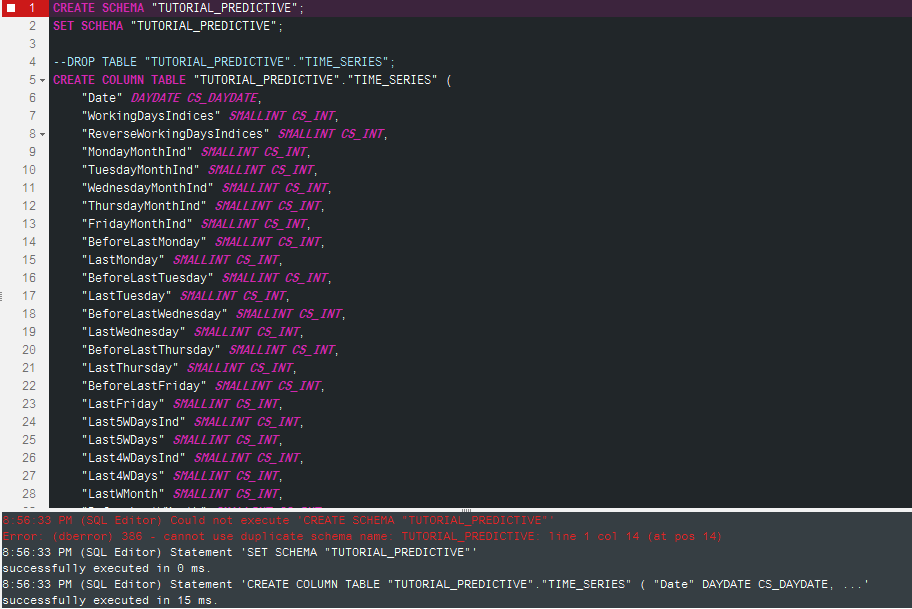
You will see the results in the console at the bottom of the page. Don’t worry about the errors. They are because you attempted to drop the user and it hasn’t been created yet. The drop is there in case you want to run this script again.

Logout of the Catalog Workbench using the icon on the top right and then logon with the APL\_USER and password Welcome17Welcome17.

Next, we need to create a table and import data into it. The data is a sample dataset available with SAP BusinessObjects Predictive Analytics. The schema is described in the appendix. Replace the SQL in the SQL Console with this code and click Run. You can ignore the errors.

|  |
| --- |
| Find the code in the accompanying code file |

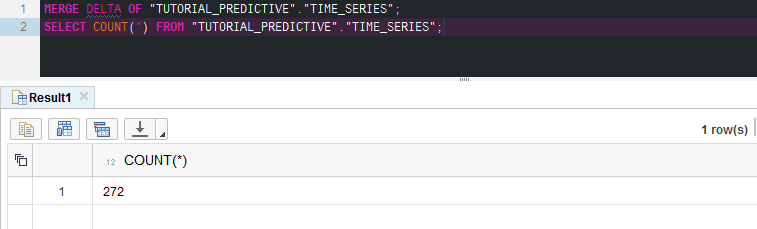
Listing



To see the number of records in the table, replace the SQL with the following and click Run.

|  |
| --- |
| SELECT COUNT(\*) FROM "TUTORIAL\_PREDICTIVE"."TIME\_SERIES"; |

Listing

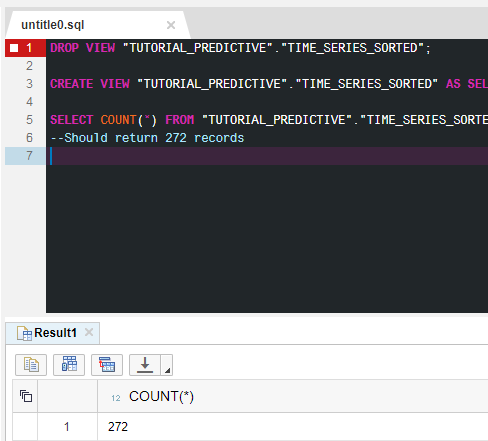


Next, use the SQL shown below to create a view to input into the forecast algorithm.

|  |
| --- |
| DROP VIEW "TUTORIAL\_PREDICTIVE"."TIME\_SERIES\_SORTED";  CREATE VIEW "TUTORIAL\_PREDICTIVE"."TIME\_SERIES\_SORTED" AS SELECT \* FROM "TUTORIAL\_PREDICTIVE"."TIME\_SERIES" ORDER BY "Date" ASC;  SELECT COUNT(\*) FROM "TUTORIAL\_PREDICTIVE"."TIME\_SERIES\_SORTED";  --Should return 272 records |

Listing

You should have 272 records.



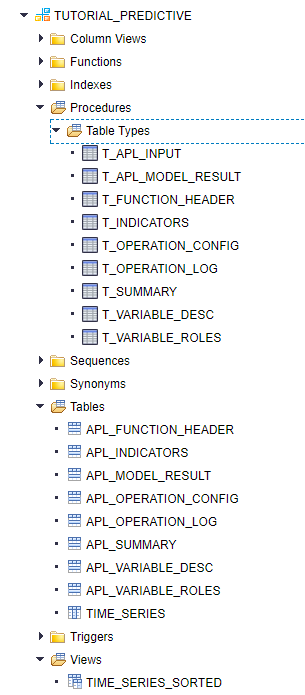
When running an APL procedure you must setup table types that define the tables structures used for inputs, outputs and configuraiton parameters. Use the code shown below to create those.

|  |
| --- |
| Find the code in the accompanying code file |

Listing

There will be many errors due to the fact that there are a number of drop statements.

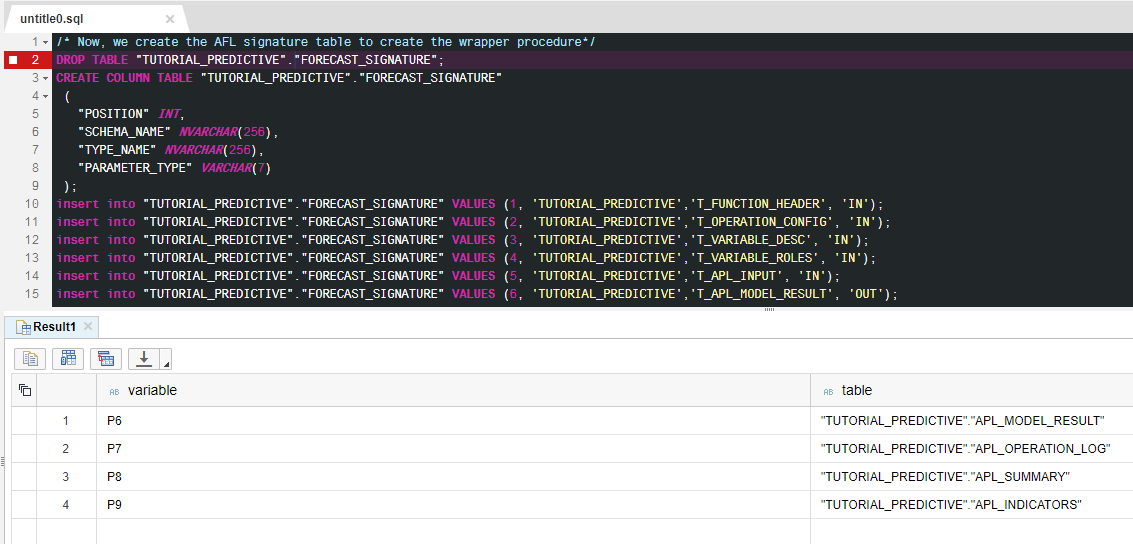
Once you run this code you will have the database artifacts shown below in your database schema. The Table Types are used for input and outputs from the PAL function.



Now you can execute the AFL wrapper with the code in Listing 6:

|  |
| --- |
| Find the code in the accompanying code file |

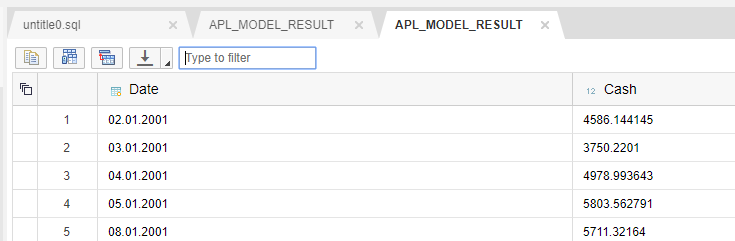
Listing



The table TUTORIAL\_PREDICTIVE::APL\_MODEL\_RESULT contains the results. Kts\_1 is the predicted value. You can use the following SQL statement to retrieve the results.

|  |
| --- |
| select \* from "TUTORIAL\_PREDICTIVE"."APL\_MODEL\_RESULT" order by "Date" desc; |

Listing



The following shows some summary statistics:

|  |
| --- |
| select \* from "TUTORIAL\_PREDICTIVE"."APL\_SUMMARY"; |

Listing

This statement provides an output log:

|  |
| --- |
| select \* from "TUTORIAL\_PREDICTIVE"."APL\_SUMMARY"; |

Listing

# ABC Analysis

Next, we will use the ABC analytics algorithm.

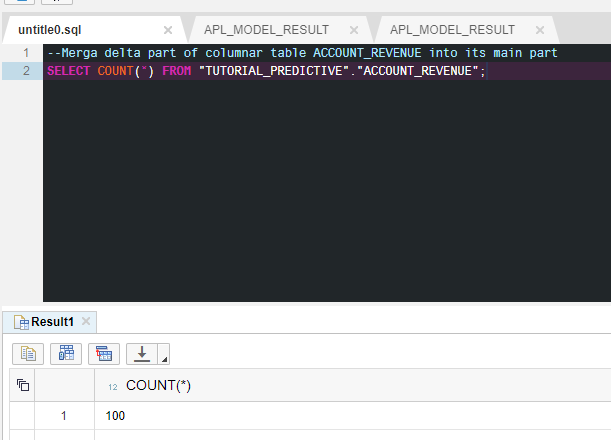
|  |
| --- |
| Find the code in the accompanying code file |

Listing

Next, run this SQL to check the number of records. There should be 100.

|  |
| --- |
| --Merga delta part of columnar table ACCOUNT\_REVENUE into its main part  SELECT COUNT(\*) FROM "TUTORIAL\_PREDICTIVE"."ACCOUNT\_REVENUE"; |

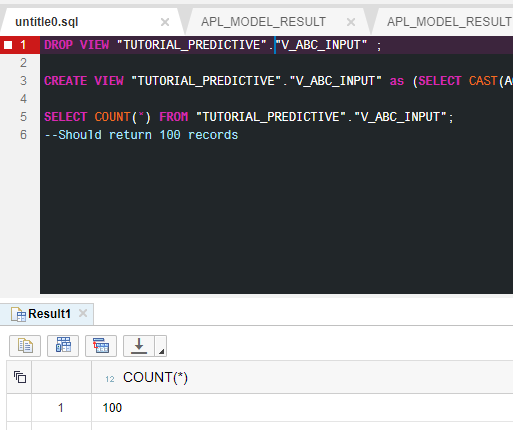
Listing



Replace the SQL code with the code shown below and run it. This code creates a view that will be the input to the algorithm.

|  |
| --- |
| DROP VIEW "TUTORIAL\_PREDICTIVE"."V\_ABC\_INPUT" ;  CREATE VIEW "TUTORIAL\_PREDICTIVE"."V\_ABC\_INPUT" as (SELECT CAST(ACCOUNTNR AS NVARCHAR(80)) as SOMEOBJECT, TOTAL\_AMOUNT as SOMEAMOUNT from "ACCOUNT\_REVENUE");  SELECT COUNT(\*) FROM "TUTORIAL\_PREDICTIVE"."V\_ABC\_INPUT";  --Should return 100 records |

Listing



Create the table types and wrapper procedure.

|  |
| --- |
| --Prepare Table Types  DROP TYPE "TUTORIAL\_PREDICTIVE".PAL\_ABC\_DATA\_VIEW\_T;  CREATE TYPE "TUTORIAL\_PREDICTIVE".PAL\_ABC\_DATA\_VIEW\_T AS TABLE ("SOMEOBJECT" NVARCHAR(80), "SOMEAMOUNT" DOUBLE);  DROP TYPE "TUTORIAL\_PREDICTIVE".PAL\_CONTROL\_T;  CREATE TYPE "TUTORIAL\_PREDICTIVE".PAL\_CONTROL\_T AS TABLE("Name" VARCHAR(100), "intArgs" INT, "doubleArgs" DOUBLE, "strArgs" VARCHAR(100));  DROP TYPE "TUTORIAL\_PREDICTIVE".PAL\_ABC\_RESULT\_T;  CREATE TYPE "TUTORIAL\_PREDICTIVE".PAL\_ABC\_RESULT\_T AS TABLE("ABC" NVARCHAR(10), "SOMEOBJECT" NVARCHAR(80));  --Create the proc using the wrapper  --Define and populate parameters table  DROP TABLE "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_PDATA\_TBL";  CREATE COLUMN TABLE "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_PDATA\_TBL"  (  "POSITION" INT,  "SCHEMA\_NAME" NVARCHAR(256),  "TYPE\_NAME" NVARCHAR(256),  "PARAMETER\_TYPE" VARCHAR(7)  );  INSERT INTO "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_PDATA\_TBL" VALUES (1,'TUTORIAL\_PREDICTIVE','PAL\_ABC\_DATA\_VIEW\_T', 'in');  INSERT INTO "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_PDATA\_TBL" VALUES (2,'TUTORIAL\_PREDICTIVE','PAL\_CONTROL\_T', 'in');  INSERT INTO "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_PDATA\_TBL" VALUES (3,'TUTORIAL\_PREDICTIVE','PAL\_ABC\_RESULT\_T','out');  --Call the wrapper procedure to generate our ABC calculation procedure  CALL SYS.AFLLANG\_WRAPPER\_PROCEDURE\_DROP('TUTORIAL\_PREDICTIVE', 'PAL\_ABC\_CALCULATE');  CALL SYS.AFLLANG\_WRAPPER\_PROCEDURE\_CREATE ('AFLPAL', 'ABC', 'TUTORIAL\_PREDICTIVE', 'PAL\_ABC\_CALCULATE', "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_PDATA\_TBL"); |

Listing

Run the wrapper.

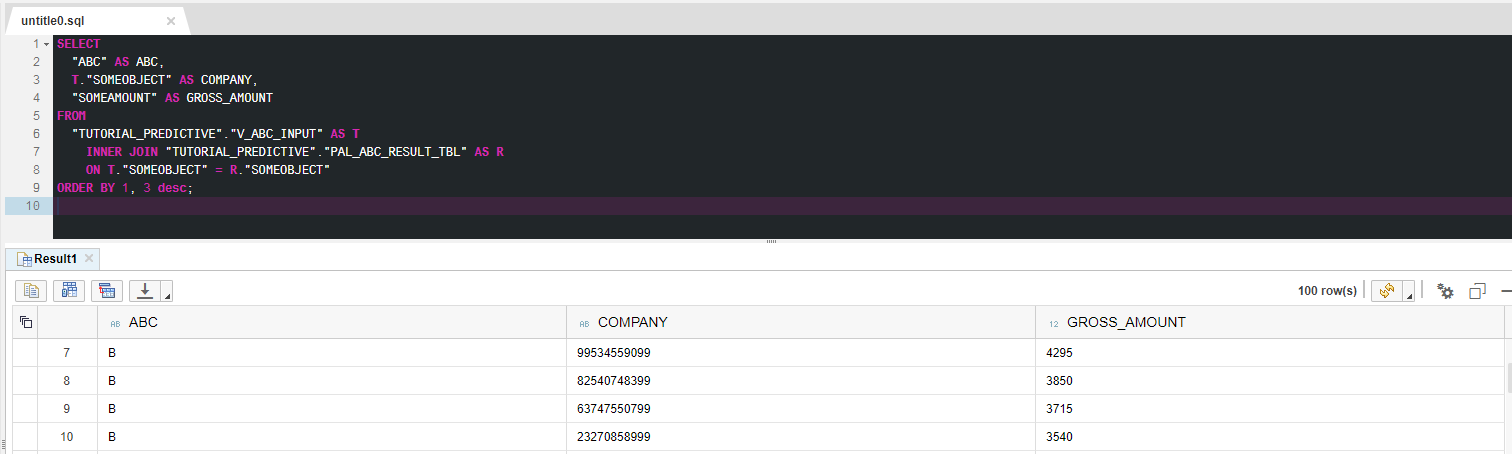
|  |
| --- |
| --Prepare and execute the procedure with ABC Analysis  --Prepare parameters table  DROP TABLE "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_CONTROL\_TBL";  CREATE COLUMN TABLE "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_CONTROL\_TBL"  (  "Name" NVARCHAR(100) NOT NULL,  "intArgs" INTEGER,  "doubleArgs" DECIMAL(15, 2),  "strArgs" NVARCHAR(100),  PRIMARY KEY ("Name")  );  TRUNCATE TABLE "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_CONTROL\_TBL";  INSERT INTO "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_CONTROL\_TBL"  SELECT 'THREAD\_NUMBER',5,null,null from DUMMY UNION ALL  SELECT 'PERCENT\_A',null,0.70,null from DUMMY UNION ALL  SELECT 'PERCENT\_B',null,0.20,null from DUMMY UNION ALL  SELECT 'PERCENT\_C',null,0.10,null from DUMMY;  --Execute the procedure  DROP TABLE "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_RESULT\_TBL";  CREATE COLUMN TABLE "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_RESULT\_TBL"  (  "ABC" NVARCHAR(10) NOT NULL,  "SOMEOBJECT" NVARCHAR(80) NOT NULL,  PRIMARY KEY ("ABC", "SOMEOBJECT")  );  TRUNCATE TABLE "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_RESULT\_TBL";  CALL "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_CALCULATE"  (  "TUTORIAL\_PREDICTIVE"."V\_ABC\_INPUT",  "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_CONTROL\_TBL",  "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_RESULT\_TBL"  ) WITH OVERVIEW; |

Listing

Check the results.

|  |
| --- |
| SELECT  "ABC" AS ABC,  T."SOMEOBJECT" AS COMPANY,  "SOMEAMOUNT" AS GROSS\_AMOUNT  FROM  "TUTORIAL\_PREDICTIVE"."V\_ABC\_INPUT" AS T  INNER JOIN "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_RESULT\_TBL" AS R  ON T."SOMEOBJECT" = R."SOMEOBJECT"  ORDER BY 1, 3 desc; |

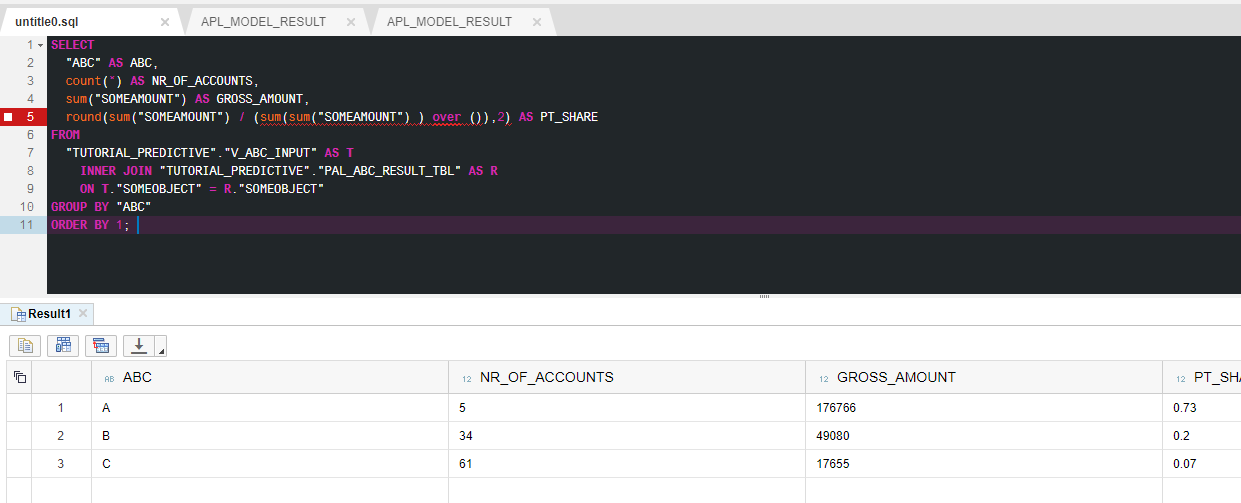
Listing



Compute the percentages of the total revenue falling into each group.

|  |
| --- |
| SELECT  "ABC" AS ABC,  count(\*) AS NR\_OF\_ACCOUNTS,  sum("SOMEAMOUNT") AS GROSS\_AMOUNT,  round(sum("SOMEAMOUNT") / (sum(sum("SOMEAMOUNT") ) over ()),2) AS PT\_SHARE  FROM  "TUTORIAL\_PREDICTIVE"."V\_ABC\_INPUT" AS T  INNER JOIN "TUTORIAL\_PREDICTIVE"."PAL\_ABC\_RESULT\_TBL" AS R  ON T."SOMEOBJECT" = R."SOMEOBJECT"  GROUP BY "ABC"  ORDER BY 1; |

Listing



## Appendix

| **Variable** | **Description** | **Example of values** |
| --- | --- | --- |
| Date | Day, month and year of the readings | A date in the format yyyy-mm-dd such as 1998-01-02 |
| Cash | Cash flow | A numerical value with n decimals |
| BeforeLastMonday  LastMonday  BeforeLastTuesday  LastTuesday  BeforeLastWednesday  LastWednesday  BeforeLastThursday  LastThursday  BeforeLastFriday  LastFriday | Boolean variables that indicate if the information is true or false | 1 if the information is true. |
| Last5WDays  Last4WDays | Boolean variables that indicate if the date is in the 5 or 4 last working days of the month | 1 if the information is true. |
| LastWMonth  BeforeLastWMonth | Boolean variables that indicate if the information is true or false | 1 if the information is true. |
| WorkingDaysIndices  ReverseWorkingDaysIndices | Indices or reverse indices of the working days | An integer value |
| MondayMonthInd  TuesdayMonthInd  WednesdayMonthInd  ThursdayMonthInd  FridayMonthInd | Indices of the week days in the month | An integer value |
| Last5WDaysInd  Last4WDaysInd | Indices of the 5 or 4 last working days of the month | An integer value |