DATA ANALYSIS AND VISUALIZATION IN AWS

Learn how to perform serverless analysis in AWS using visual data analysis tools- AWS Glue DataBrew and Amazon QuickSight. A data analyst will be able to build an end-to-end data analysis tool following this workshop.

Saunak Chandra & Arturo Duarte, AWS PSA Analytics

Problem statement:

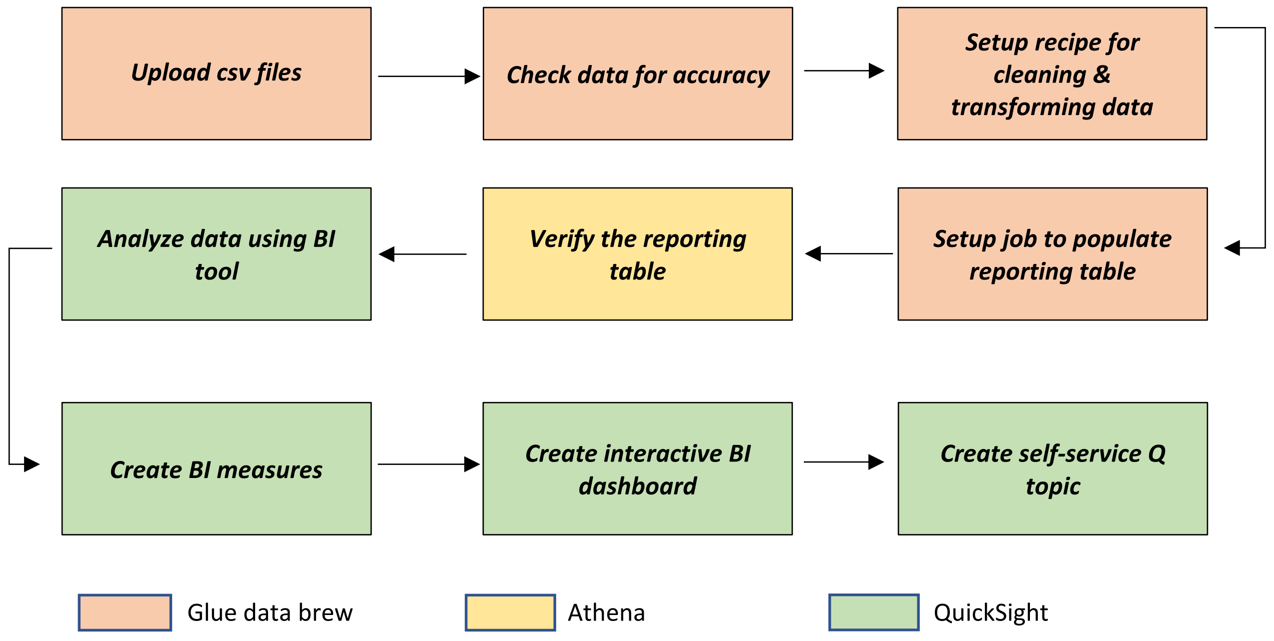
You are a data analyst for Formula 1 business group. You are given several CSV files containing historical race results. Your job is to perform a data analysis- look for any inconsistencies that may exist in the data and provide a self-service BI tool- which a formula 1 enthusiast can ask various questions to get answers from.

The challenge is this data is not ready for data analysis such as 1/ this data is broken into several csv files containing **race, results, constructors, circuits** etc., 2/ there are data quality issues such as unknown values, 3/ some attributes in the data are too granular to perform meaningful analysis; an example is age of the driver.

You are using AWS services to perform these jobs. The objective here is to build a data analysis and visualization tool-

1. That can perform initial analysis of the data to check for consistency and accuracy. In case of missing or invalid data for certain fields you should change those values using mathematical functions such as mean value, most frequent values.
2. Build BI dashboard to interactively explore formula 1 facts, trends and KPIs. The end users also can to ask ad-hoc questions in natural language to get answers from this tool.
3. Setup an end-to-end process to automate data ingestion, data cleaning, data transformation; then populate your final tables with this transformed data and finally refresh the reports.

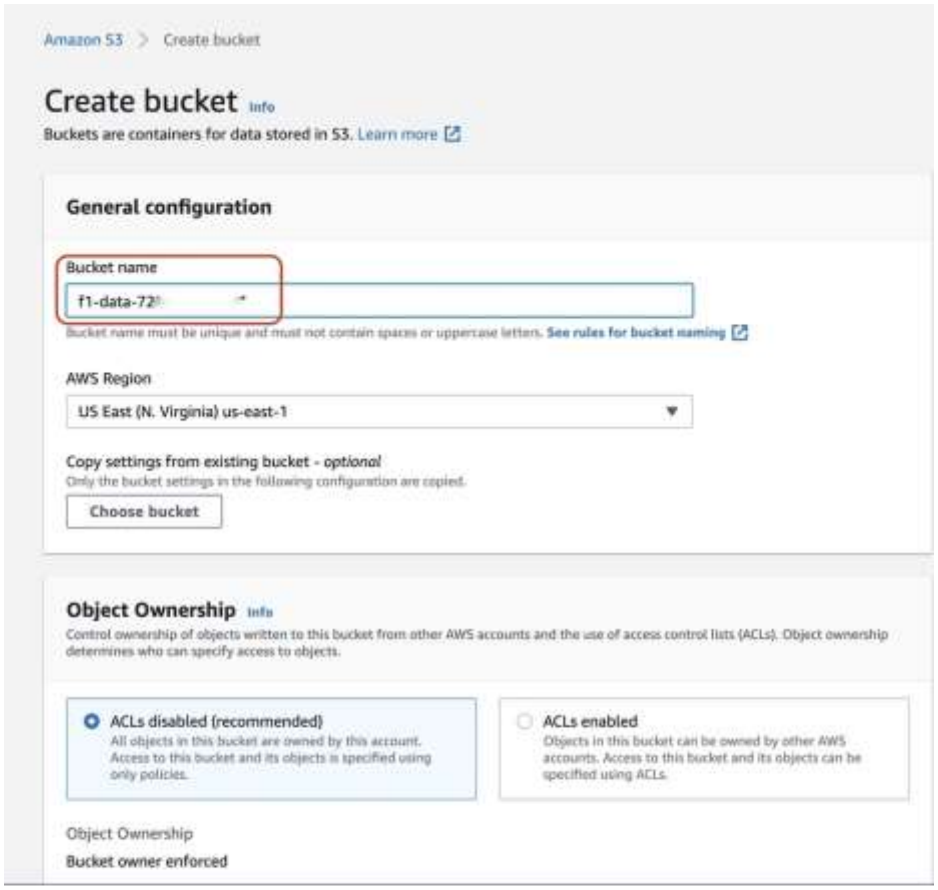
In AWS you can perform the above steps using AWS Glue Databrew, Amazon Athena and Amazon QuickSight. The logical steps to automate the data analysis process is captured in the below diagram-


Upload csv files
Check data for accuracy
Setup recipe for cleaning & transforming data
Setup job to populate reporting table
Verify the reporting table
Analyze data using BI tool
Create BI measures
Create interactive BI dashboard
Create self-service Q topic
Glue data brew
Athena
QuickSight

Pre-requisite:

Create an S3 bucket

1. Create an s3 bucket named ***f1-data-<AWS-account_number>*** to store the CSV files for this workshop.



Amazon S3 > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name

f1-data-72

Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region

US East (N. Virginia) us-east-1

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

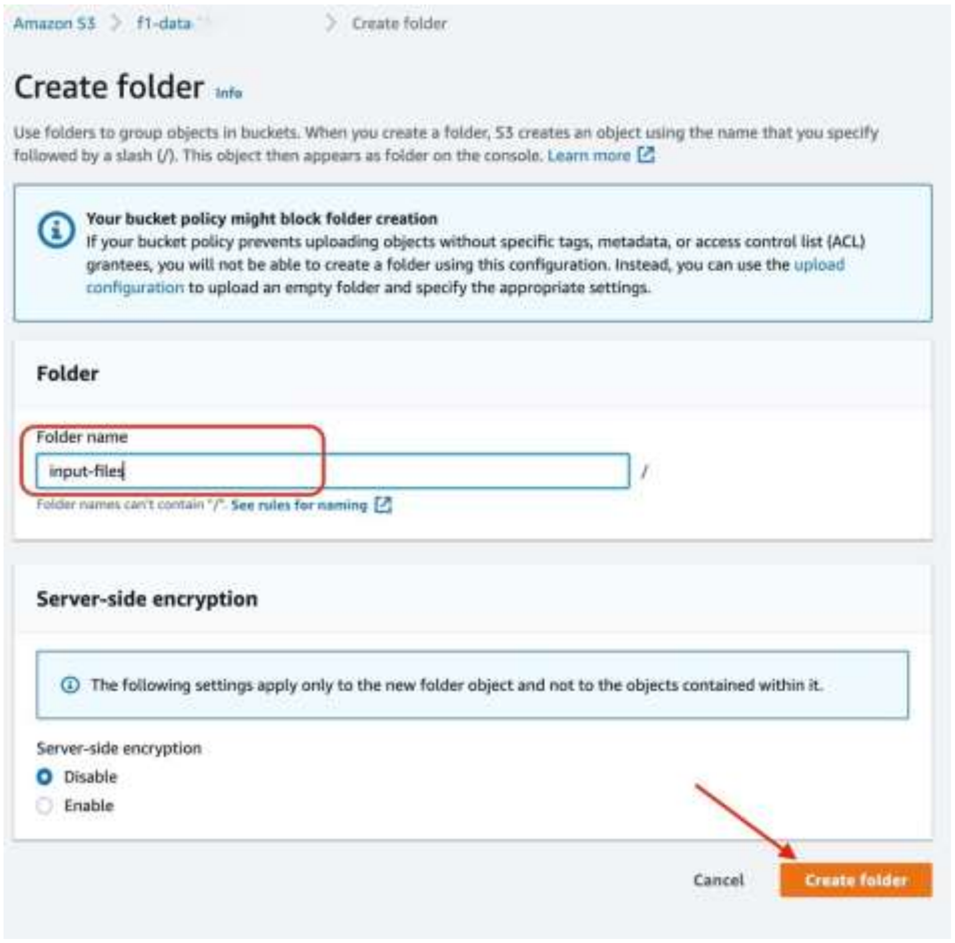
● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced

2. Create an S3 folder ***input-files***.


Amazon S3 > f1-data > Create folder

Create folder Info

Use folders to group objects in buckets. When you create a folder, S3 creates an object using the name that you specify followed by a slash (/). This object then appears as folder on the console. Learn more

Your bucket policy might block folder creation
If your bucket policy prevents uploading objects without specific tags, metadata, or access control list (ACL) grantees, you will not be able to create a folder using this configuration. Instead, you can use the upload configuration to upload an empty folder and specify the appropriate settings.

Folder

Folder name
input-files /
Folder names can't contain "/". See rules for naming

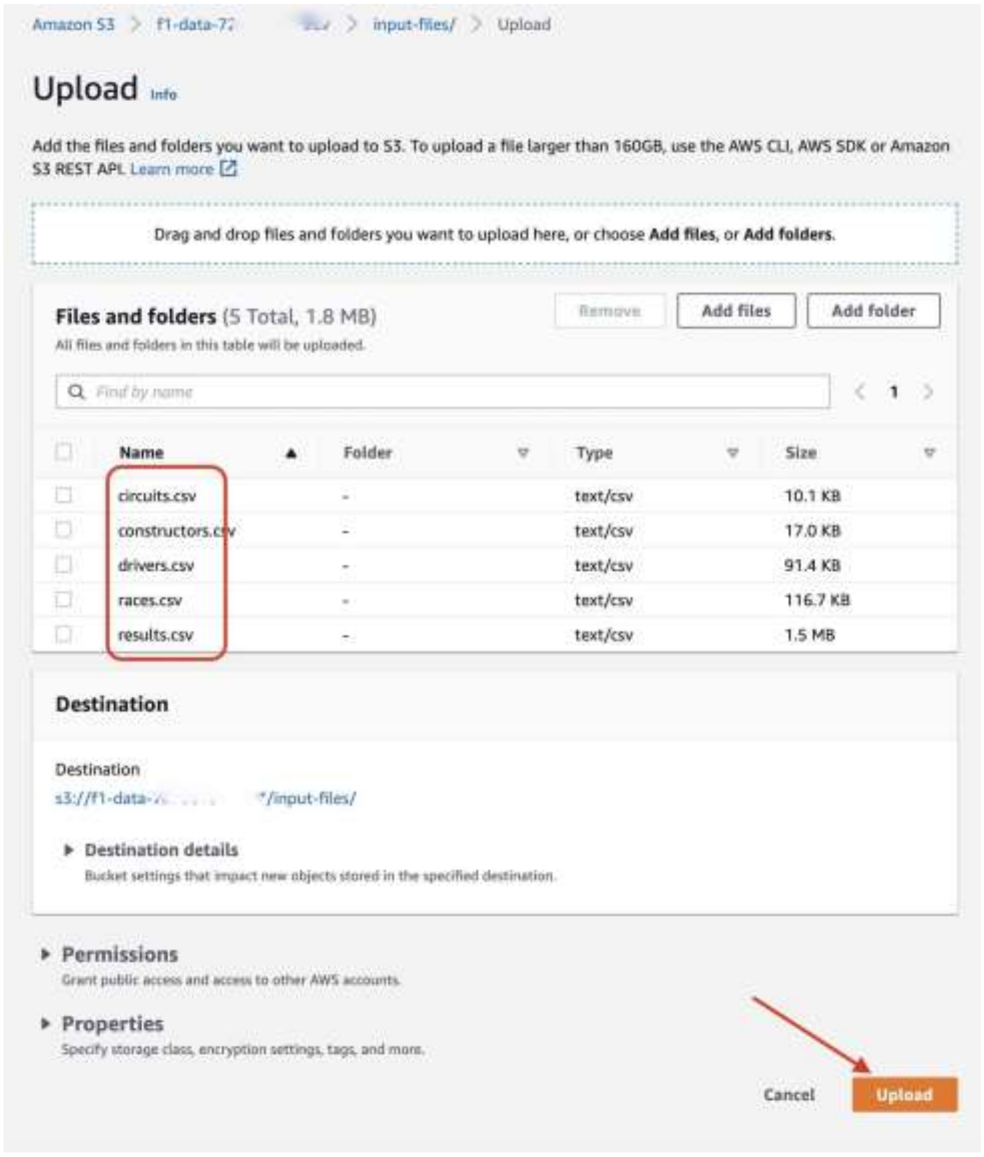
Server-side encryption

The following settings apply only to the new folder object and not to the objects contained within it.

Server-side encryption
[✓ Disable]
Enable

3. Upload the below CSV files into the above S3 folder.

https://awspsa-sampledata.s3.amazonaws.com/f1data/results/results.csv
https://awspsa-sampledata.s3.amazonaws.com/f1data/races/races.csv
https://awspsa-sampledata.s3.amazonaws.com/f1data/constructors/constructors.csv
https://awspsa-sampledata.s3.amazonaws.com/f1data/circuits/circuits.csv
https://awspsa-sampledata.s3.amazonaws.com/f1data/driver/drivers.csv


Amazon S3 > f1-data-7 > input-files/ > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

Files and folders (5 Total, 1.8 MB)

All files and folders in this table will be uploaded.

Name	Folder	Type	Size
circuits.csv	-	text/csv	10.1 KB
constructors.csv	-	text/csv	17.0 KB
drivers.csv	-	text/csv	91.4 KB
races.csv	-	text/csv	116.7 KB
results.csv	-	text/csv	1.5 MB

Destination

Destination
s3://f1-data-/input-files/

Destination details
Bucket settings that impact new objects stored in the specified destination.

Permissions
Grant public access and access to other AWS accounts.

Properties
Specify storage class, encryption settings, tags, and more.

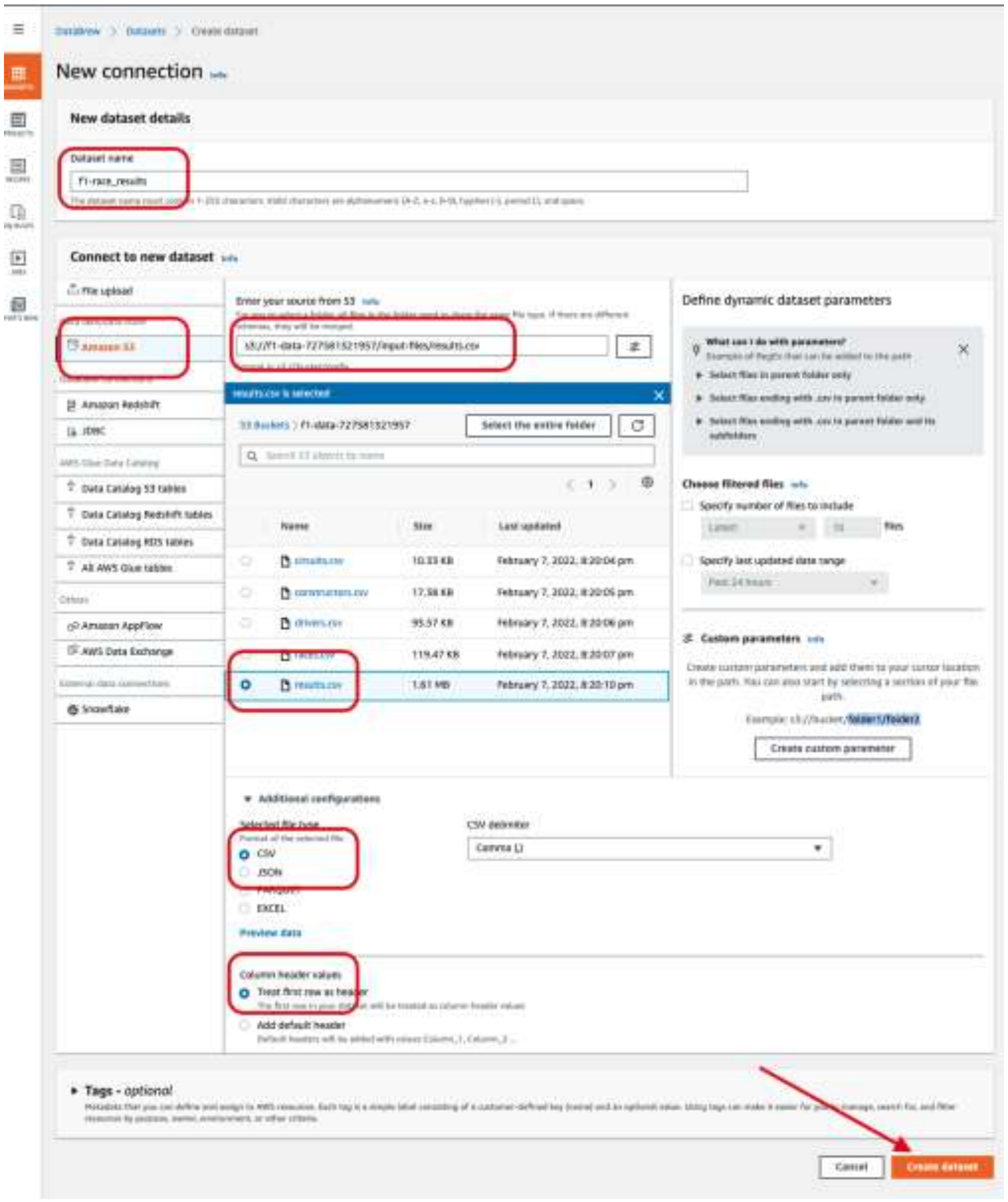
1. Data Preparation

1.1 Create Glue Databrew Datasets

1. Navigate to **AWS Glue DataBrew** service by logging into your AWS account. Select the AWS region of your preference; if none mentioned use US East (N. Virginia).
2. Click **DATASETS** from the left pane. Click on **Connect new dataset**.
3. Enter ***f1-race-results*** as **Dataset name**.
4. Select **Amazon S3** under **Connect to new dataset.**
5. Enter the below S3 path for **Enter your source from S3**.

```
s3://f1-data-<AWS-Account_number>/input-files/results.csv
```

6. Scroll down the page until the **Preview data** is visible. Note: you may need to expand the **Additional configurations** if Preview data is not visible.
7. Click **Create dataset** in the New dataset details page.



New connection

New dataset details

Dataset name: f1-race_results

Connect to new dataset

Enter your source from S3: s3://f1-data-727581321957/input-files/results.csv

results.csv is selected

S3 Buckets > f1-data-727581321957

	Name	Size	Last updated
	circuits.csv	10.33 KB	February 7, 2022, 8:20:04 pm
	constructors.csv	17.58 KB	February 7, 2022, 8:20:05 pm
	drivers.csv	95.57 KB	February 7, 2022, 8:20:06 pm
	races.csv	119.47 KB	February 7, 2022, 8:20:07 pm
●	results.csv	1.61 MB	February 7, 2022, 8:20:10 pm

Additional configurations

Selected file type: [✓ CSV] JSON PARQUET EXCEL

CSV delimiter: Comma (,)

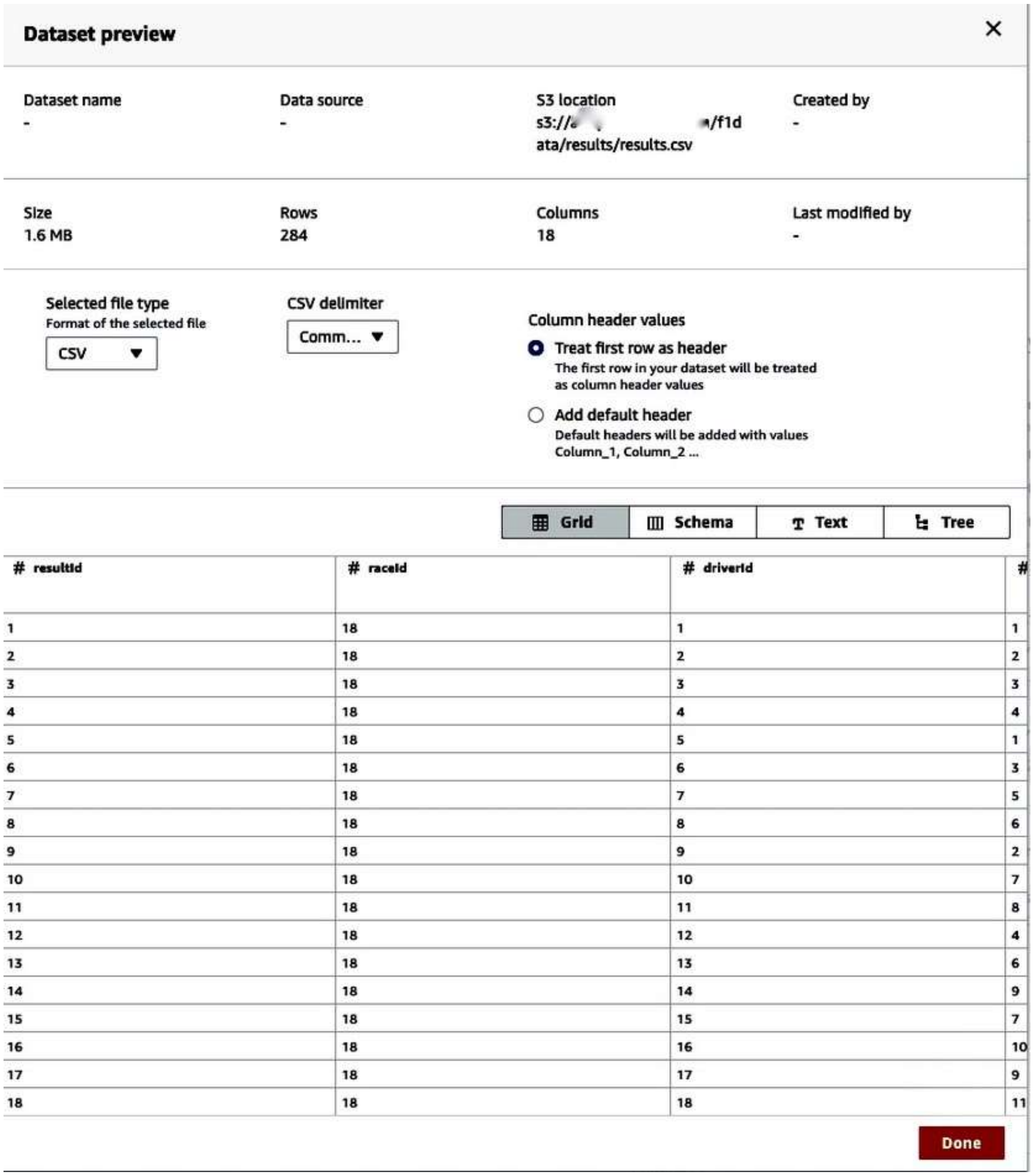
Preview data

Column header values: [✓ Treat first row as header] Add default header

8. (Optional)

Click on Preview data and examine the various fields in the f1-race-results dataset in the Grid format. Make yourself aware of the fields available. Note the below-

I. There are various id fields- *resultId, raceId, driverId, constructorId*
II. Numerical field names are prefixed with “#” and text field names with “ABC”.
III. Click Done to close the preview screen.



Dataset preview

Dataset name	Data source	S3 location	Created by
-	-	s3://a[redacted]a/f1data/results/results.csv	-

Size	Rows	Columns	Last modified by
1.6 MB	284	18	-

Selected file type
Format of the selected file
CSV

CSV delimiter
Comm...

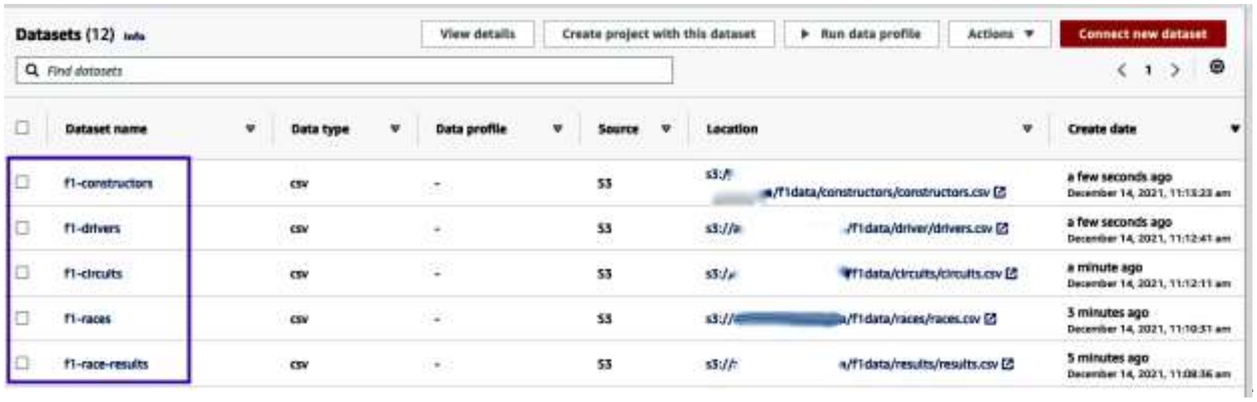
Column header values
[✓ Treat first row as header]
The first row in your dataset will be treated as column header values
Add default header
Default headers will be added with values Column_1, Column_2 ...

# resultId	# raceId	# driverId	#
1	18	1	1
2	18	2	2
3	18	3	3
4	18	4	4
5	18	5	1
6	18	6	3
7	18	7	5
8	18	8	6
9	18	9	2
10	18	10	7
11	18	11	8
12	18	12	4
13	18	13	6
14	18	14	9
15	18	15	7
16	18	16	10
17	18	17	9
18	18	18	11

Similarly create datasets- ***f1-races, f1-circuits, f1-drivers and f1-constructors.*** The S3 locations for these datasets are

- f1-races
- f1-circuits
- f1-drivers
- f1-constructors

At the end of these steps you have 5 datasets created as shown below



Dataset name	Data type	Data profile	Source	Location	Create date
f1-constructors	csv	-	S3	s3://[redacted]/f1data/constructors/constructors.csv	a few seconds ago December 14, 2021, 11:13:23 am
f1-drivers	csv	-	S3	s3://[redacted]/f1data/driver/drivers.csv	a few seconds ago December 14, 2021, 11:12:41 am
f1-circuits	csv	-	S3	s3://[redacted]f1data/circuits/circuits.csv	a minute ago December 14, 2021, 11:12:11 am
f1-races	csv	-	S3	s3://[redacted]/f1data/races/races.csv	3 minutes ago December 14, 2021, 11:10:31 am
f1-race-results	csv	-	S3	s3://[redacted]/f1data/results/results.csv	5 minutes ago December 14, 2021, 11:08:36 am

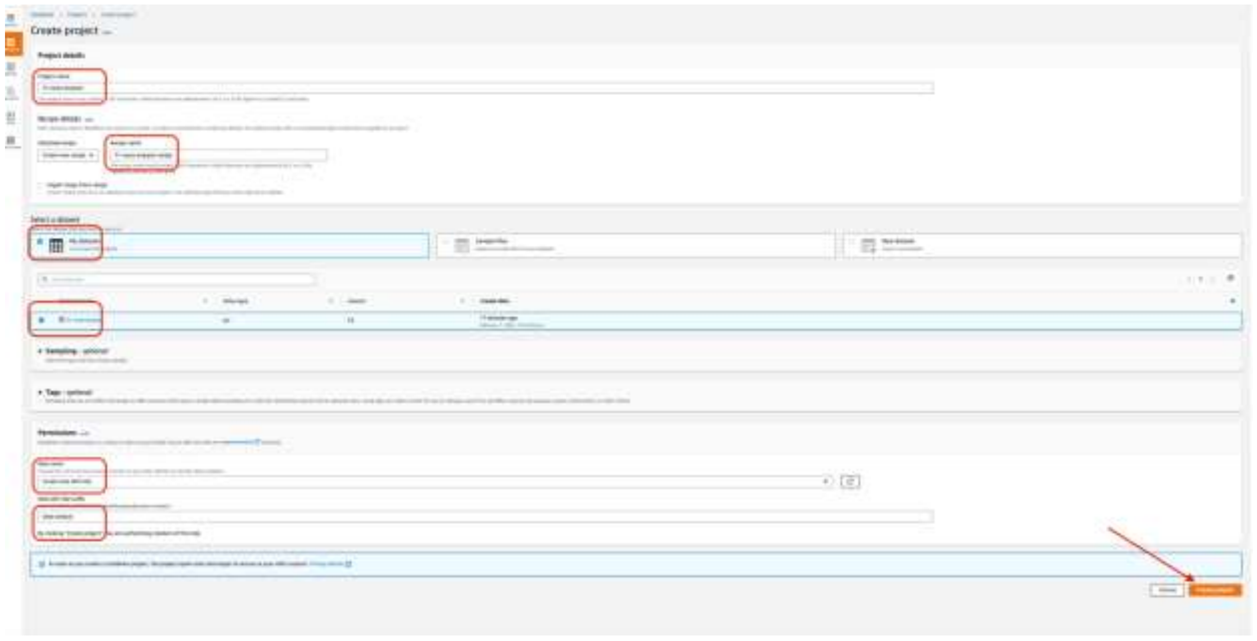
1.2 Create Glue Databrew Project

Now the datasets are uploaded you are ready to check the data for accuracies and then perform data cleaning and transformation. This is essential step before your proceed further for BI analysis as any inaccuracies that may exist in the data may provide inconsistent view of the reports.

Data cleaning and transformation is done in Glue DataBrew project. A Glue DataBrew project is the workspace for data analysis and transformation. When you create a project in Glue DataBrew, it downloads a sample of your dataset(s) and provide you a ways to define steps for cleaning and transforming data, preview transformed data and finally create job that applies the recipe steps on the entire dataset.

In this exercise we will create ***f1-races-analysis*** project by connecting to the datasets we created earlier.

1. Click **PROJECTS** from the left pane. Click on **Connect project**.
2. Enter ***f1-races-analysis*** as **Project name**.
3. Check a new recipe is getting created as ***f1-races-analysis-recipe.***
4. Under select a dataset click **My datasets** > **f1-race-results.**
5. Scroll down until Permissions. Under Role name select option **Create new IAM role.**
6. Enter ***data-analyst*** as **New IAM role suffix.**
7. Hit Create project.

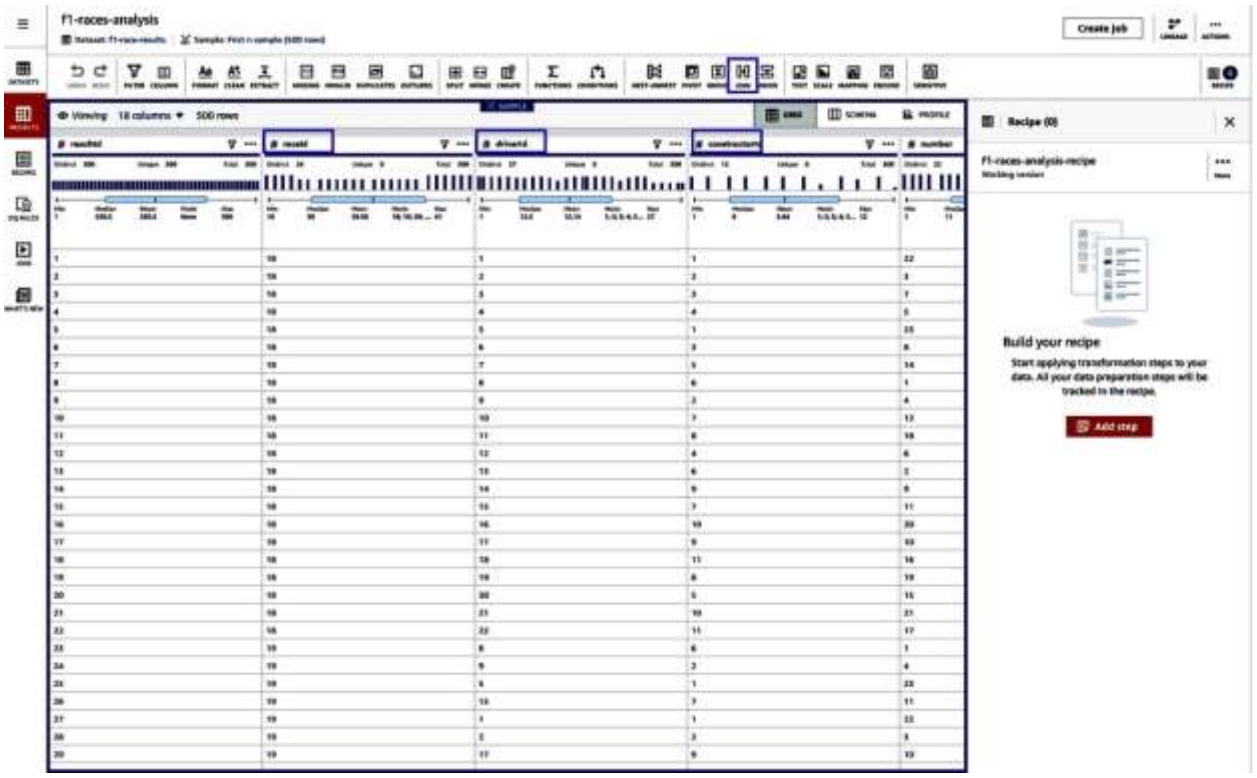


1.3 Working in Glue Databrew Workspace

Now a sample data is loaded into you project workspace you are ready to perform data analysis and transformation. First note that the ***f1-race-results*** dataset alone have missing details on race, drivers, constructors and circuits. We need to join this with the rest of the datasets.

1.3.1 Join datasets

I. In the project workspace identify **JOIN** icon which is located between GROUP and UNION. It will look as two vertical bars and an arrow pointing to the left bar from the right bar.


f1-races-analysis

Dataset: f1-race-results | Sample: First n sample (500 rows)

Viewing 18 columns ▾ 500 rows

# resultId	# raceId	# driverId	# constructorId	# number
1	18	1	1	22
2	18	2	2	3
3	18	3	3	7
4	18	4	4	5
5	18	5	1	23
6	18	6	3	8
7	18	7	5	14
8	18	8	6	1
9	18	9	2	4
10	18	10	7	12
11	18	11	8	18
12	18	12	4	6
13	18	13	6	2
14	18	14	5	9
15	18	15	7	11
16	18	16	10	20
17	18	17	9	10
18	18	18	11	16
19	18	19	8	19
20	18	20	5	15
21	18	21	10	21
22	18	22	11	17
23	19	8	6	1
24	19	9	2	4
25	19	5	1	23
26	19	15	7	11
27	19	1	1	22
28	19	2	2	3
29	19	17	9	10

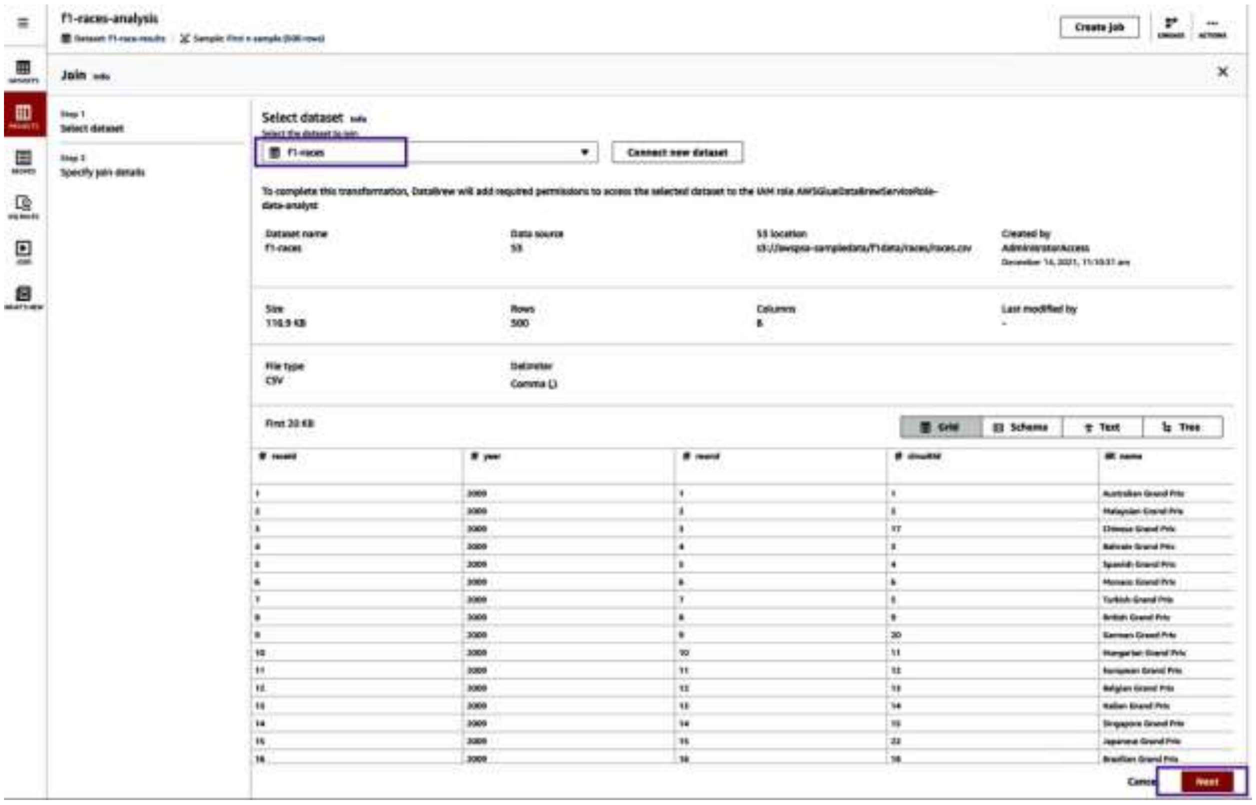
Recipe (0)

f1-races-analysis-recipe
Working version

Build your recipe

Start applying transformation steps to your data. All your data preparation steps will be tracked in the recipe.

II. Select ***f1-races*** from the **Select dataset** dropdown. Hit **Next**.


f1-races-analysis

Dataset: f1-race-results | Sample: First n sample (500 rows)

Join

Step 1
Select dataset

Step 2
Specify join details

Select dataset

Select the dataset to join

f1-races

To complete this transformation, DataBrew will add required permissions to access the selected dataset to the IAM role AWSGlueDataBrewServiceRole-data-analyst

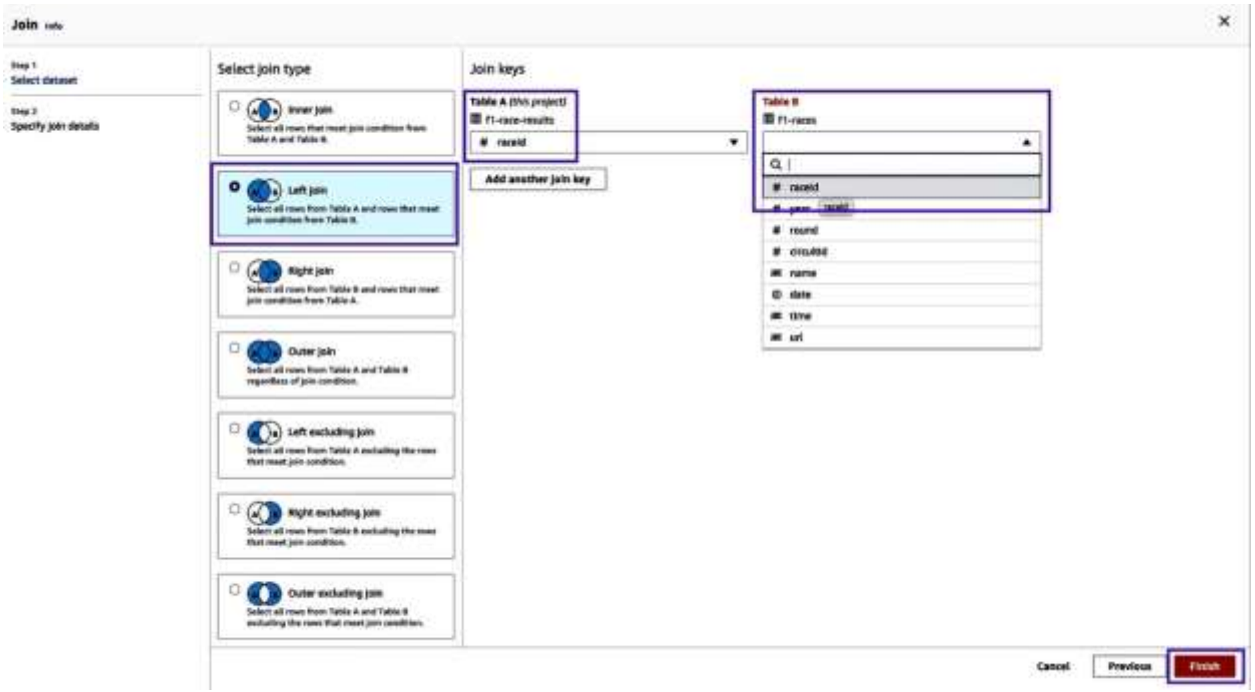
Dataset name f1-races	Data source S3	S3 location s3://awspsa-sampledata/f1data/races/races.csv	Created by AdministratorAccess December 14, 2021, 11:10:37 am
Size 116.9 KB	Rows 500	Columns 8	Last modified by -
File type CSV	Delimiter Comma (,)		

First 20 KB

# raceId	# year	# round	# circuitId	name
1	2009	1	1	Australian Grand Prix
2	2009	2	2	Malaysian Grand Prix
3	2009	3	17	Chinese Grand Prix
4	2009	4	3	Bahrain Grand Prix
5	2009	5	4	Spanish Grand Prix
6	2009	6	6	Monaco Grand Prix
7	2009	7	5	Turkish Grand Prix
8	2009	8	9	British Grand Prix
9	2009	9	20	German Grand Prix
10	2009	10	11	Hungarian Grand Prix
11	2009	11	12	European Grand Prix
12	2009	12	13	Belgian Grand Prix
13	2009	13	14	Italian Grand Prix
14	2009	14	15	Singapore Grand Prix
15	2009	15	22	Japanese Grand Prix
16	2009	16	18	Brazilian Grand Prix

III. This will open up the join defining screen. Click **Left join** as the join type. For the Join keys select the below keys-

Table A f1-race-results: raceId = Table B f1-races: raceId



IV. Click **Joined table preview**. Note that the fields from ***f1-races*** dataset are included in the joined dataset. Hit **Finish**.

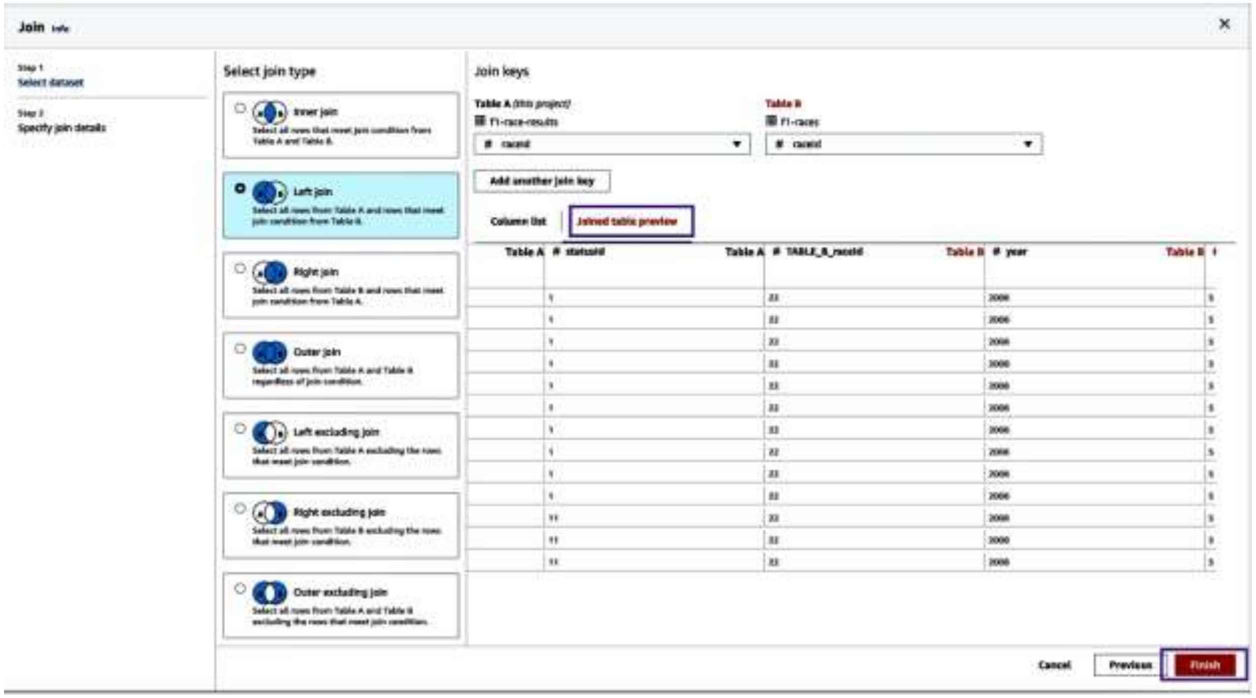
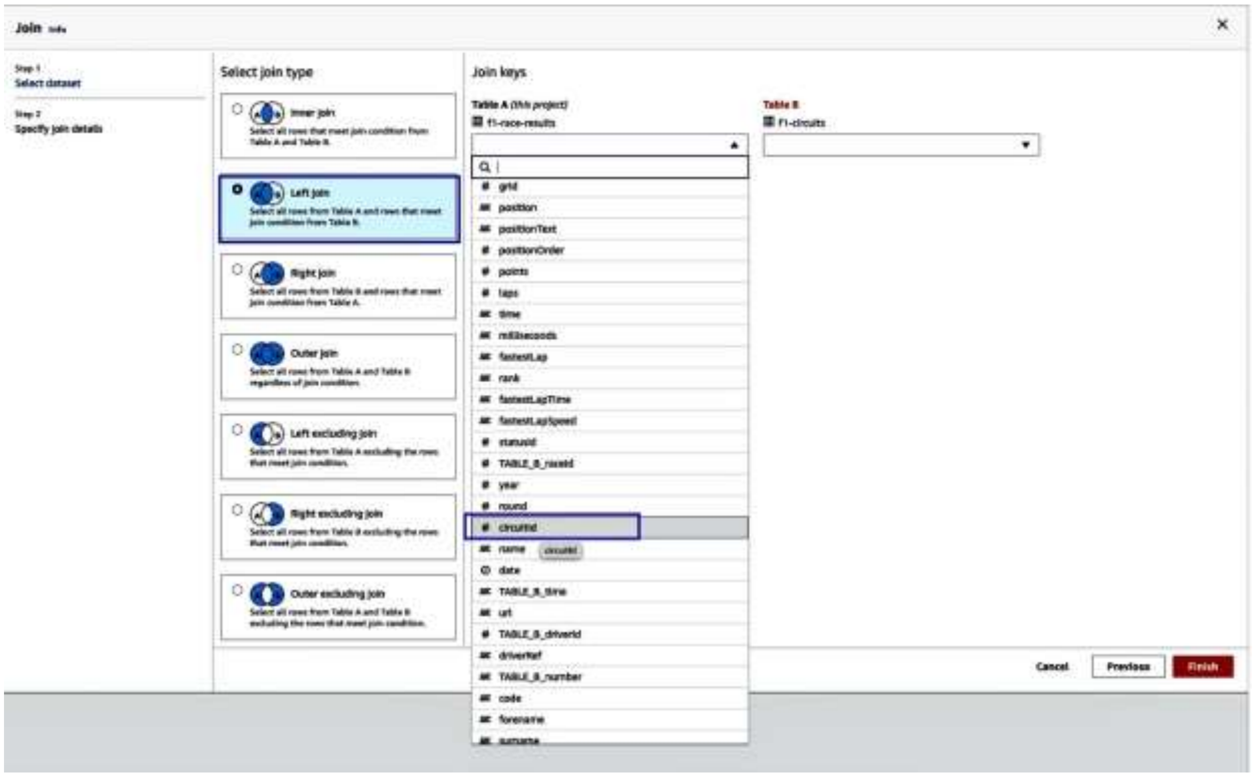


Table A # statusId	Table A # TABLE_A_raceId	Table B # year	Table B #
1	22	2008	5
1	22	2008	5
1	22	2008	5
1	22	2008	5
1	22	2008	5
1	22	2008	5
1	22	2008	5
1	22	2008	5
1	22	2008	5
1	22	2008	5
11	22	2008	5
11	22	2008	5
11	22	2008	5

V. Apply the similar joins for ***f1-circuits***, ***f1-drivers***, ***f1-constructors***. When you define the join make sure the Table A always shows ***f1-race-results***.


Join Info

Step 1
Select dataset

Step 2
Specify join details

Select join type

- Inner join: Select all rows that meet join condition from Table A and Table B.
- [✓ Left join]: Select all rows from Table A and rows that meet join condition from Table B.
- Right join: Select all rows from Table B and rows that meet join condition from Table A.
- Outer join: Select all rows from Table A and Table B regardless of join condition.
- Left excluding join: Select all rows from Table A excluding the rows that meet join condition.
- Right excluding join: Select all rows from Table B excluding the rows that meet join condition.
- Outer excluding join: Select all rows from Table A and Table B excluding the rows that meet join condition.

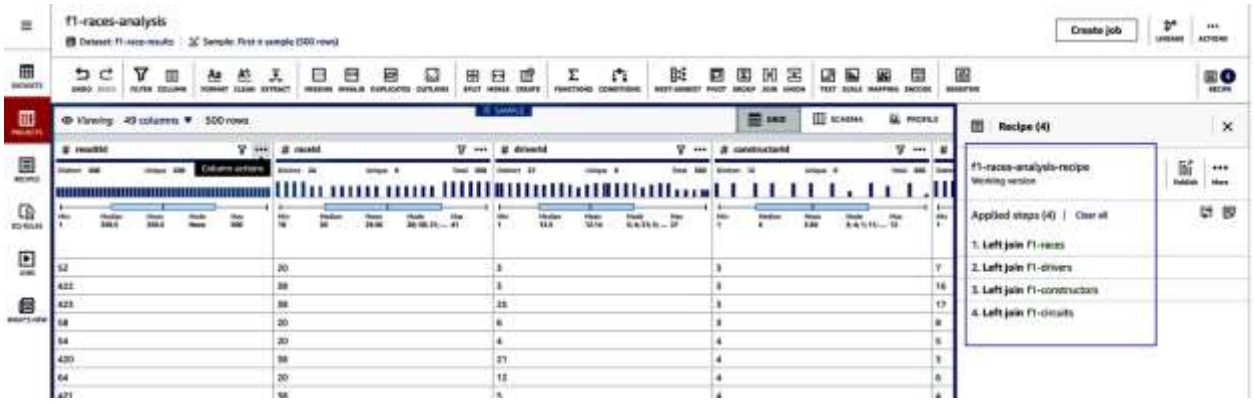
Join keys

Table A (this project): f1-race-results

Table B: f1-circuits

grid, position, positionText, positionOrder, points, laps, time, milliseconds, fastestLap, rank, fastestLapTime, fastestLapSpeed, statusId, TABLE_B_raceId, year, round, circuitId, name, date, TABLE_B_time, url, TABLE_B_driverId, driverRef, TABLE_B_number, code, forename, surname

VI. After the 3 joins defined you can verify the recipe steps have appeared.


f1-races-analysis

Recipe (4)

f1-races-analysis-recipe
Working version

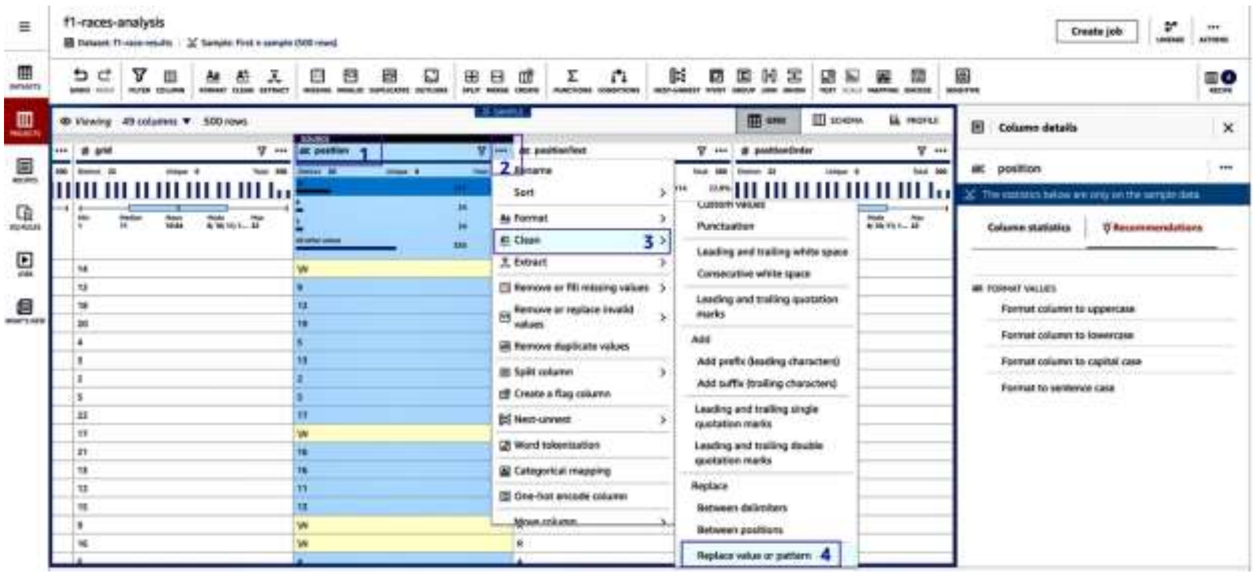
Applied steps (4) | Clear all

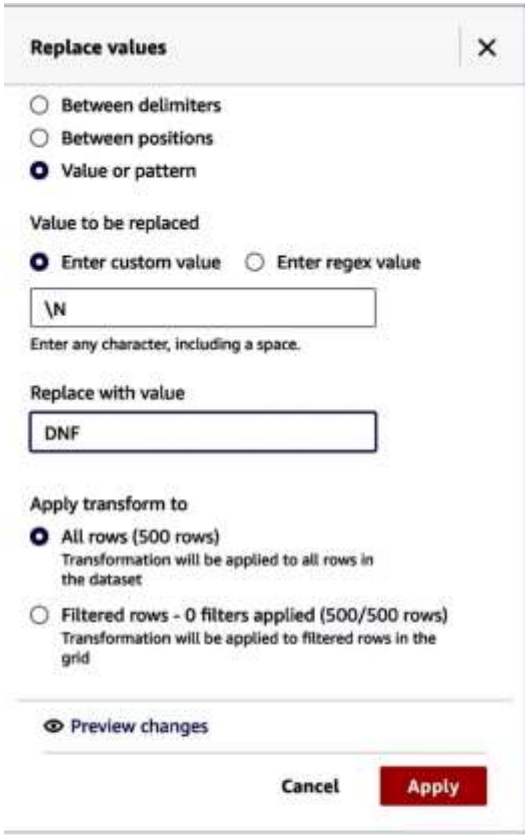
1. Left join f1-races
2. Left join f1-drivers
3. Left join f1-constructors
4. Left join f1-circuits

At the end of these steps you have created a joined table in the project workspace combining data from 5 datasets. This table is logical since you have not selected a location where this table will be created. We will do that at later stage.

1.3.2 Clean data

Now let's inspect for any data issues. As we scroll towards right we find the ***position*** field has values marked as "\N". These are drivers who could not finish the race. Let's replace them as more subtle "DNF" (Did Not Finish) value. This is also evident from the ***positionText*** field which is marked as "R as retired.




Replace values

- ○ Between delimiters
- ○ Between positions
- [✓ Value or pattern]

Value to be replaced

[✓ Enter custom value] ○ Enter regex value

\N

Enter any character, including a space.

Replace with value

DNF

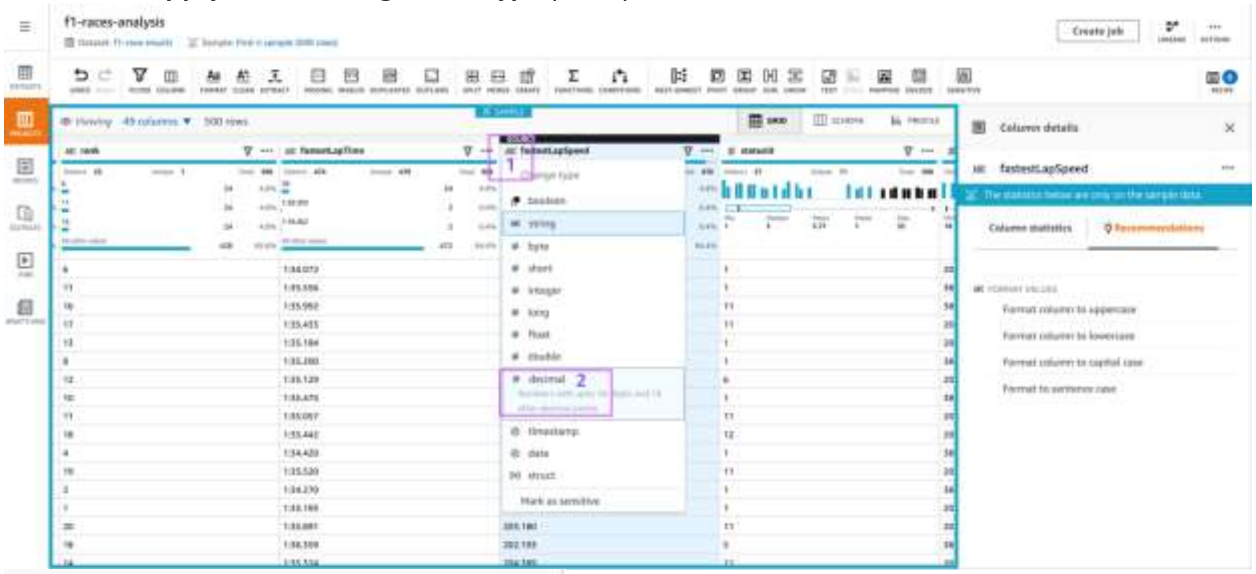
Apply transform to

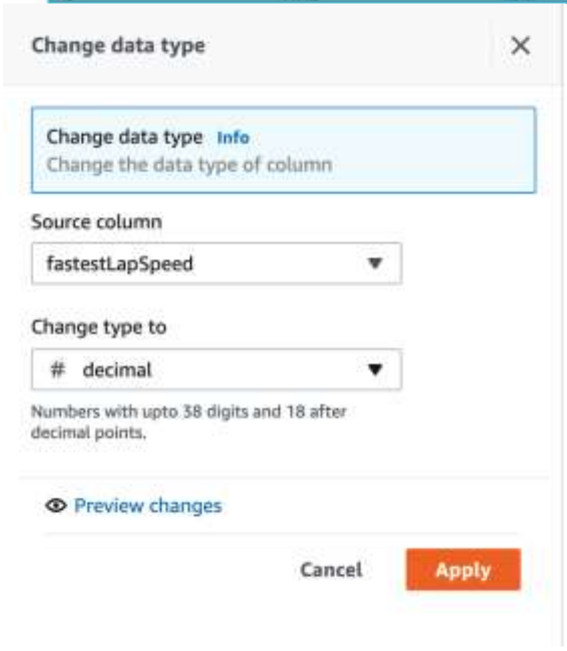
- [✓ All rows (500 rows)]
  Transformation will be applied to all rows in the dataset
- ○ Filtered rows - 0 filters applied (500/500 rows)
  Transformation will be applied to filtered rows in the grid

1.3.3 Change Data type

Further scrolling down will reveal that the ***fastestLapSpeed*** field has been identified as String data type by Glue DataBrew. We will change that to Decimal field.

1. Click on “ABC” data type left of the field name ***fastestLapSpeed.***
2. Select **decimal**.
3. Hit **Apply** in the **Change data type** prompt.




Change data type

Change data type Info
Change the data type of column

Source column
fastestLapSpeed

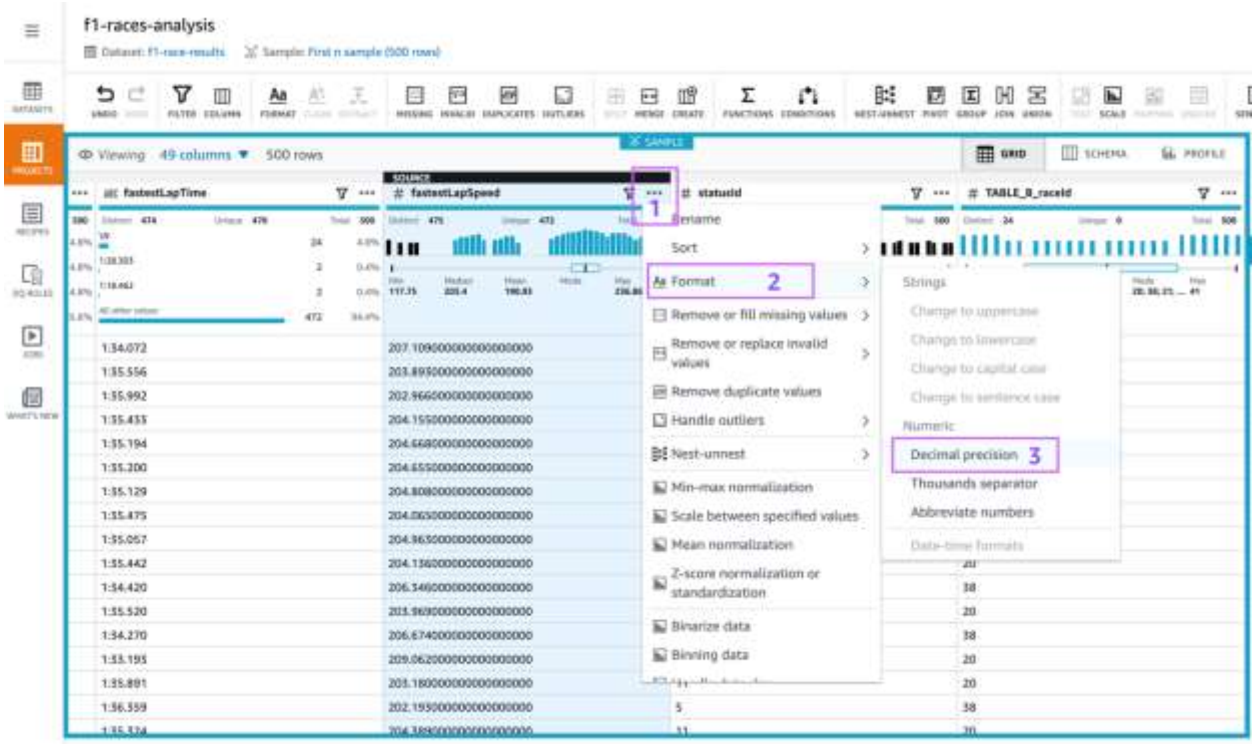
Change type to
# decimal

Numbers with upto 38 digits and 18 after decimal points.

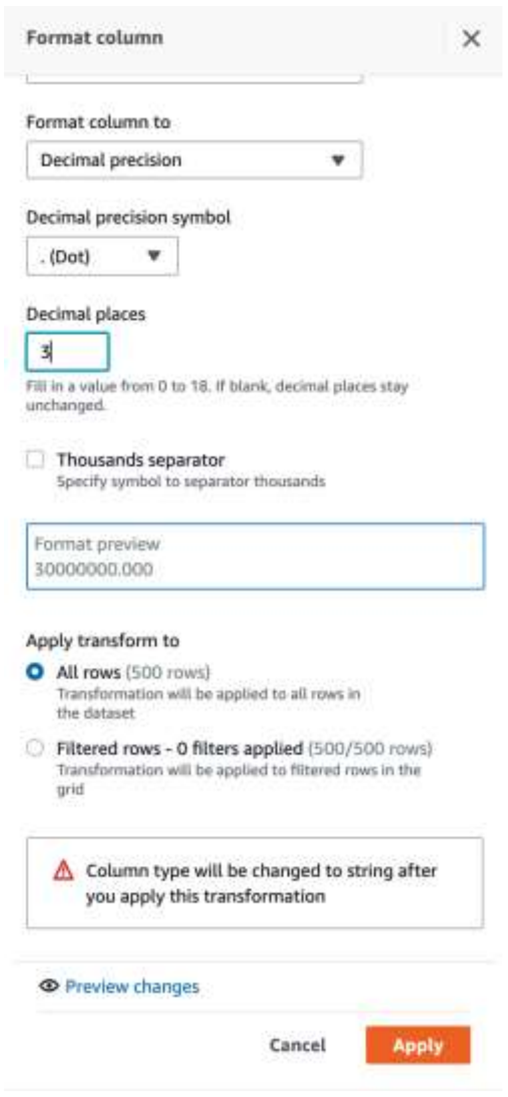
Preview changes

Cancel Apply

4. After changing the decimal data type the fields is overflowed with spurious 0’s upto 18 decimal places. This is not helpful to view results and storing the field values. We will change the decimal precision to 3.



fastestLapTime	fastestLapSpeed
1:34.072	207.109000000000000000
1:35.556	203.893000000000000000
1:35.992	202.966000000000000000
1:35.433	204.155000000000000000
1:35.194	204.668000000000000000
1:35.200	204.655000000000000000
1:35.129	204.808000000000000000
1:35.475	204.065000000000000000
1:35.057	204.963000000000000000
1:35.442	204.136000000000000000
1:34.420	206.346000000000000000
1:35.520	203.969000000000000000
1:34.270	206.674000000000000000
1:33.193	209.062000000000000000
1:35.891	203.180000000000000000
1:36.359	202.193000000000000000
1:35.324	204.389000000000000000


Format column

Format column to
Decimal precision

Decimal precision symbol
. (Dot)

Decimal places
3
Fill in a value from 0 to 18. If blank, decimal places stay unchanged.

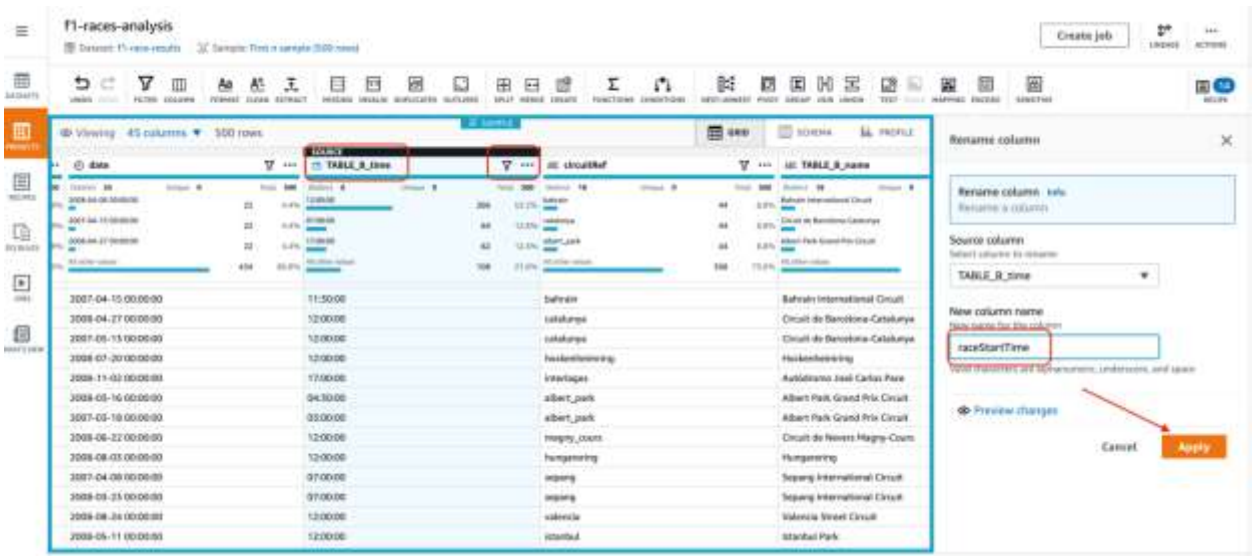
☐ Thousands separator
Specify symbol to separator thousands

Format preview
30000000.000

Apply transform to
● All rows (500 rows)
Transformation will be applied to all rows in the dataset
○ Filtered rows - 0 filters applied (500/500 rows)
Transformation will be applied to filtered rows in the grid

Column type will be changed to string after you apply this transformation

1.3.4 Split Column



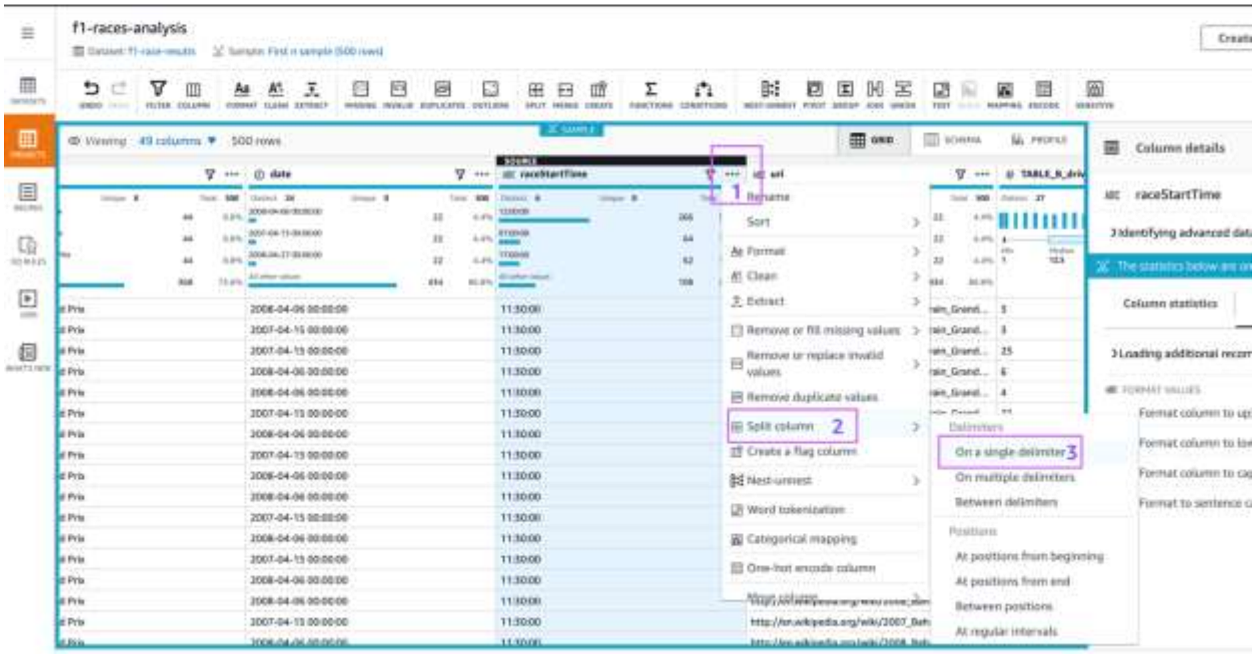
Rename column

Rename column Info
Rename a column

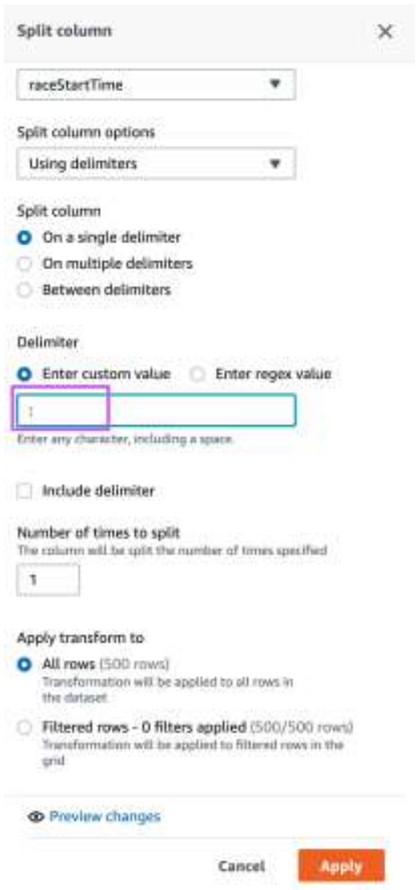
Source column
Select column to rename
TABLE_R_time

New column name
New name for the column
raceStartTime

date	TABLE_R_time	circuitRef	TABLE_R_name
2007-04-15 00:00:00	11:50:00	bahrain	Bahrain International Circuit
2008-04-27 00:00:00	12:00:00	catalunya	Circuit de Barcelona-Catalunya
2007-05-13 00:00:00	12:00:00	catalunya	Circuit de Barcelona-Catalunya
2008-07-20 00:00:00	12:00:00	hockenheimring	Hockenheimring
2008-11-02 00:00:00	17:00:00	interlagos	Autódromo José Carlos Pace
2008-03-16 00:00:00	04:30:00	albert_park	Albert Park Grand Prix Circuit
2007-03-18 00:00:00	03:00:00	albert_park	Albert Park Grand Prix Circuit
2008-06-22 00:00:00	12:00:00	magny_cours	Circuit de Nevers Magny-Cours
2008-08-03 00:00:00	12:00:00	hungaroring	Hungaroring
2007-04-08 00:00:00	07:00:00	sepang	Sepang International Circuit
2008-03-23 00:00:00	07:00:00	sepang	Sepang International Circuit
2008-08-24 00:00:00	12:00:00	valencia	Valencia Street Circuit
2008-05-11 00:00:00	12:00:00	istanbul	Istanbul Park



date	raceStartTime
2008-04-06 00:00:00	11:30:00
2007-04-15 00:00:00	11:30:00
2007-04-15 00:00:00	11:30:00
2008-04-06 00:00:00	11:30:00
2008-04-06 00:00:00	11:30:00
2007-04-15 00:00:00	11:30:00
2008-04-06 00:00:00	11:30:00
2007-04-15 00:00:00	11:30:00
2008-04-06 00:00:00	11:30:00
2008-04-06 00:00:00	11:30:00
2007-04-15 00:00:00	11:30:00
2008-04-06 00:00:00	11:30:00
2007-04-15 00:00:00	11:30:00
2008-04-06 00:00:00	11:30:00
2008-04-06 00:00:00	11:30:00
2007-04-15 00:00:00	11:30:00



Split column

raceStartTime

Split column options

Using delimiters

Split column

[✓ On a single delimiter]
On multiple delimiters
Between delimiters

Delimiter

[✓ Enter custom value] Enter regex value

:

Enter any character, including a space.

Include delimiter

Number of times to split

The column will be split the number of times specified

1

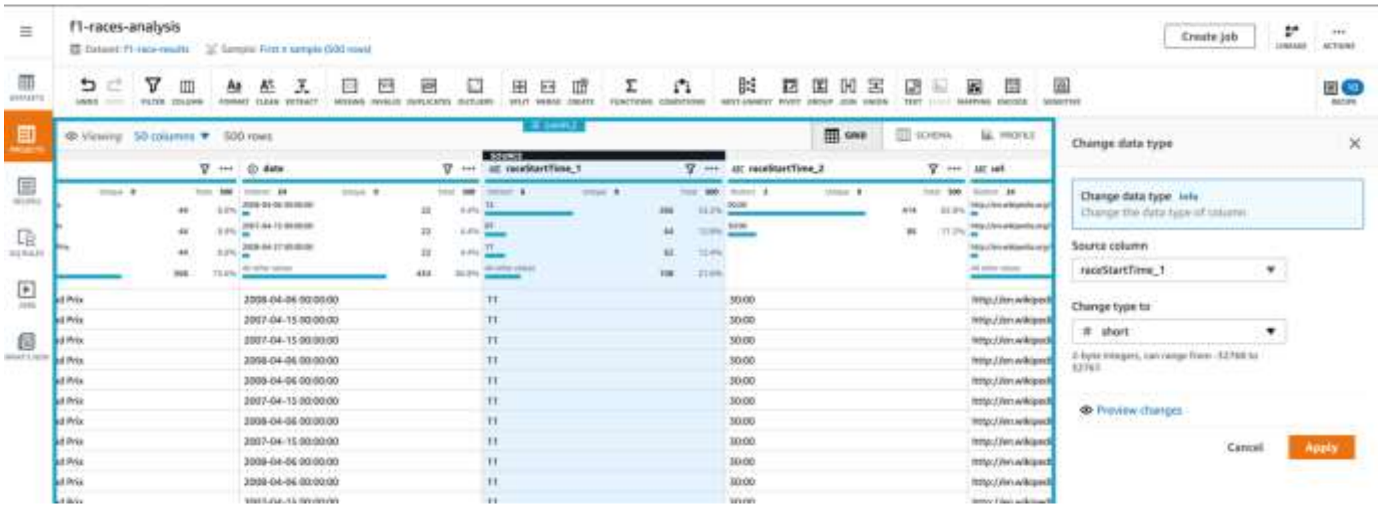
Apply transform to

[✓ All rows (500 rows)]
Transformation will be applied to all rows in the dataset

Filtered rows - 0 filters applied (500/500 rows)
Transformation will be applied to filtered rows in the grid

Preview changes

Cancel Apply



f1-races-analysis

Viewing 50 columns 500 rows

date	raceStartTime_1	raceStartTime_2	url
2008-04-06 00:00:00	11	30:00	http://en.wikipedi
2007-04-15 00:00:00	11	30:00	http://en.wikipedi
2007-04-15 00:00:00	11	30:00	http://en.wikipedi
2008-04-06 00:00:00	11	30:00	http://en.wikipedi
2008-04-06 00:00:00	11	30:00	http://en.wikipedi
2007-04-15 00:00:00	11	30:00	http://en.wikipedi
2008-04-06 00:00:00	11	30:00	http://en.wikipedi
2007-04-15 00:00:00	11	30:00	http://en.wikipedi
2008-04-06 00:00:00	11	30:00	http://en.wikipedi
2008-04-06 00:00:00	11	30:00	http://en.wikipedi

Change data type

Change data type Info

Change the data type of column

Source column

raceStartTime_1

Change type to

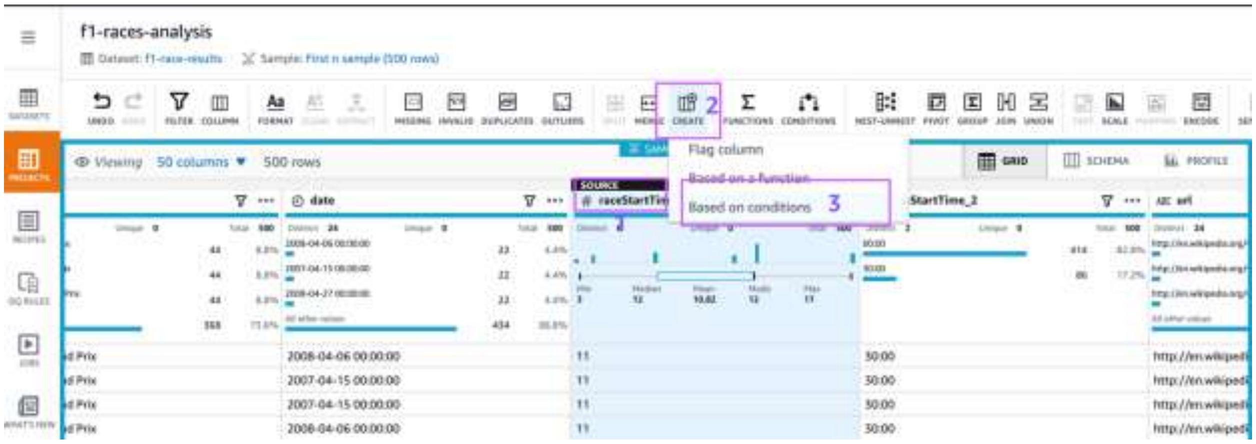
short

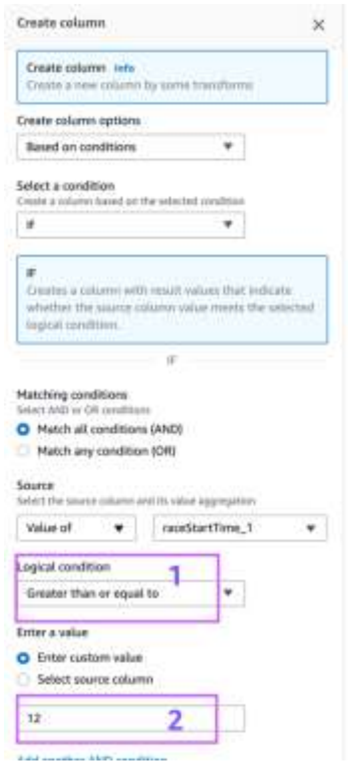
2-byte integers, can range from -32768 to 32767

Preview changes

Cancel Apply

1.3.5 Create Column





Create column

Create column Info
Create a new column by some transforms

Create column options: Based on conditions

Select a condition
Create a column based on the selected condition
If

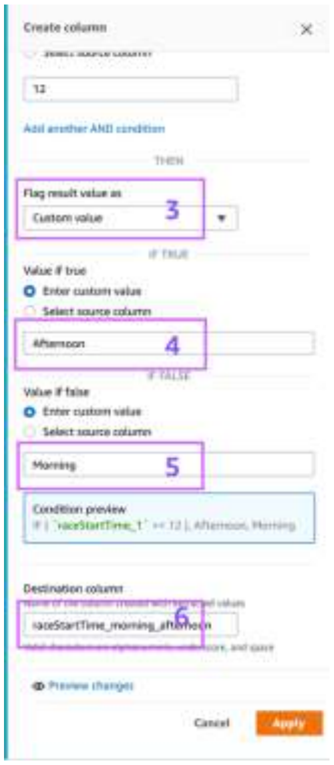
IF
Creates a column with result values that indicate whether the source column value meets the selected logical condition.

Matching conditions
Select AND or OR conditions
[✓ Match all conditions (AND)]
Match any condition (OR)

Source
Select the source column and its value aggregation
Value of | raceStartTime_1

Logical condition: Greater than or equal to

Enter a value
[✓ Enter custom value]
Select source column
12



Create column

12

Add another AND condition

THEN

Flag result value as: Custom value

IF TRUE
Value if true
[✓ Enter custom value]
Select source column
Afternoon

IF FALSE
Value if false
[✓ Enter custom value]
Select source column
Morning

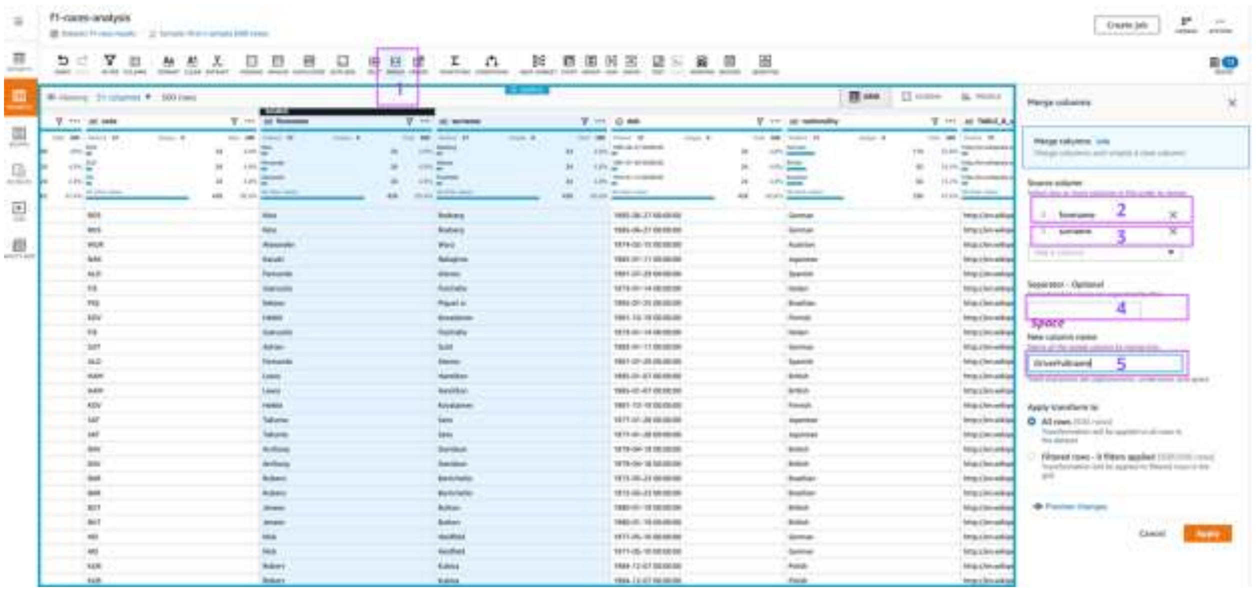
Condition preview
If (`raceStartTime_1` >= 12), Afternoon, Morning

Destination column
raceStartTime_morning_afternoon

Preview changes
Cancel
Apply

1.3.6 Merge Column

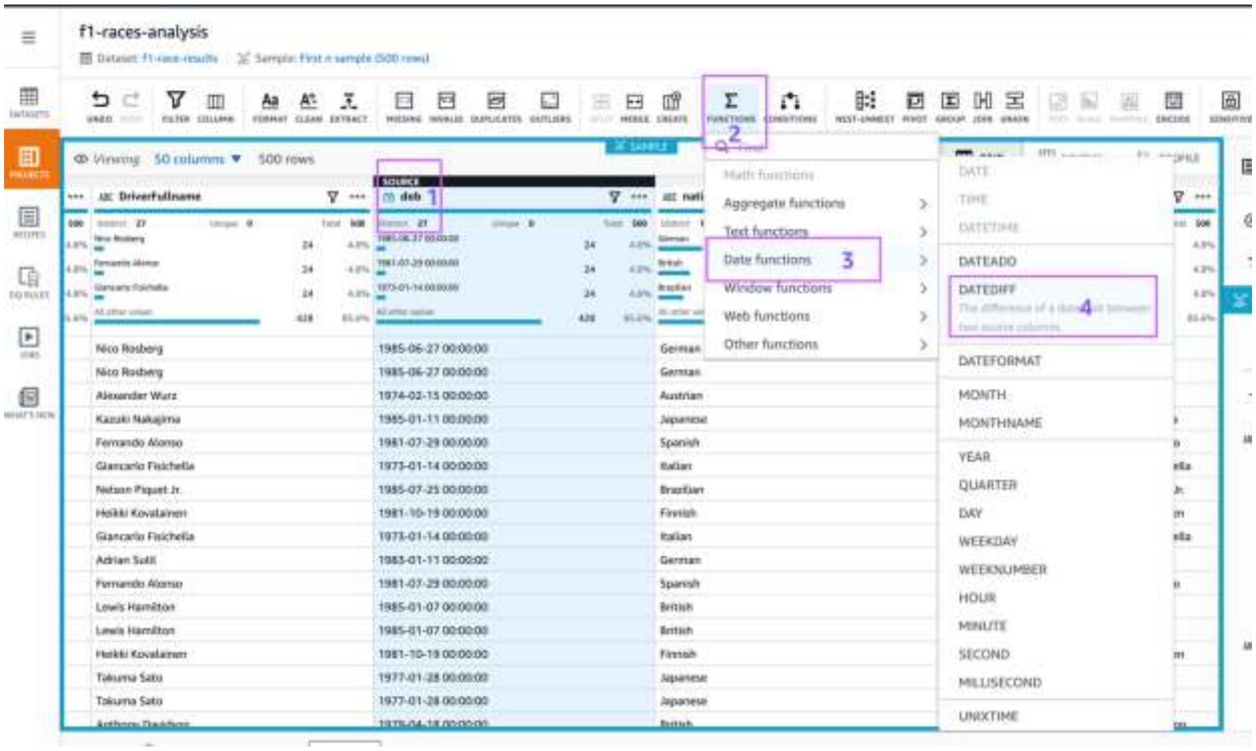
We notice there are 2 fields for driver first name and last name. We will merge these 2 fields into a single ***DriverFullname*** field.



1.3.7 Date difference Column

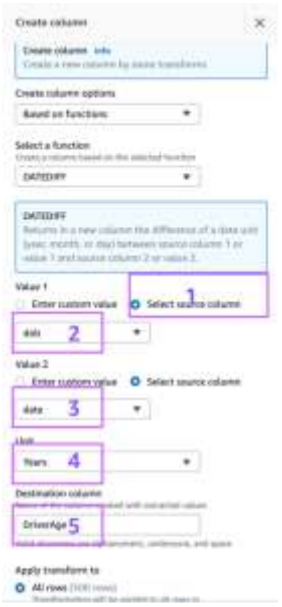
An interesting field is driver dob (date of birth), which we can use to extract driver age on the race date.

1. Click on ***dob*** field. This will highlight the field.
2. Select **FUNCTIONS** from the top menu options. Under FUNCTIONS we have many options of which we will select **Date functions** > **DATEDIFF**.



DriverFullname	dob	nati
Nico Rosberg	1985-06-27 00:00:00	German
Nico Rosberg	1985-06-27 00:00:00	German
Alexander Wurz	1974-02-15 00:00:00	Austrian
Kazuki Nakajima	1985-01-11 00:00:00	Japanese
Fernando Alonso	1981-07-29 00:00:00	Spanish
Giancarlo Fisichella	1973-01-14 00:00:00	Italian
Nelson Piquet Jr.	1985-07-25 00:00:00	Brazilian
Heikki Kovalainen	1981-10-19 00:00:00	Finnish
Giancarlo Fisichella	1973-01-14 00:00:00	Italian
Adrian Sutil	1983-01-11 00:00:00	German
Fernando Alonso	1981-07-29 00:00:00	Spanish
Lewis Hamilton	1985-01-07 00:00:00	British
Lewis Hamilton	1985-01-07 00:00:00	British
Heikki Kovalainen	1981-10-19 00:00:00	Finnish
Takuma Sato	1977-01-28 00:00:00	Japanese
Takuma Sato	1977-01-28 00:00:00	Japanese

3. For Value 1 choose **Select source column** as ***dob***. For Value 2 choose the field ***date***. Select "Years" as **Unit** and ***DriverAge*** as **Destination column**.



Create column

Create column options: Based on functions

Select a function: DATEDIFF

DATEDIFF

Value 1: Select source column — dob

Value 2: Select source column — date

Unit: Years

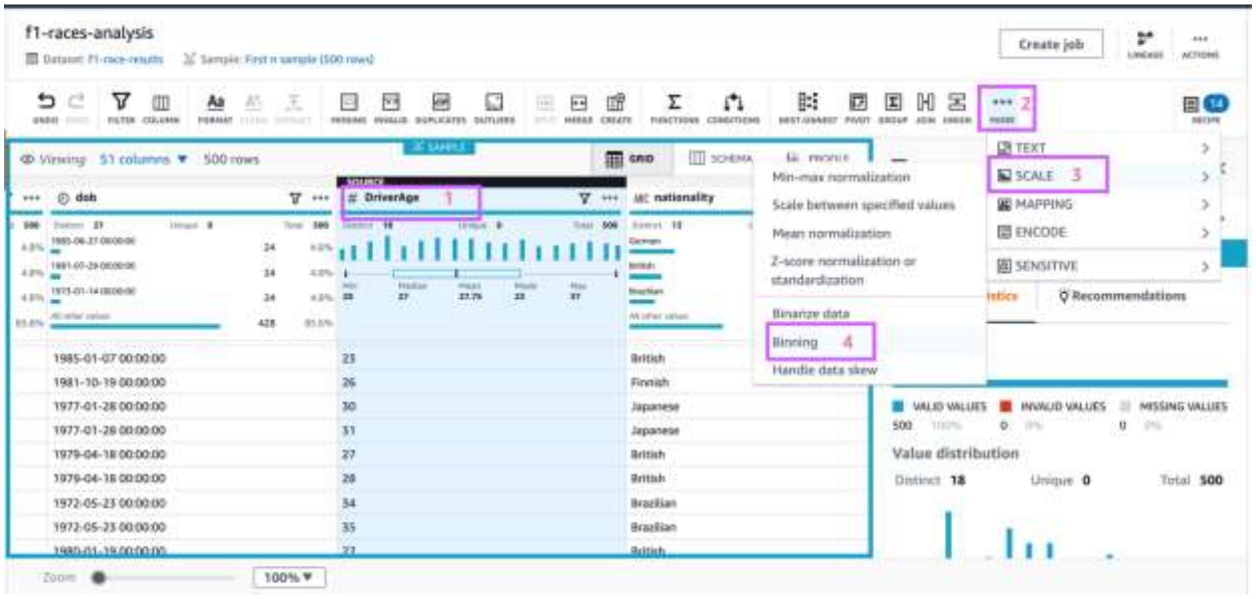
Destination column: DriverAge

Apply transform to: All rows

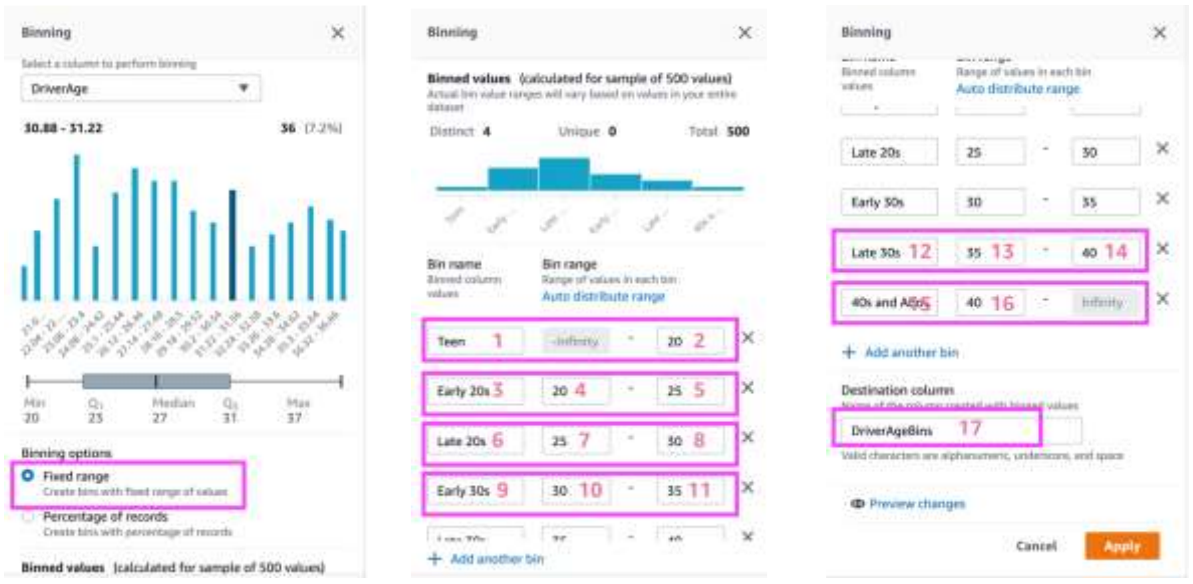
4. Hit **Apply**.

1.3.8 Binning

Now we have the driver age which is numerical values between 20 and 37. However in many analyses we would like to bucket the ages for the purpose of histogram. This is done by binning function.



f1-races-analysis



Select a column to perform binning: DriverAge

30.88 - 31.22 36 (7.2%)

Min 20, Q1 23, Median 27, Q3 31, Max 37

Binning options

- Fixed range — Create bins with fixed range of values
- Percentage of records — Create bins with percentage of records

Binned values (calculated for sample of 500 values)

Actual bin value ranges will vary based on values in your entire dataset

Distinct 4 Unique 0 Total 500

Bin name	Bin range		
Teen 1	-Infinity	-	20 2
Early 20s 3	20 4	-	25 5
Late 20s 6	25 7	-	30 8
Early 30s 9	30 10	-	35 11
Late 30s 12	35 13	-	40 14
40s and Above 15	40 16	-	Infinity

Destination column: DriverAgeBins 17

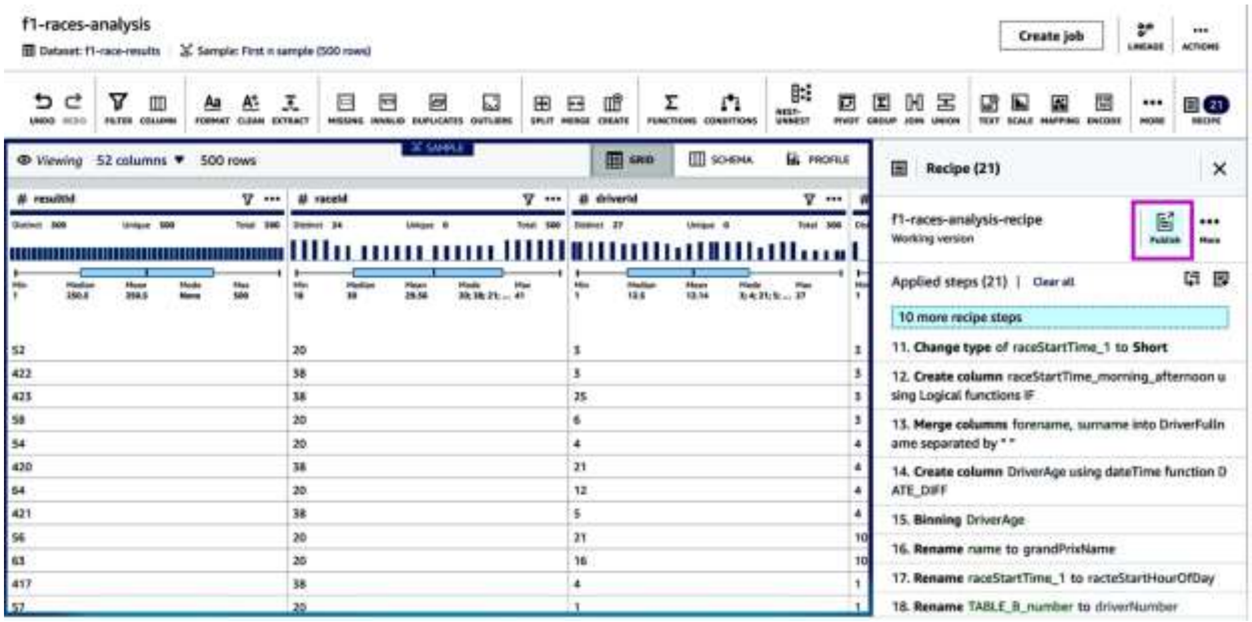
Valid characters are alphanumeric, underscore, and space

1.3.9 Rename columns

We will rename the below columns
name → grandPrixName
raceStartTime_1 → racteStartHourOfDay
TABLE_B_number → driverNumber
TABLE_B_name → constructorName
TABLE_B_nationality → constructorNationality
TABLE_B_name1 → circuitName

1.3.10 Publish Recipe

Now the transformation steps are defined we are ready to publish this recipe for future use.


f1-races-analysis

Dataset: f1-race-results | Sample: First n sample (500 rows)

Viewing 52 columns ▼ 500 rows

resultId	raceId	driverId
52	20	3
422	38	3
423	38	25
58	20	6
54	20	4
420	38	21
64	20	12
421	38	5
56	20	21
63	20	16
417	38	4
57	20	1

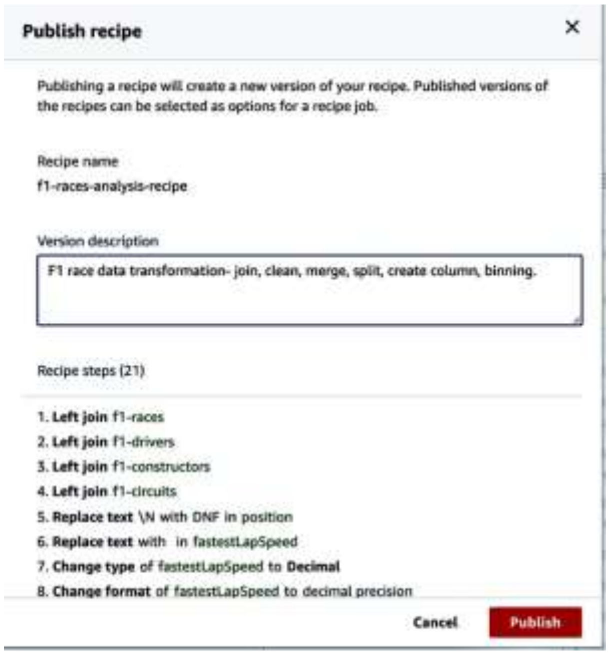
Recipe (21)

f1-races-analysis-recipe
Working version

Applied steps (21) | Clear all

10 more recipe steps

11. **Change type** of raceStartTime_1 to **Short**
12. **Create column** raceStartTime_morning_afternoon using Logical functions IF
13. **Merge columns** forename, surname into DriverFullname separated by " "
14. **Create column** DriverAge using dateTime function DATE_DIFF
15. **Binning** DriverAge
16. **Rename** name to grandPrixName
17. **Rename** raceStartTime_1 to racteStartHourOfDay
18. **Rename** TABLE_B_number to driverNumber


Publish recipe

Publishing a recipe will create a new version of your recipe. Published versions of the recipes can be selected as options for a recipe job.

Recipe name
f1-races-analysis-recipe

Version description
F1 race data transformation- join, clean, merge, split, create column, binning.

Recipe steps (21)

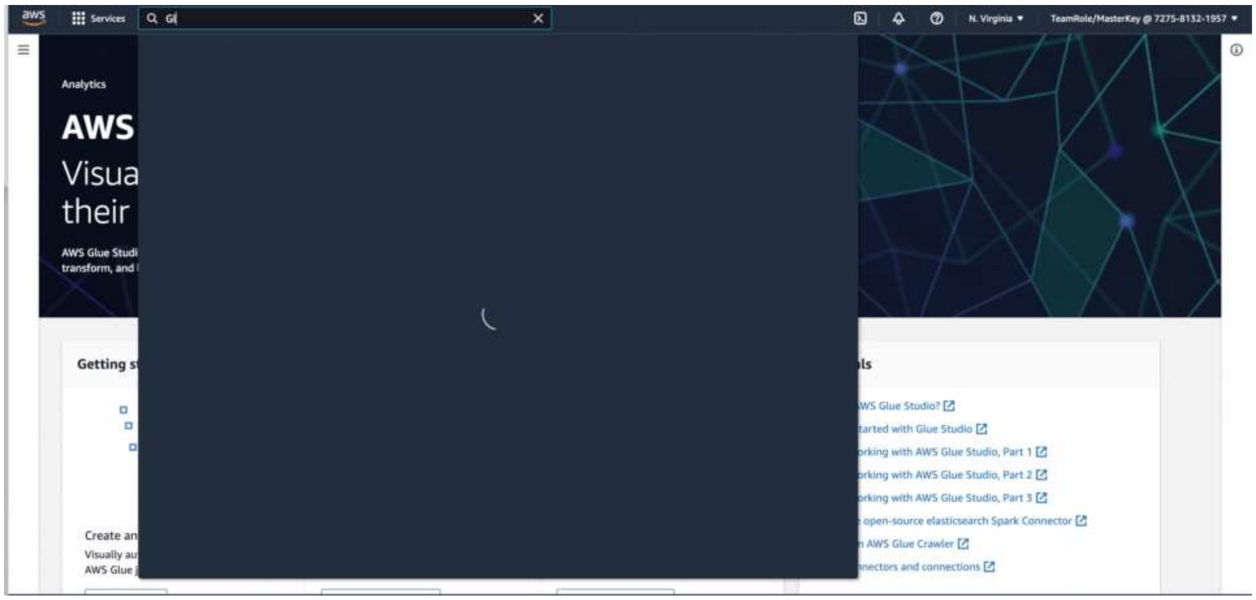
1. **Left join** f1-races
2. **Left join** f1-drivers
3. **Left join** f1-constructors
4. **Left join** f1-circuits
5. **Replace text** \N with DNF in position
6. **Replace text** with in fastestLapSpeed
7. **Change type** of fastestLapSpeed to **Decimal**
8. **Change format** of fastestLapSpeed to decimal precision

1.4 Create Job

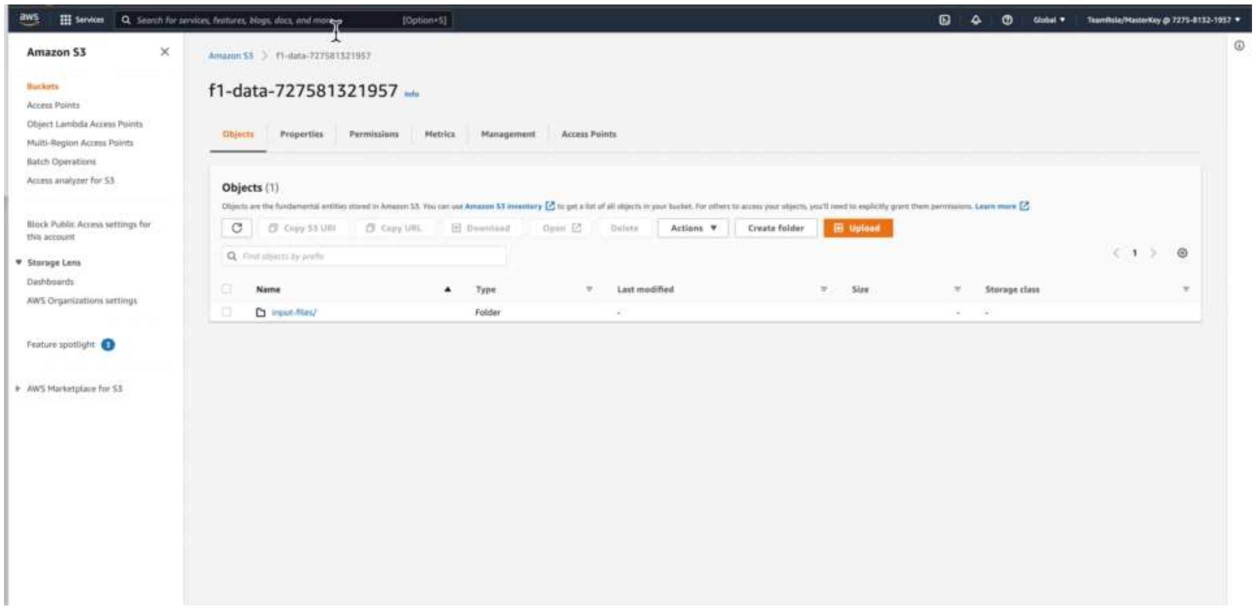
The above transformation steps are used to create a transform job that can be run as and when any new race data comes. The job will output the transformed data into another s3 folder- this folder will be registered as a Data Catalog S3 table by Glue DataBrew.

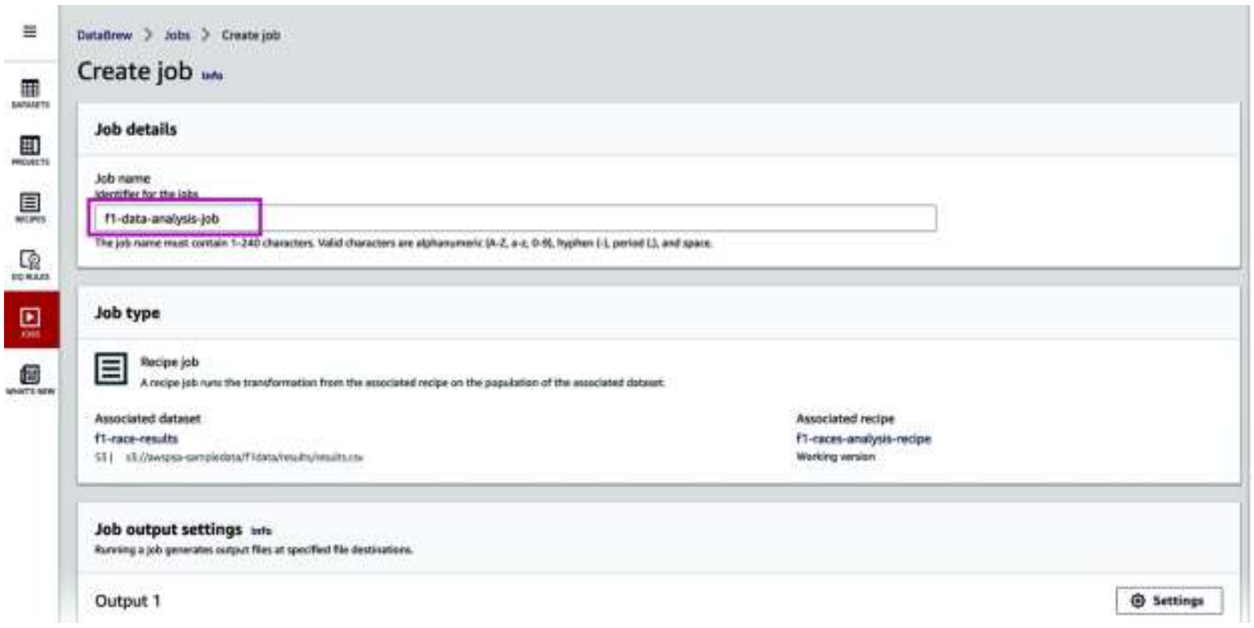
First you need to create a database in Glue if you are using Glue for the first time. If you have Glue database “default” already exists you can skip the below step.

Navigate to service **AWS Glue > Glue console > Glue catalog > Databases > Add database**


Analytics
AWS
Visua
their
Getting s
N. Virginia
TeamRole/MasterKey @ 7275-8132-1957
WS Glue Studio?
tarted with Glue Studio
rking with AWS Glue Studio, Part 1
rking with AWS Glue Studio, Part 2
rking with AWS Glue Studio, Part 3
open-source elasticsearch Spark Connector
AWS Glue Crawler
nectors and connections

Note: You may also need to attach the "S3FullAccess" policy to the "AWSGlueDataBrewServiceRole-data-analyst" role that was created during Project creation.





Create job Info

Job details

Job name
Identifier for the jobs
f1-data-analysis-job
The job name must contain 1-240 characters. Valid characters are alphanumeric (A-Z, a-z, 0-9), hyphen (-), period (.), and space.

Job type

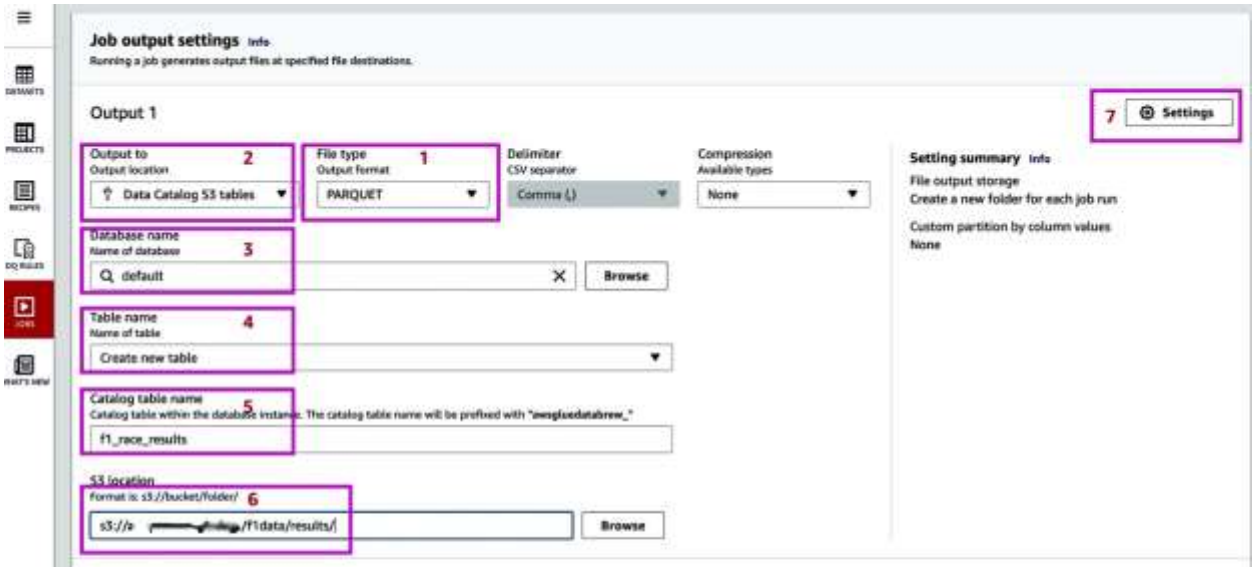
Recipe job
A recipe job runs the transformation from the associated recipe on the population of the associated dataset.

Associated dataset
f1-race-results
S3 | s3://awspsa-sampledata/f1data/results/results.csv

Associated recipe
f1-races-analysis-recipe
Working version

Job output settings Info
Running a job generates output files at specified file destinations.

Output 1



Job output settings Info
Running a job generates output files at specified file destinations.

Output 1

Output to — Output location: Data Catalog S3 tables (2)

File type — Output format: PARQUET (1)

Delimiter — CSV separator: Comma (,)

Compression — Available types: None

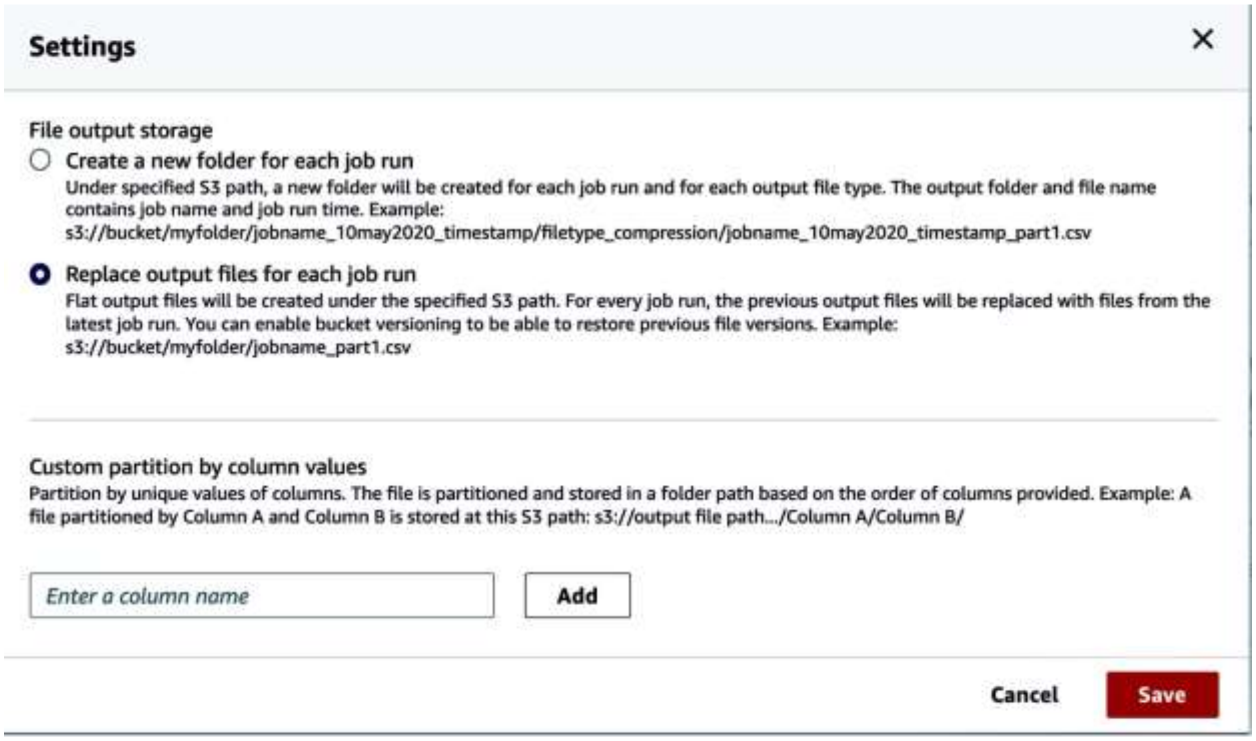
Database name — Name of database: default (3)

Table name — Name of table: Create new table (4)

Catalog table name (5)
Catalog table within the database instance. The catalog table name will be prefixed with "awsgluedatabrew_"
f1_race_results

S3 location
Format is: s3://bucket/folder/ (6)
s3://a[redacted]/f1data/results/

Setting summary Info
File output storage
Create a new folder for each job run
Custom partition by column values
None



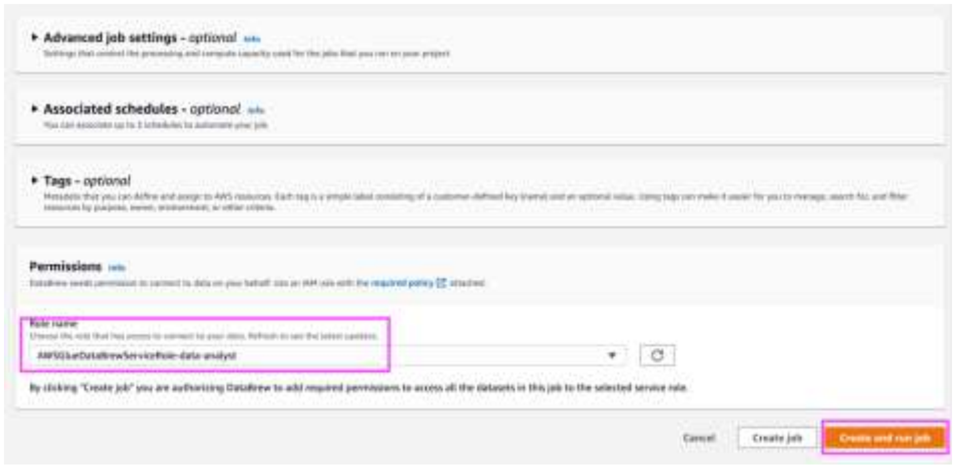
Settings

File output storage

- ( ) Create a new folder for each job run
  Under specified S3 path, a new folder will be created for each job run and for each output file type. The output folder and file name contains job name and job run time. Example:
  s3://bucket/myfolder/jobname_10may2020_timestamp/filetype_compression/jobname_10may2020_timestamp_part1.csv
- [✓] Replace output files for each job run
  Flat output files will be created under the specified S3 path. For every job run, the previous output files will be replaced with files from the latest job run. You can enable bucket versioning to be able to restore previous file versions. Example:
  s3://bucket/myfolder/jobname_part1.csv

Custom partition by column values

Partition by unique values of columns. The file is partitioned and stored in a folder path based on the order of columns provided. Example: A file partitioned by Column A and Column B is stored at this S3 path: s3://output file path.../Column A/Column B/



▸ **Advanced job settings** - *optional* Info
Settings that control the processing and compute capacity used for the jobs that you run on your project

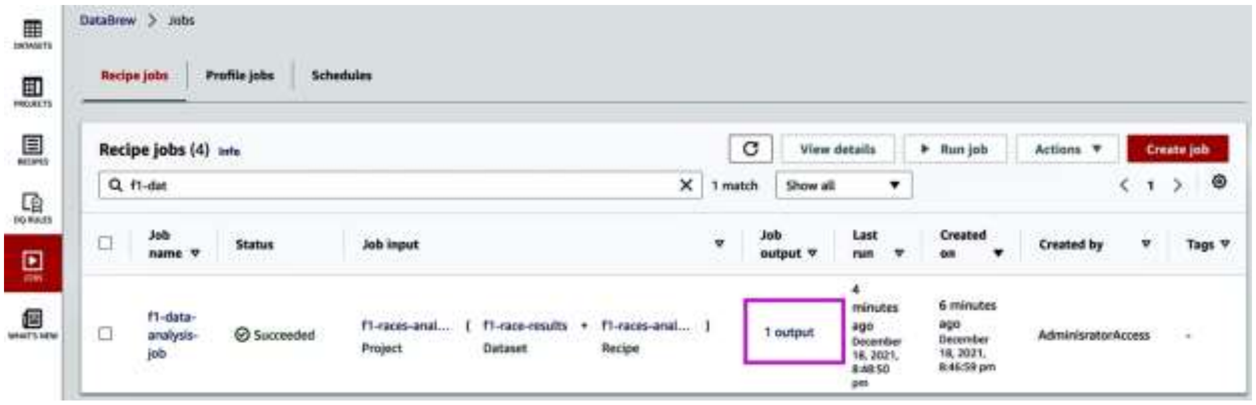
▸ **Associated schedules** - *optional* Info
You can associate up to 3 schedules to automate your job

▸ **Tags** - *optional*
[illegible]

Permissions Info
DataBrew needs permission to connect to data on your behalf. Use an IAM role with the required policy attached.

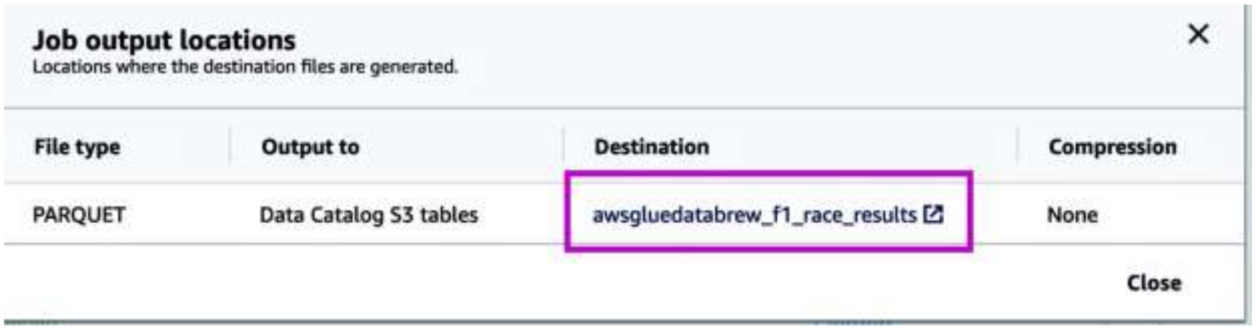
Role name
Choose the role that has access to connect to your data. Refresh to see the latest updates.
AWSGlueDataBrewServiceRole-data-analyst

By clicking "Create job" you are authorizing DataBrew to add required permissions to access all the datasets in this job to the selected service role.


DataBrew > Jobs

Recipe jobs (4)

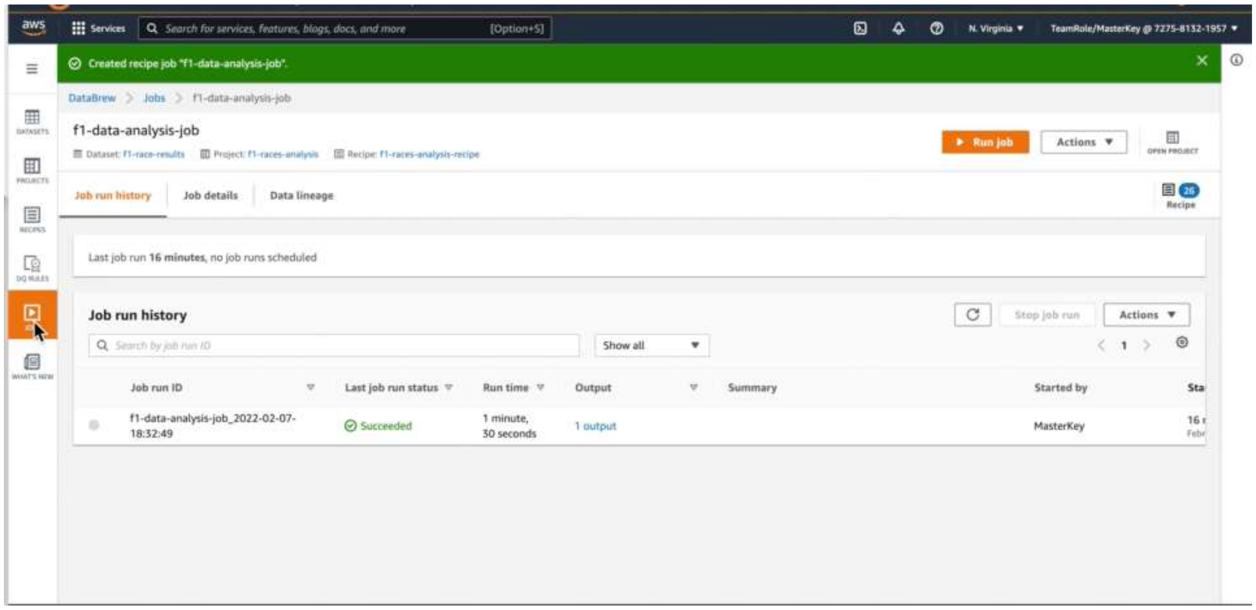
Job name	Status	Job input	Job output	Last run	Created on	Created by	Tags
f1-data-analysis-job	Succeeded	f1-races-anal... Project (f1-race-results Dataset + f1-races-anal... Recipe)	1 output	4 minutes ago December 16, 2021, 8:48:50 pm	6 minutes ago December 18, 2021, 8:46:59 pm	AdministratorAccess	-


Job output locations
Locations where the destination files are generated.

File type	Output to	Destination	Compression
PARQUET	Data Catalog S3 tables	awsgluedatabrew_f1_race_results	None

1.5 View output table

Once the Job succeeded you are ready to view the output table and run some query on.


Created recipe job "f1-data-analysis-job".

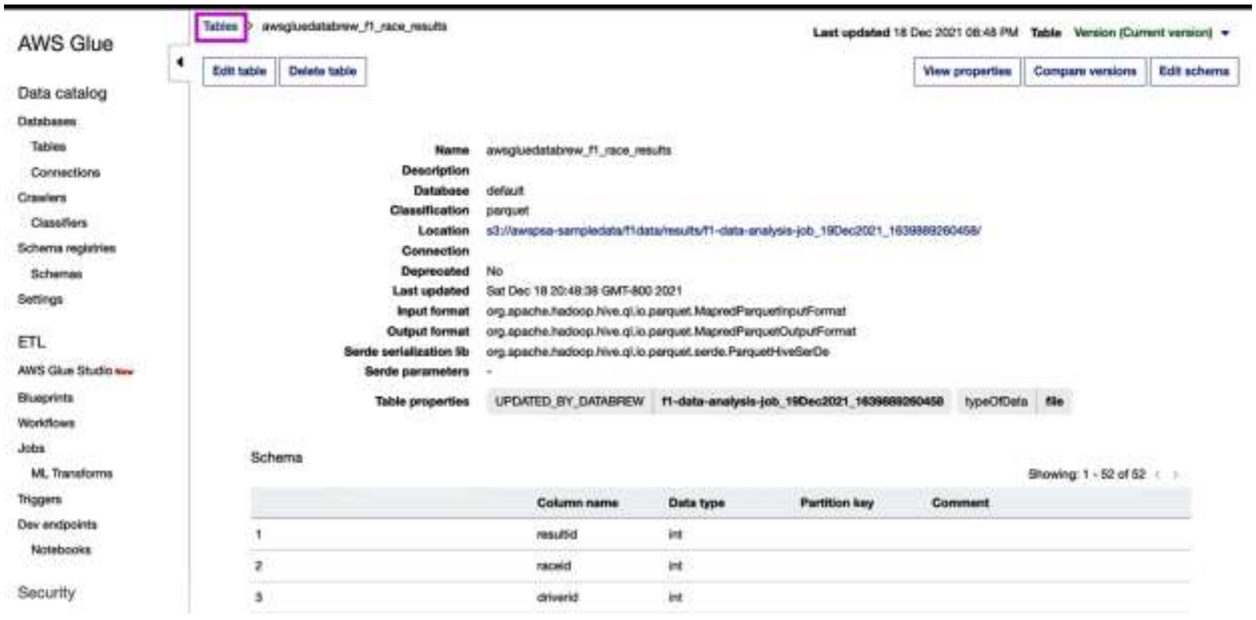
DataBrew > Jobs > f1-data-analysis-job

f1-data-analysis-job
Dataset: f1-race-results | Project: f1-races-analysis | Recipe: f1-races-analysis-recipe

Last job run 16 minutes, no job runs scheduled

Job run history

Job run ID	Last job run status	Run time	Output	Summary	Started by	Sta
f1-data-analysis-job_2022-02-07-18:32:49	Succeeded	1 minute, 30 seconds	1 output		MasterKey	16 r Febr



Tables > awsgluedatabrew_f1_race_results

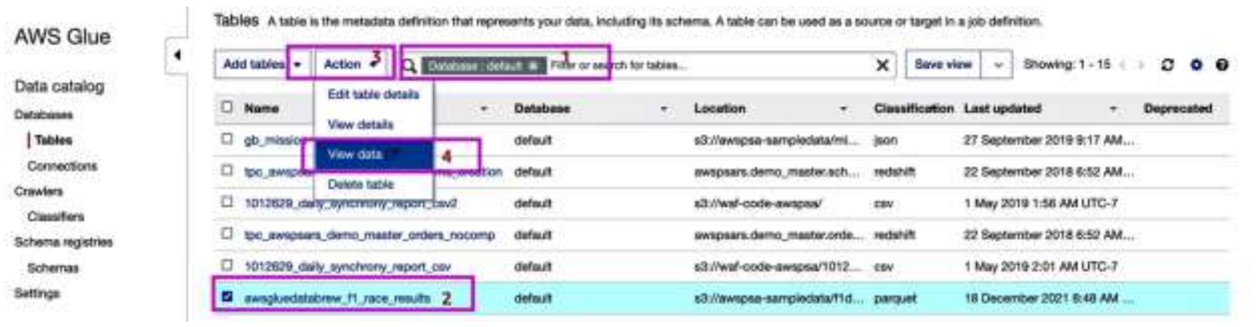
Last updated 18 Dec 2021 08:48 PM **Table** Version (Current version)

Name	awsgluedatabrew_f1_race_results
Description	
Database	default
Classification	parquet
Location	s3://awspsa-sampledata/f1data/results/f1-data-analysis-job_19Dec2021_1639889260458/
Connection	
Deprecated	No
Last updated	Sat Dec 18 20:48:38 GMT-800 2021
Input format	org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat
Output format	org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat
Serde serialization lib	org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe
Serde parameters	-
Table properties	UPDATED_BY_DATABREW **f1-data-analysis-job_19Dec2021_1639889260458** typeOfData **file**

Schema

Showing: 1 - 52 of 52

	Column name	Data type	Partition key	Comment
1	resultid	int		
2	raceid	int		
3	driverid	int		

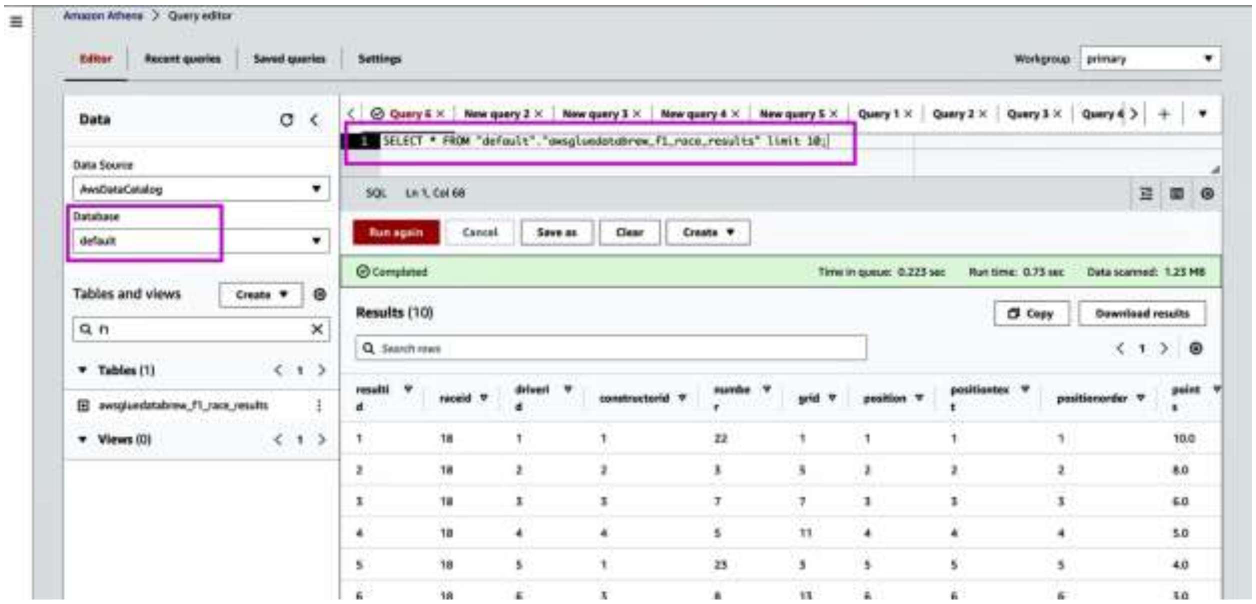


Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Filter or search for tables...

Showing: 1 - 15

Name	Database	Location	Classification	Last updated	Deprecated
☐ gb_missio...	default	s3://awspsa-sampledata/mi...	json	27 September 2019 9:17 AM...	
☐ tpc_awsp... ...ion	default	awspsars.demo_master.sch...	redshift	22 September 2018 6:52 AM...	
☐ 1012629_daily_synchrony_report_csv2	default	s3://waf-code-awspsa/	csv	1 May 2019 1:56 AM UTC-7	
☐ tpc_awspsars_demo_master_orders_nocomp	default	awspsars.demo_master.orde...	redshift	22 September 2018 6:52 AM...	
☐ 1012629_daily_synchrony_report_csv	default	s3://waf-code-awspsa/1012...	csv	1 May 2019 2:01 AM UTC-7	
☑ awsgluedatabrew_f1_race_results	default	s3://awspsa-sampledata/f1d...	parquet	18 December 2021 8:48 AM ...	



```
SELECT * FROM "default"."awsgluedatabrew_f1_race_results" limit 10;
```

Completed Time in queue: 0.223 sec Run time: 0.73 sec Data scanned: 1.25 MB

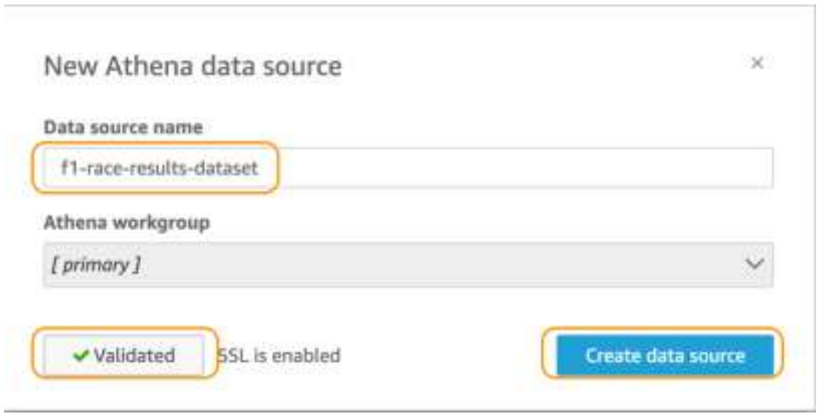
Results (10)

resultid	raceid	driverid	constructorid	number	grid	position	positiontext	positionorder	points
1	18	1	1	22	1	1	1	1	10.0
2	18	2	2	3	5	2	2	2	8.0
3	18	3	3	7	7	3	3	3	6.0
4	18	4	4	5	11	4	4	4	5.0
5	18	5	1	23	3	5	5	5	4.0
6	18	6	3	8	13	6	6	6	3.0

2. Dashboarding in QuickSight

Create dataset

Datasets > New dataset > Athena

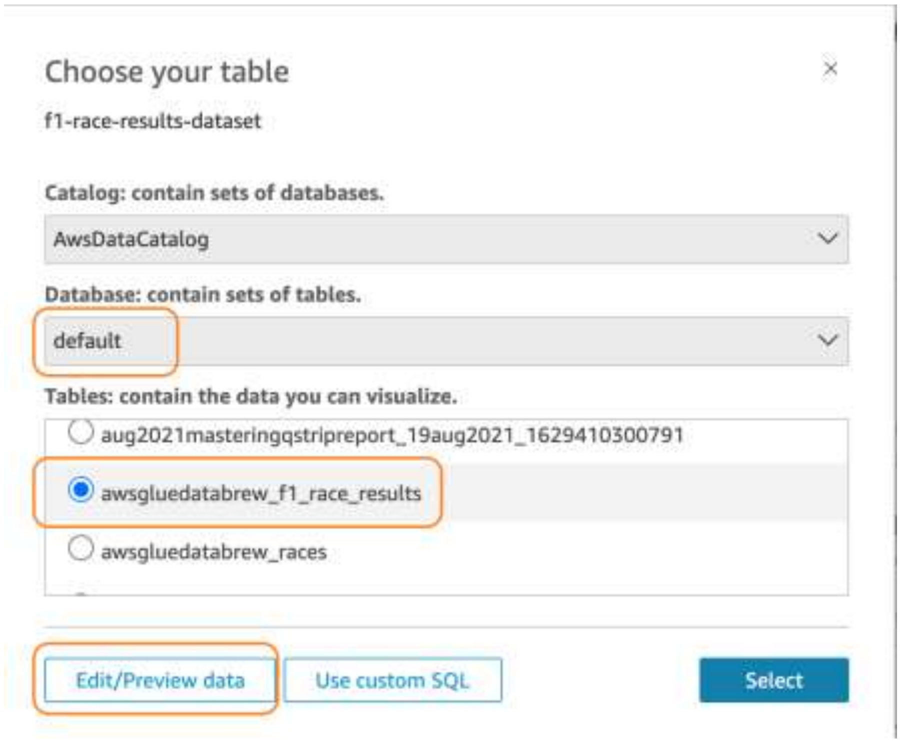


New Athena data source

Data source name: f1-race-results-dataset

Athena workgroup: [primary]

✓ Validated SSL is enabled



Choose your table

f1-race-results-dataset

Catalog: contain sets of databases.

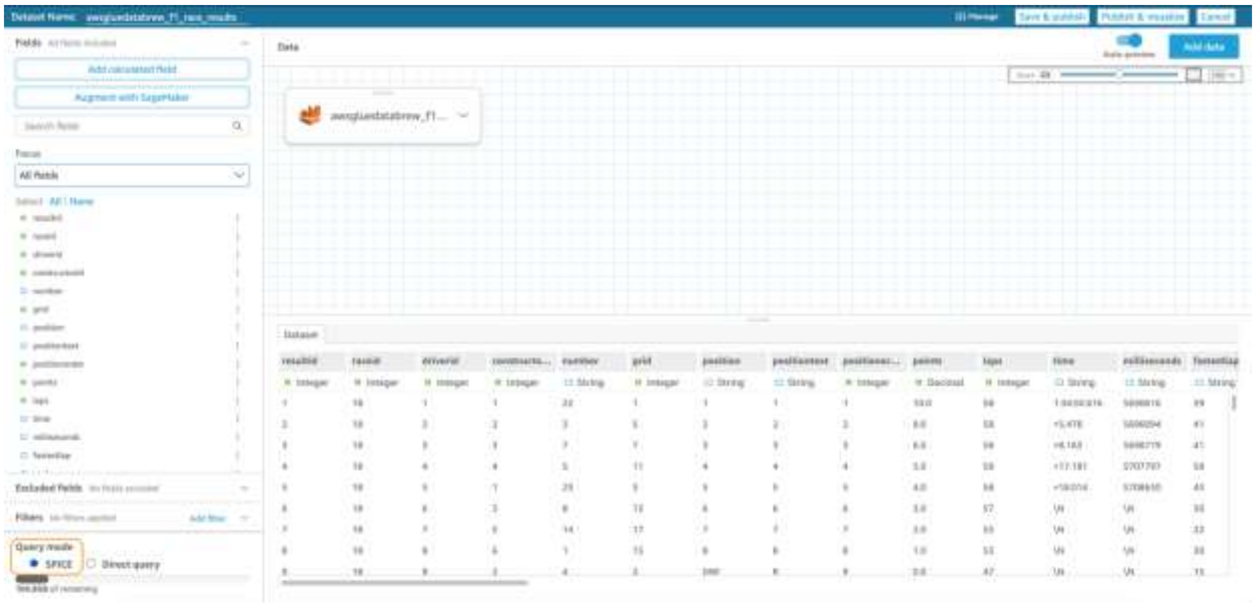
AwsDataCatalog

Database: contain sets of tables.

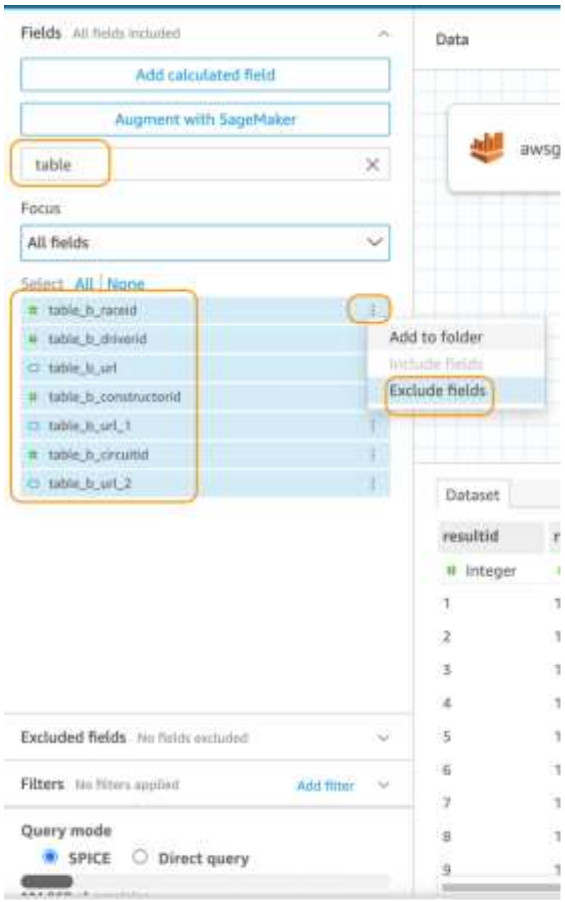
default

Tables: contain the data you can visualize.

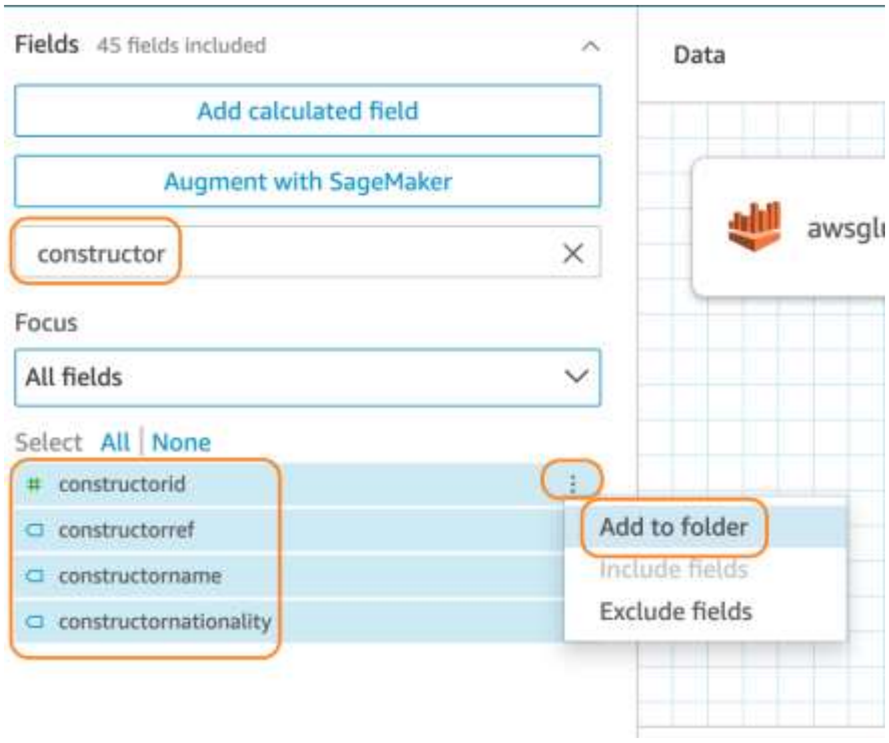
- ○ aug2021masteringqstripreport_19aug2021_1629410300791
- ● awsgluedatabrew_f1_race_results
- ○ awsgluedatabrew_races

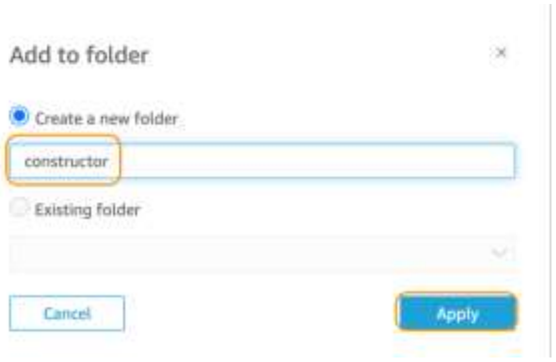


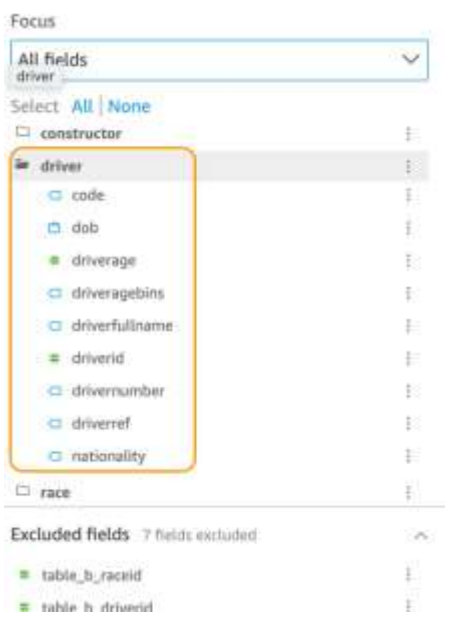
Exclude fields



Organize Fields in Folders







Focus

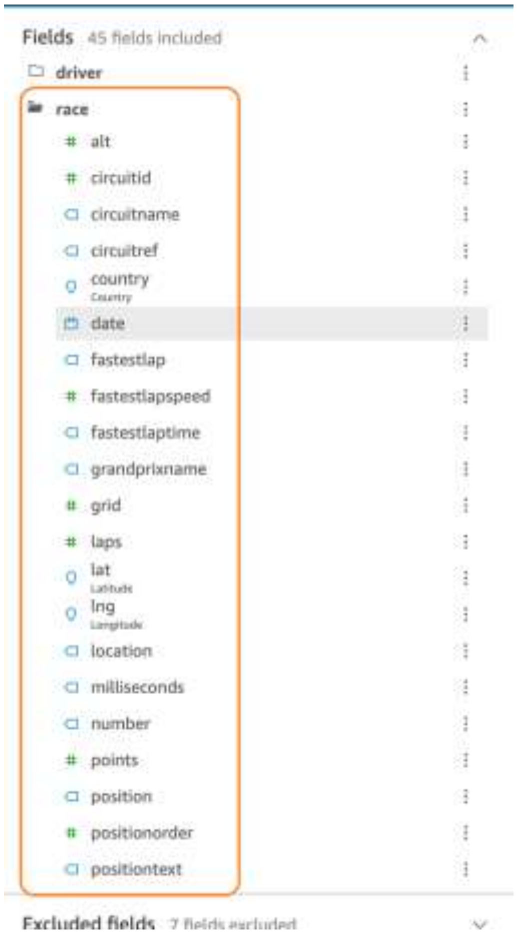
All fields
driver

Select All | None

constructor
driver
code
dob
driverage
driveragebins
driverfullname
driverid
drivernumber
driverref
nationality
race

Excluded fields 7 fields excluded

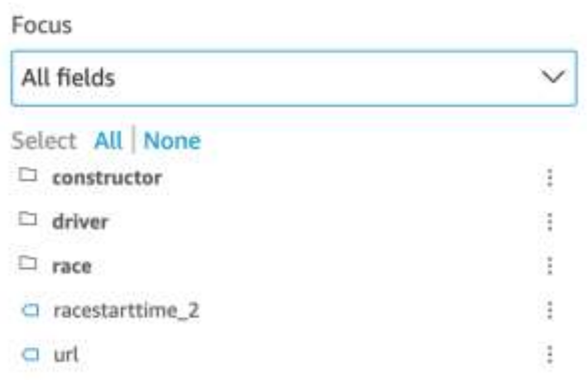
table_b_raceid
table_b_driverid



Fields 45 fields included

driver
race
alt
circuitid
circuitname
circuitref
country
Country
date
fastestlap
fastestlapspeed
fastestlaptime
grandprixname
grid
laps
lat
Latitude
lng
Longitude
location
milliseconds
number
points
position
positionorder
positiontext

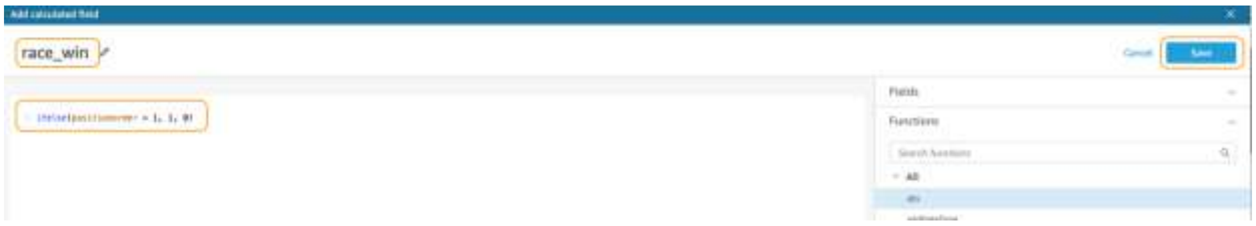
Excluded fields 7 fields excluded



Create calculated fields

1. *race_win*

ifelse(positionorder = 1, 1, 0)



2. *driver_championship_points*

sumOver(points, [driverid, year], PRE_FILTER)

3. *driver_championship_rank*

rank([sum(points) DESC], [year])

4. *driver_champion_name*

firstValue(driverfullname, [sum(points) DESC], [year])

5. *constructor_championship_points*

sumOver(points, [constructorid, year], PRE_FILTER)

6. *constructor_championship_rank*

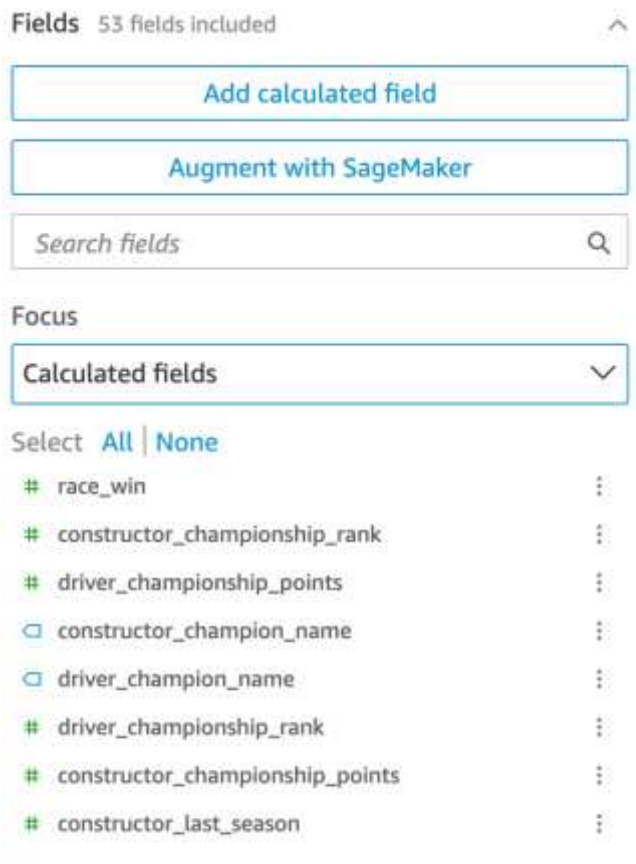
denseRank([{constructor_championship_points} DESC], [year], PRE_FILTER)

7. *constructor_champion_name*

firstValue(constructorname, [sum(points) DESC], [year])

8. *constructor_last_season*

maxOver(year, [constructorid], PRE_FILTER)



Fields 53 fields included

Add calculated field

Augment with SageMaker

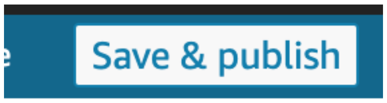
Search fields

Focus

Calculated fields

Select All | None

- race_win
- constructor_championship_rank
- driver_championship_points
- constructor_champion_name
- driver_champion_name
- driver_championship_rank
- constructor_championship_points
- constructor_last_season

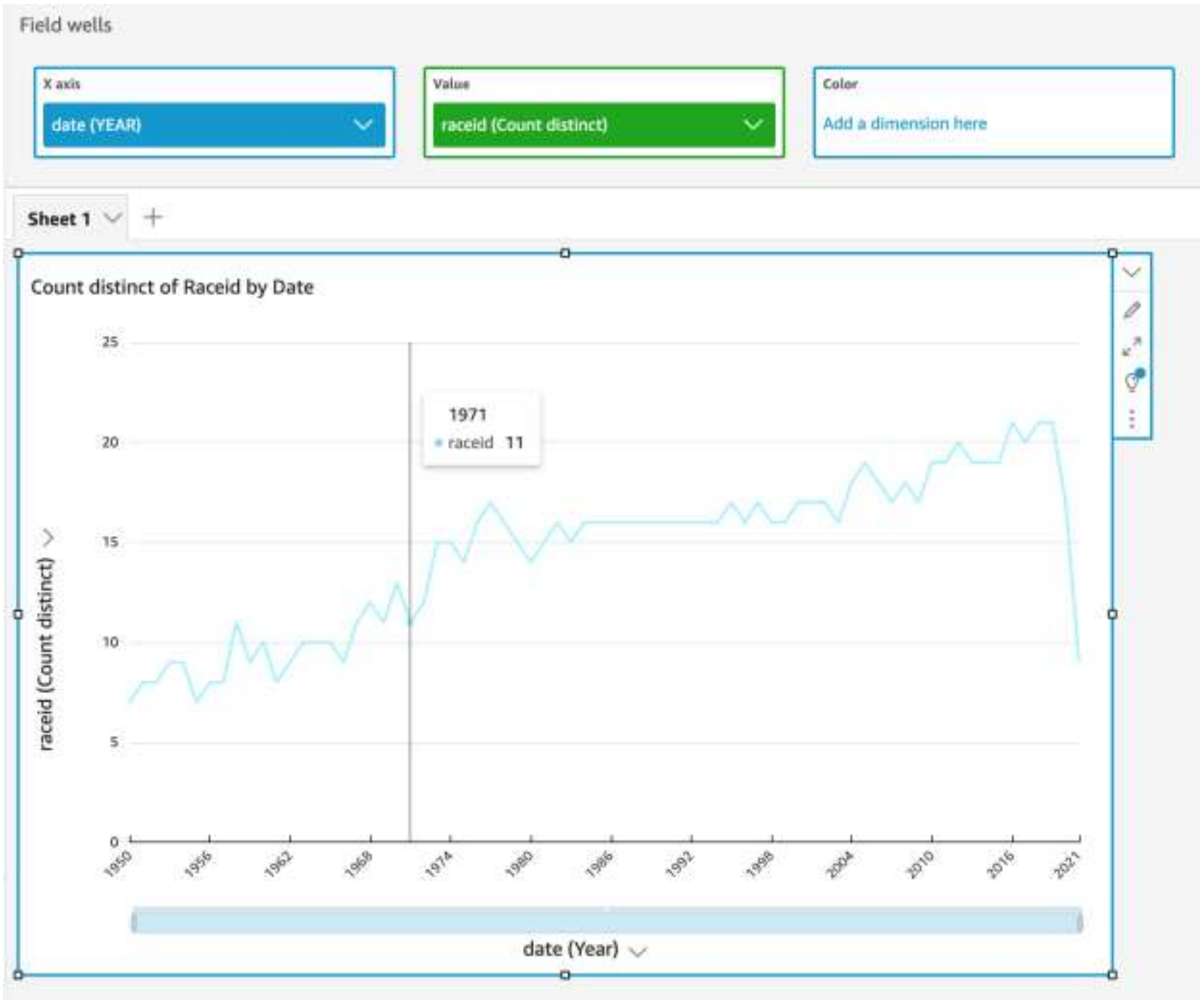


Start analysis

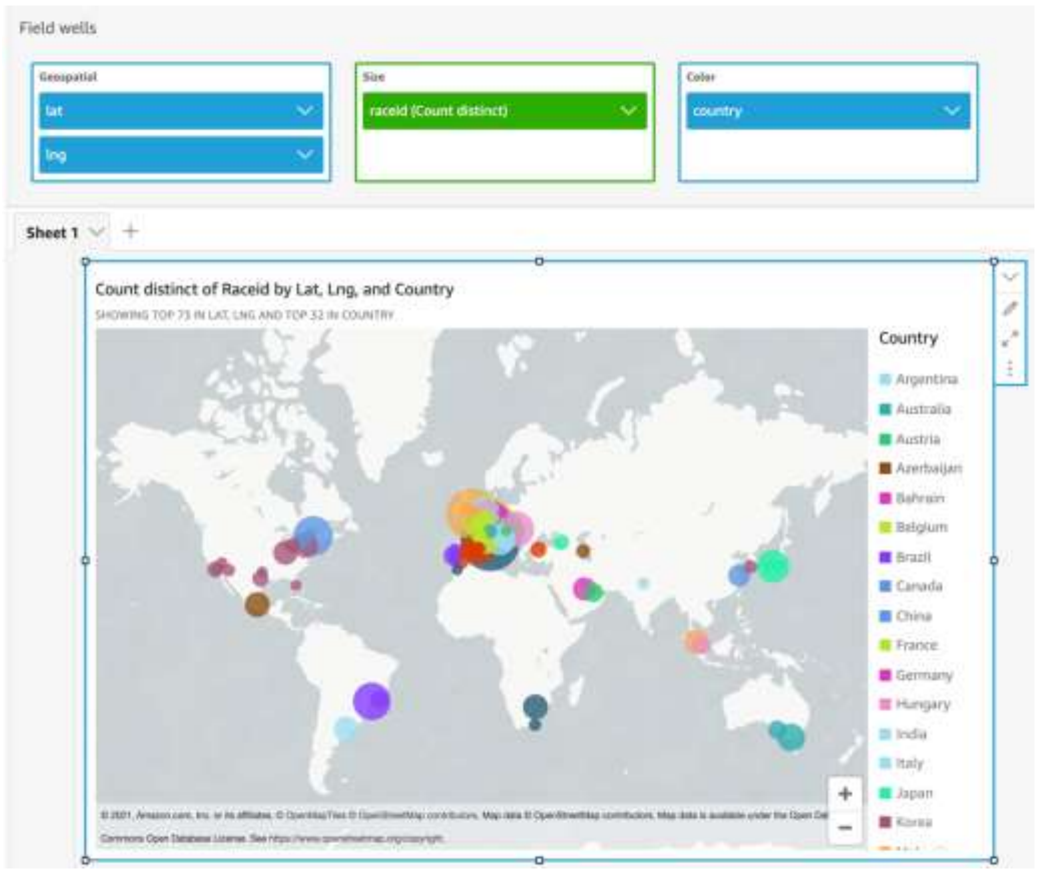
Sheet - Overview

<u>Races over the years</u>

Fields to select: *date (YEAR), raceId(Count distinct)*



<u>Geo locations where races been played</u>
Points on Map
Fields to select:
lat, lng → Geospatial
racedId(Count distinct) →Size
country → Color


Field wells

Geospatial: lat, lng

Size: raceId (Count distinct)

Color: country

Sheet 1

Count distinct of Raceid by Lat, Lng, and Country

SHOWING TOP 73 IN LAT, LNG AND TOP 32 IN COUNTRY

<u>Races in Countries & Circuits</u>

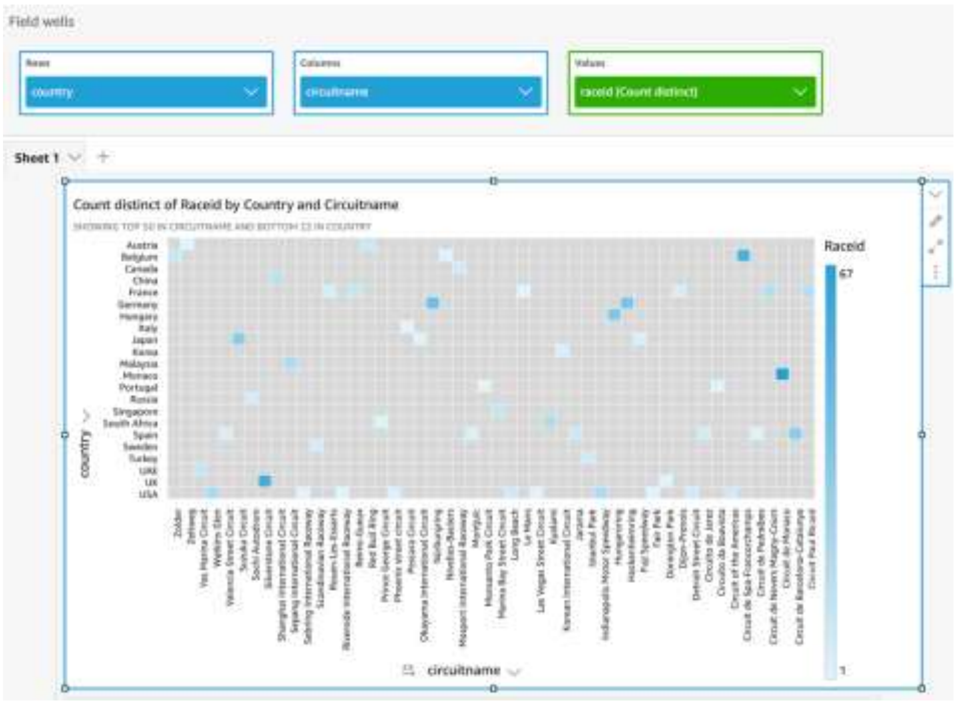
Heat map

Fields to select:

country → Rows

circuitname →Columns

racedId(Count distinct) → Values


Field wells

Rows: country

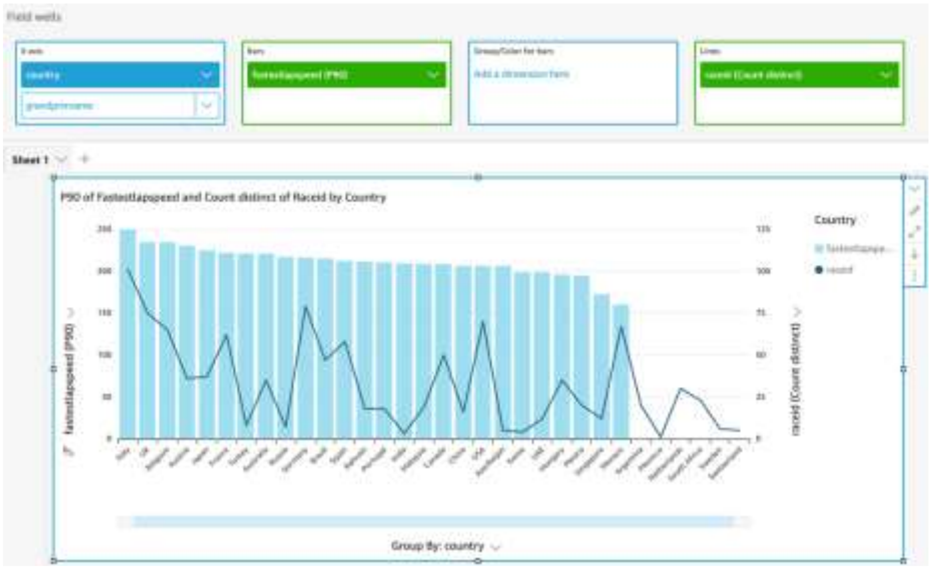
Columns: circuitname

Values: raceid (Count distinct)

Sheet 1

Count distinct of Raceid by Country and Circuitname

<u>Fastest LapSpeed</u>
Clustered bar combo chart
Fields to select:
Country, grandprixname drill down → X axis
fastestlatspeed (P90) →Bar
racedId(Count distinct) → Lines


Field wells

X axis: country, grandprixname

Bars: fastestlapspeed (P90)

Group/Color for bars: Add a dimension here

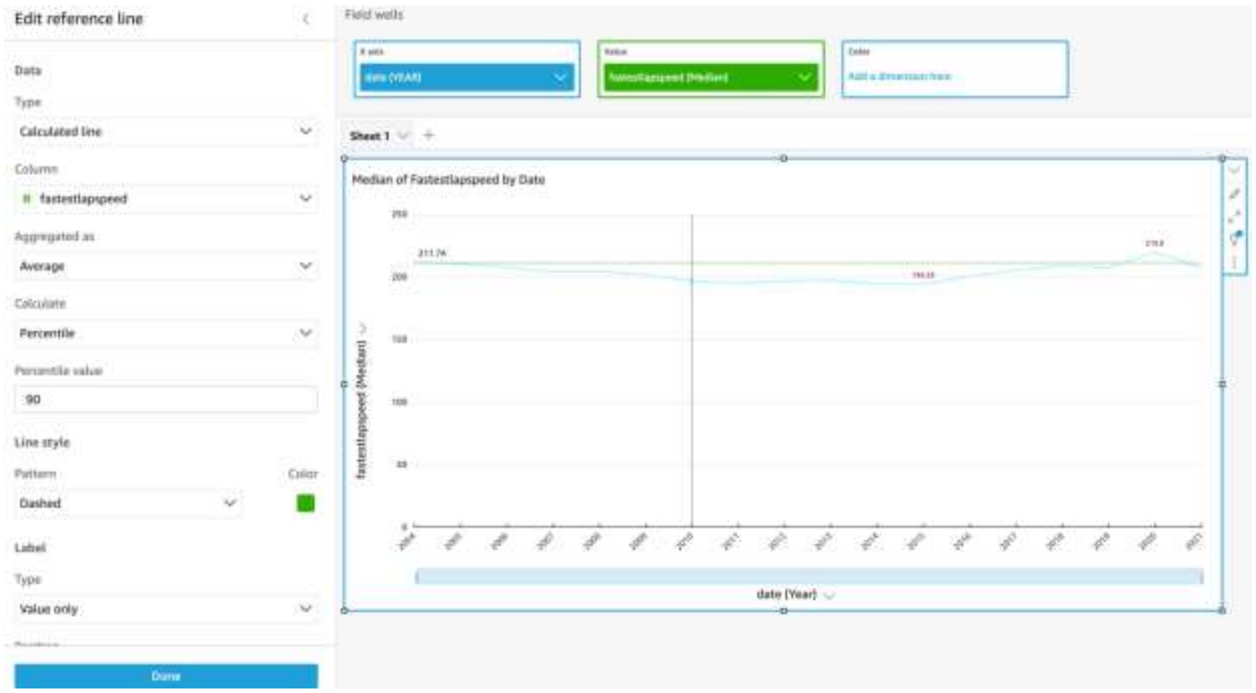
Lines: raceid (Count distinct)

Sheet 1

P90 of Fastestlapspeed and Count distinct of Raceid by Country

<u>LapSpeed Overtime</u>
Line chart
Fields to select:

Date(Year)→X axis
fastestlapspeed (Median) →Value

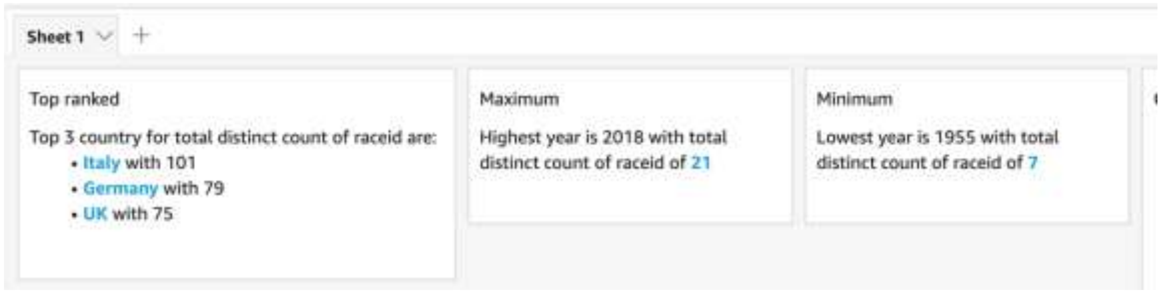

Edit reference line

Data
Type: Calculated line
Column: fastestlapspeed
Aggregated as: Average
Calculate: Percentile
Percentile value: 90

Line style
Pattern: Dashed
Color: 🟩

Label
Type: Value only

Insights


Sheet 1

Top ranked
Top 3 country for total distinct count of raceid are:
- Italy with 101
- Germany with 79
- UK with 75

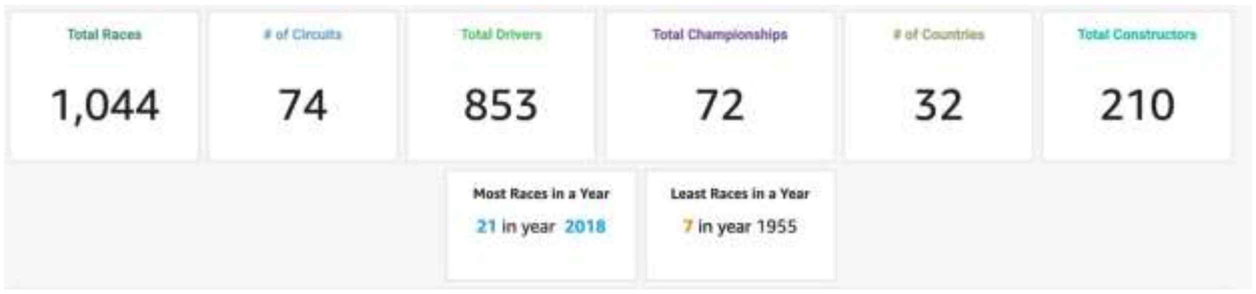
Maximum
Highest year is 2018 with total distinct count of raceid of 21

Minimum
Lowest year is 1955 with total distinct count of raceid of 7

Extras:
KPIs

- # of Races
- # of Circuits
- # of Drivers
- # of Championships
- # of Countries

Formatting of Insights, KPIs and Analysis



Total Races
1,044

# of Circuits
74

Total Drivers
853

Total Championships
72

# of Countries
32

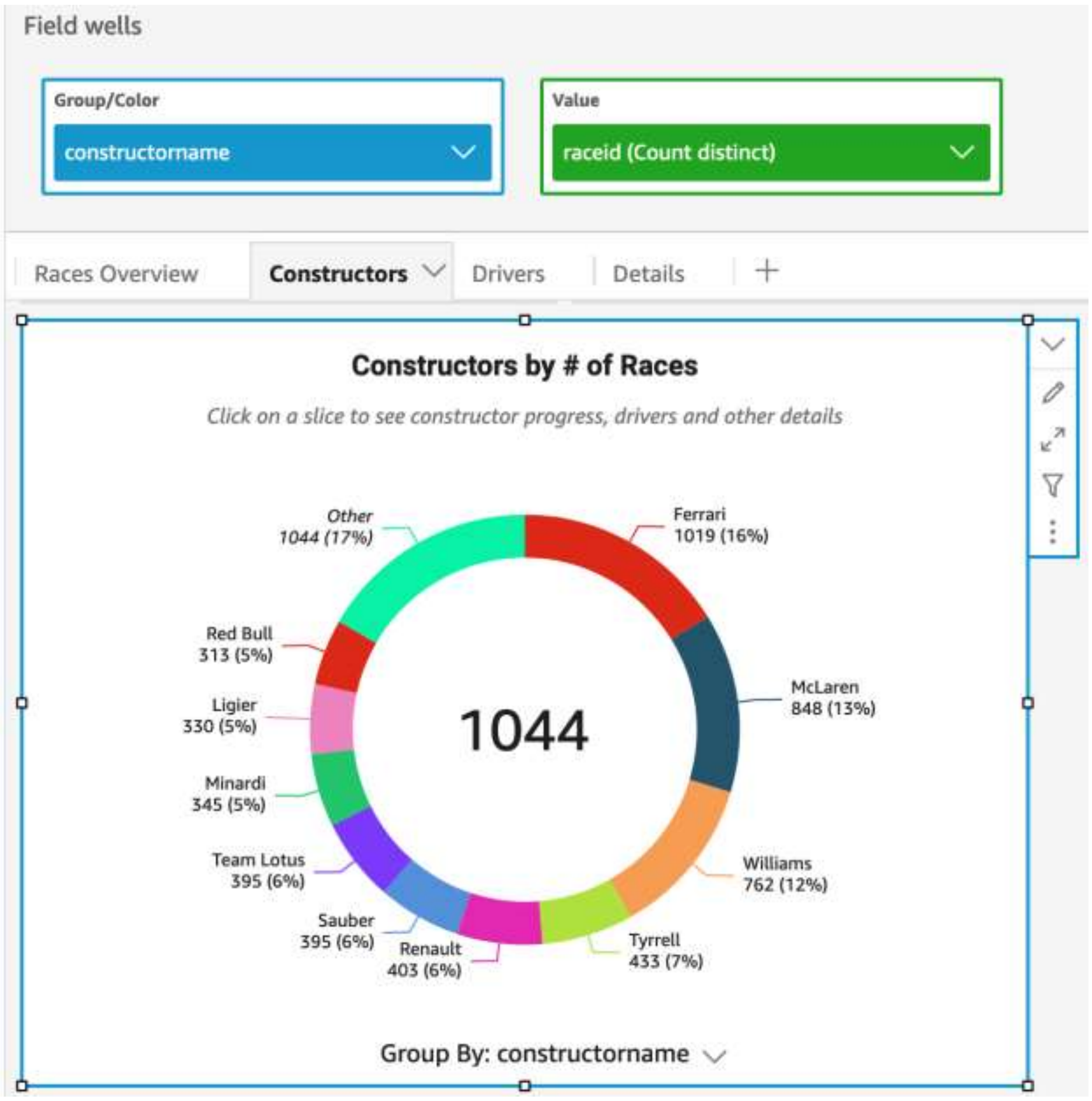
Total Constructors
210

Most Races in a Year
21 in year 2018

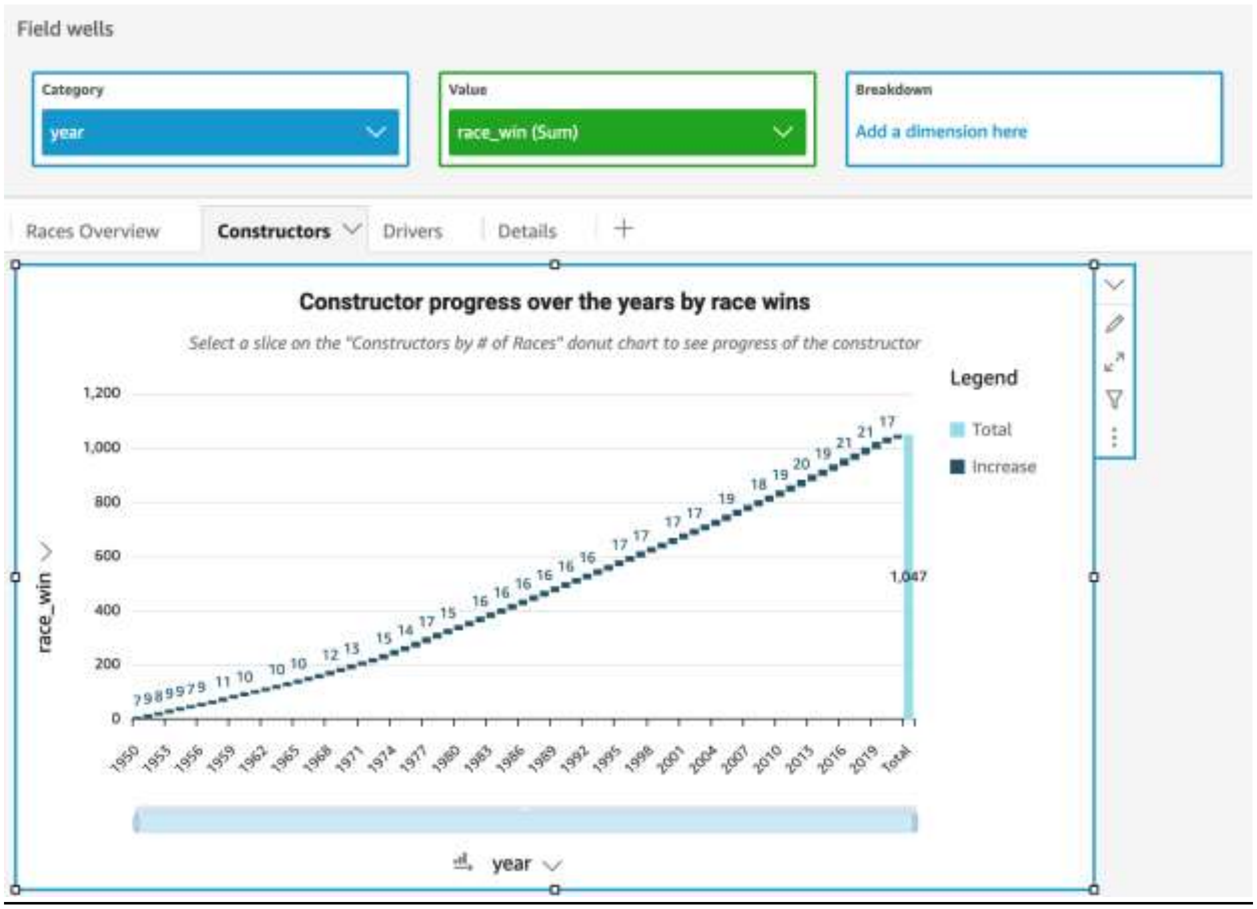
Least Races in a Year
7 in year 1955

Sheet – Constructors

Constructors by # of Races



<u>**Constructor progress over the years by race wins**</u>



Field wells

Category: year

Value: race_win (Sum)

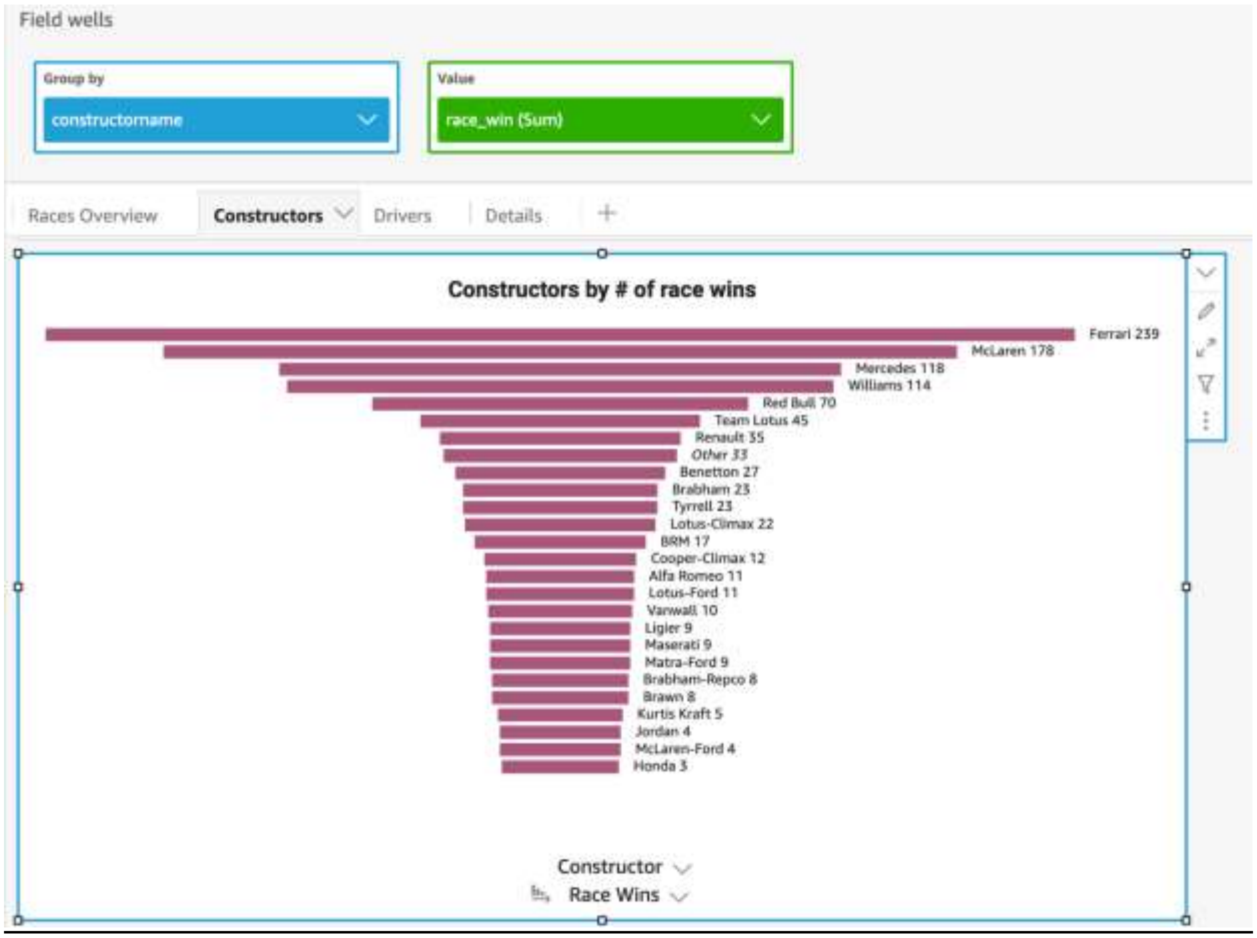
Breakdown: Add a dimension here

Races Overview | Constructors | Drivers | Details

Constructor progress over the years by race wins

Select a slice on the "Constructors by # of Races" donut chart to see progress of the constructor

Constructors by # of race wins

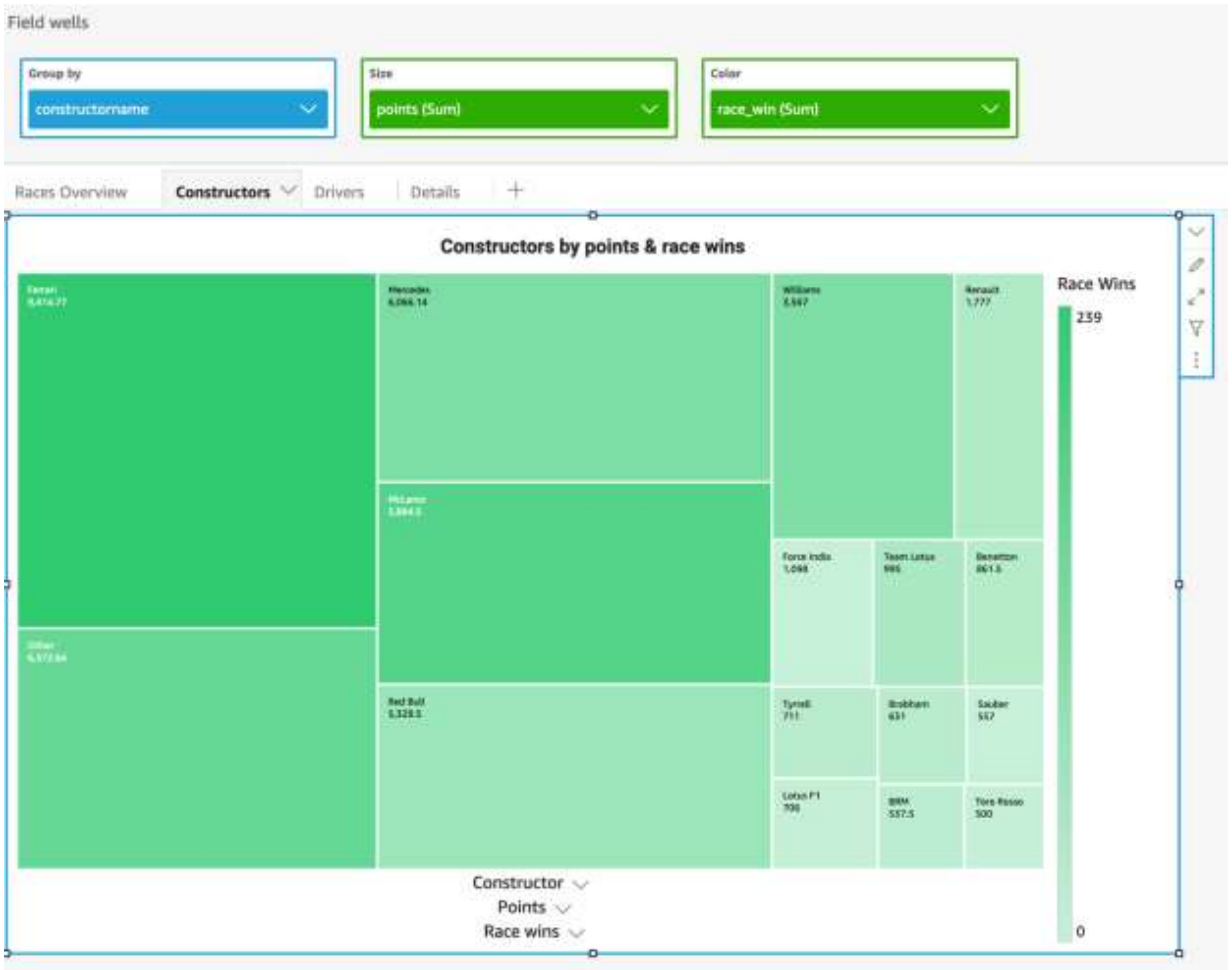


Field wells

Group by: constructorname

Value: race_win (Sum)

Constructors by points & race wins

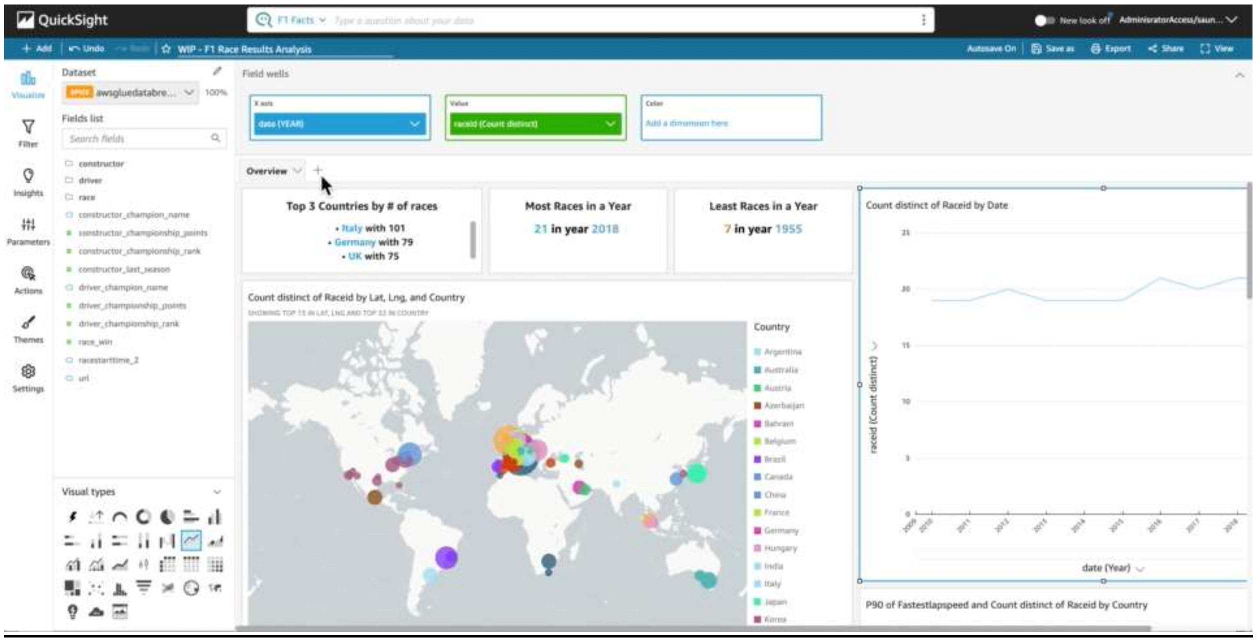

Field wells

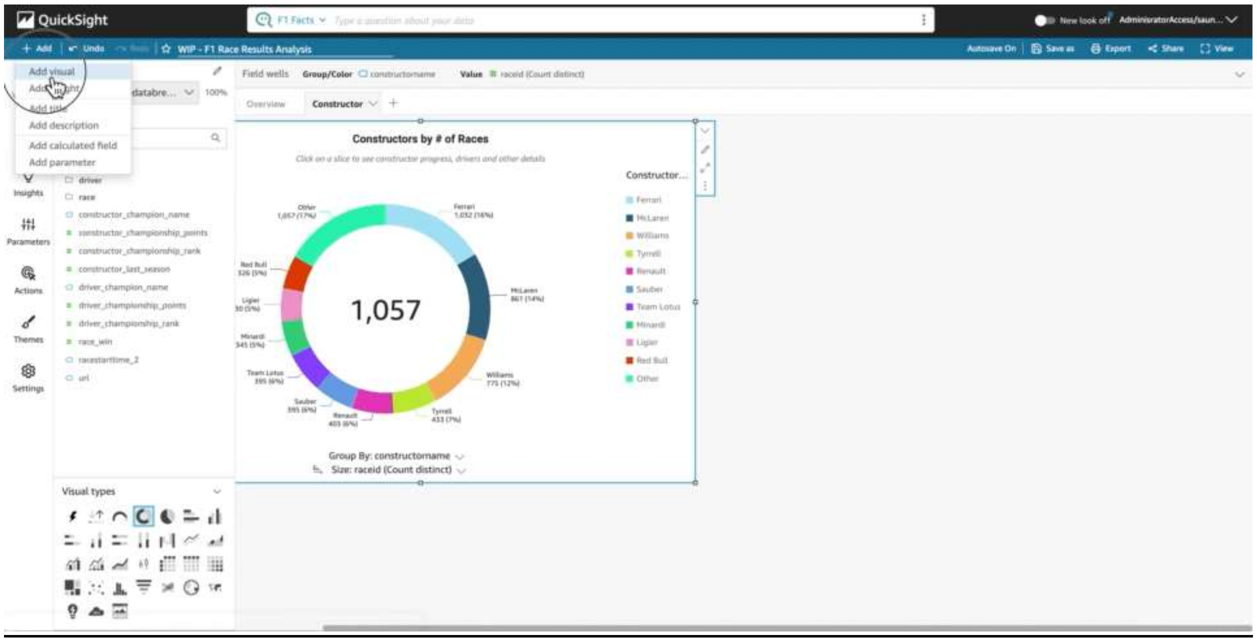
Group by: constructorname

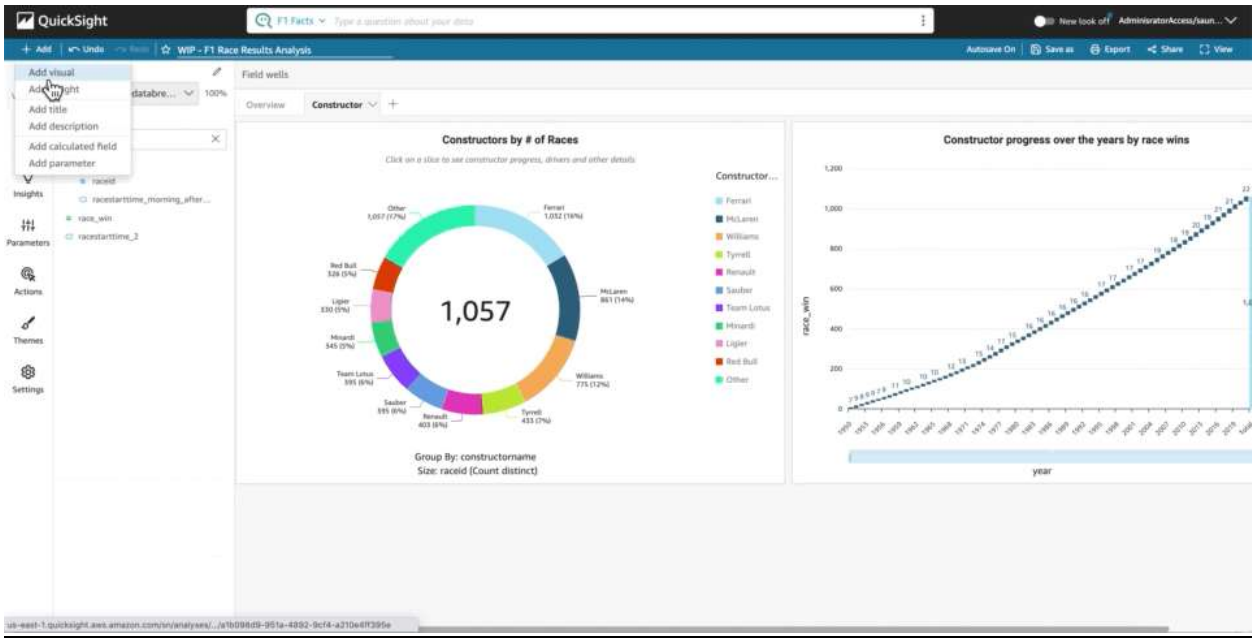
Size: points (Sum)

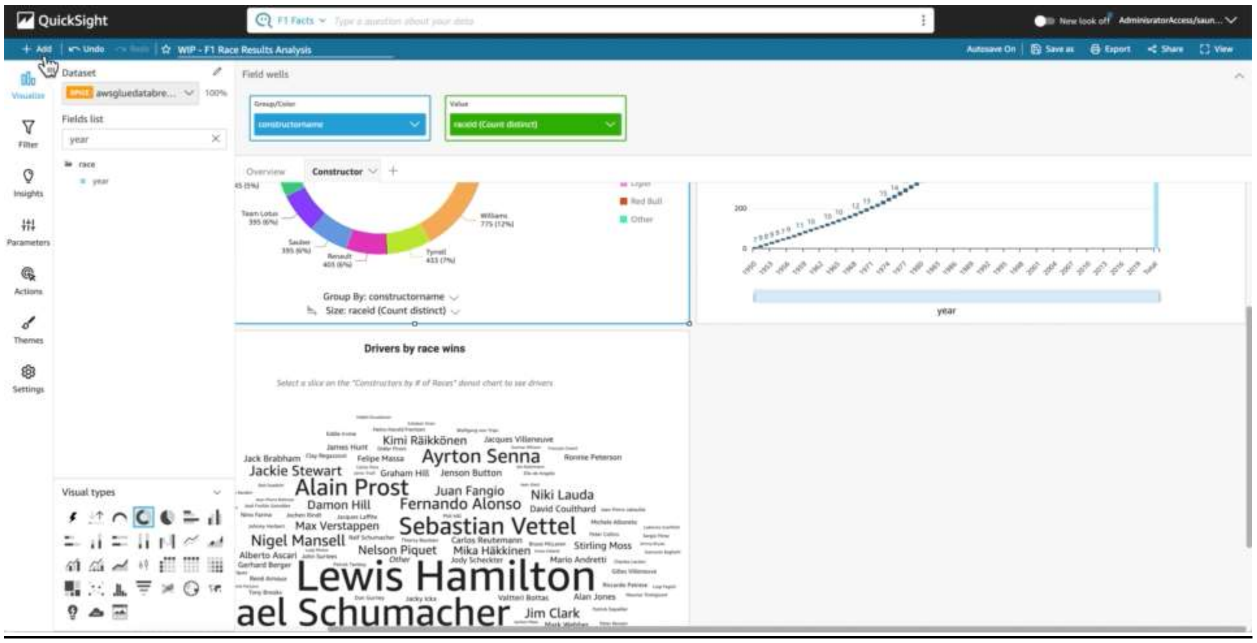
Color: race_win (Sum)

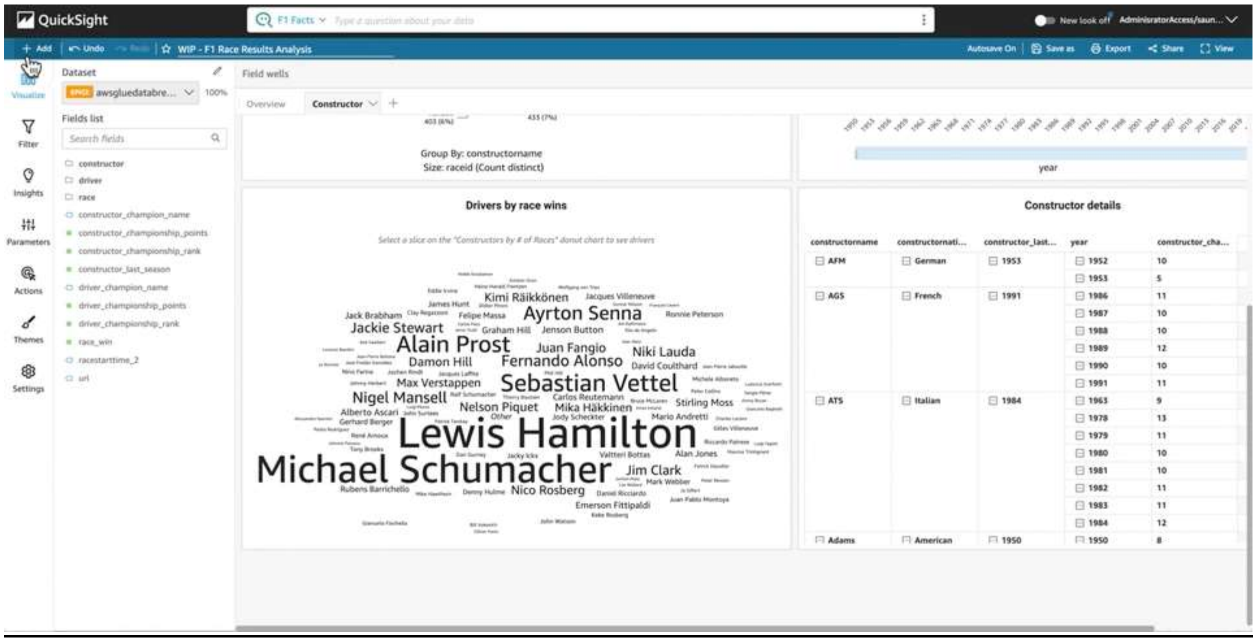
Constructor details


QuickSight
F1 Facts
WIP - F1 Race Results Analysis
Field wells
X axis: date (YEAR)
Value: raceid (Count distinct)
Overview
Top 3 Countries by # of races
• Italy with 101
• Germany with 79
• UK with 75
Most Races in a Year
21 in year 2018
Least Races in a Year
7 in year 1955
Count distinct of Raceid by Date
Count distinct of Raceid by Lat, Lng, and Country
P90 of Fastestlapspeed and Count distinct of Raceid by Country


QuickSight
Add visual
Add insight
Add title
Add description
Add calculated field
Add parameter
Constructor
Constructors by # of Races
1,057
Group By: constructorname
Size: raceid (Count distinct)


QuickSight
F1 Facts
+ Add
Undo
WIP - F1 Race Results Analysis
Autosave On
Save as
Export
Share
View
Add visual
Add insight
Add title
Add description
Add calculated field
Add parameter
Field wells
Overview
Constructor
Constructors by # of Races
Click on a slice to see constructor progress, drivers and other details
1,057
Ferrari 1,052 (16%)
McLaren 861 (14%)
Williams 775 (12%)
Tyrrell 433 (7%)
Renault 403 (6%)
Sauber 395 (6%)
Team Lotus 395 (6%)
Minardi 345 (5%)
Ligier 330 (5%)
Red Bull 326 (5%)
Other 1,057 (17%)
Group By: constructorname
Size: raceid (Count distinct)
Constructor progress over the years by race wins
race_win
year
Insights
Parameters
Actions
Themes
Settings


QuickSight
F1 Facts
+ Add
Undo
WIP - F1 Race Results Analysis
Dataset
awsgluedatabre...
Fields list
year
race
Field wells
Group/Color
constructorname
Value
raceid (Count distinct)
Overview
Constructor
Group By: constructorname
Size: raceid (Count distinct)
Drivers by race wins
Select a slice on the "Constructors by # of Races" donut chart to see drivers
Lewis Hamilton
Michael Schumacher
Sebastian Vettel
Alain Prost
Ayrton Senna
Fernando Alonso
Nigel Mansell
Jackie Stewart
Niki Lauda
Kimi Räikkönen
Max Verstappen
Nelson Piquet
Damon Hill
Juan Fangio
Jim Clark
Visual types
Visualize
Filter
Insights
Parameters
Actions
Themes
Settings



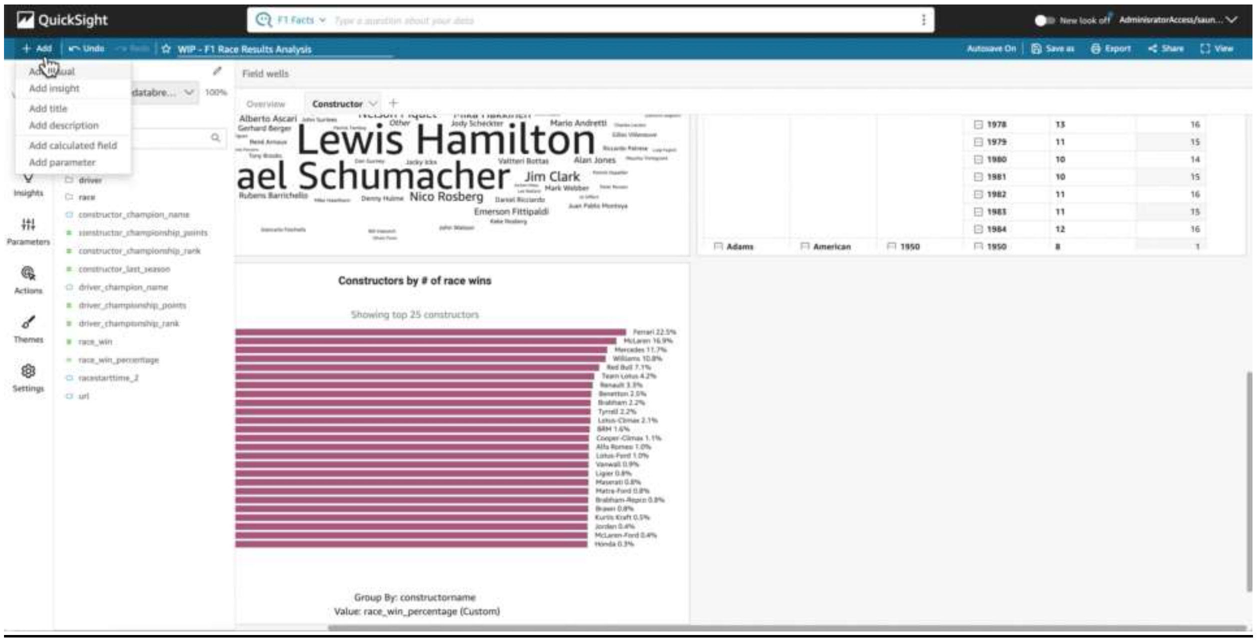
Group By: constructorname
Size: raceid (Count distinct)

Drivers by race wins

Select a slice on the "Constructors by # of Races" donut chart to see drivers

Constructor details

constructorname	constructornati...	constructor_last...	year	constructor_cha...
AFM	German	1953	1952	10
			1953	5
AGS	French	1991	1986	11
			1987	10
			1988	10
			1989	12
			1990	10
			1991	11
ATS	Italian	1984	1963	9
			1978	13
			1979	11
			1980	10
			1981	10
			1982	11
			1983	11
			1984	12
Adams	American	1950	1950	8

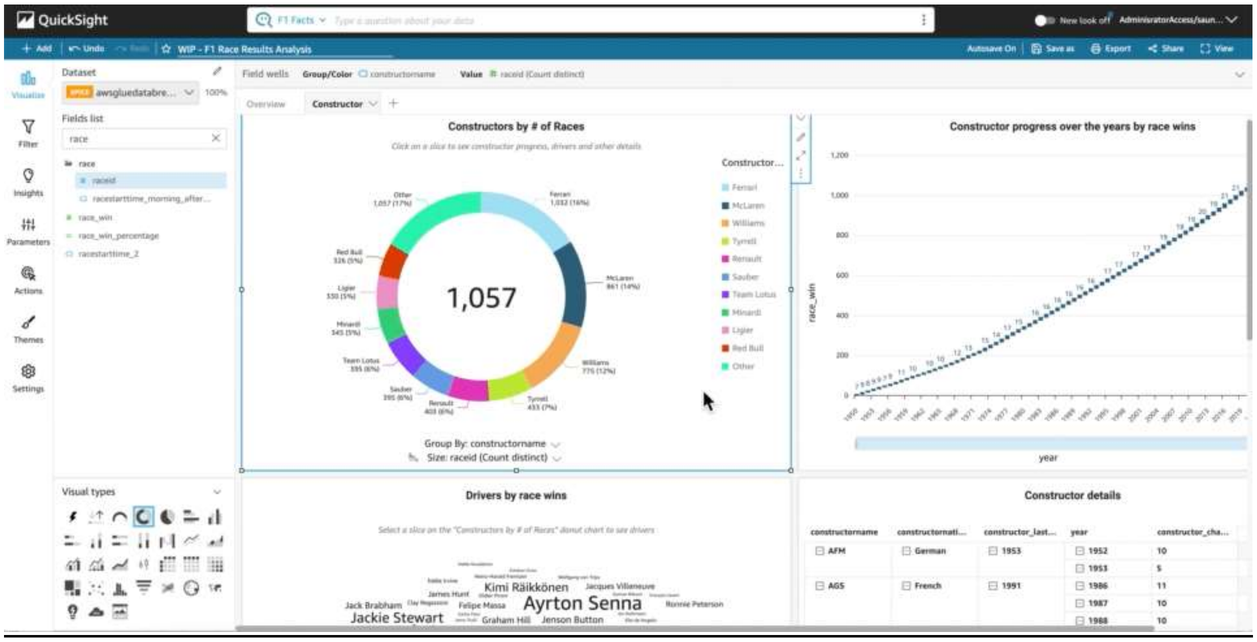


			1978	13	16
			1979	11	15
			1980	10	14
			1981	10	15
			1982	11	16
			1983	11	15
			1984	12	16
Adams	American	1950	1950	8	1

Constructors by # of race wins

Showing top 25 constructors

Group By: constructorname
Value: race_win_percentage (Custom)



Field wells Group/Color constructorname Value raceid (Count distinct)

constructorname	constructornati...	constructor_last...	year	constructor_cha...
AFM	German	1953	1952	10
			1953	5
AGS	French	1991	1986	11
			1987	10
			1988	10