



# Bay Area Bike Share Database



Final Project  
Group 4  
DATS6102 FALL 2016

# The Data

- Bay Area Bike Share, bike sharing system. 700 bikes and 67 stations throughout the San Francisco region
- Bay Area bikes are rented from and returned to any station in the system
- Operated through Motivate, an American firm focused on large-scale bike systems
- Year 3, September 2015 - August 2016

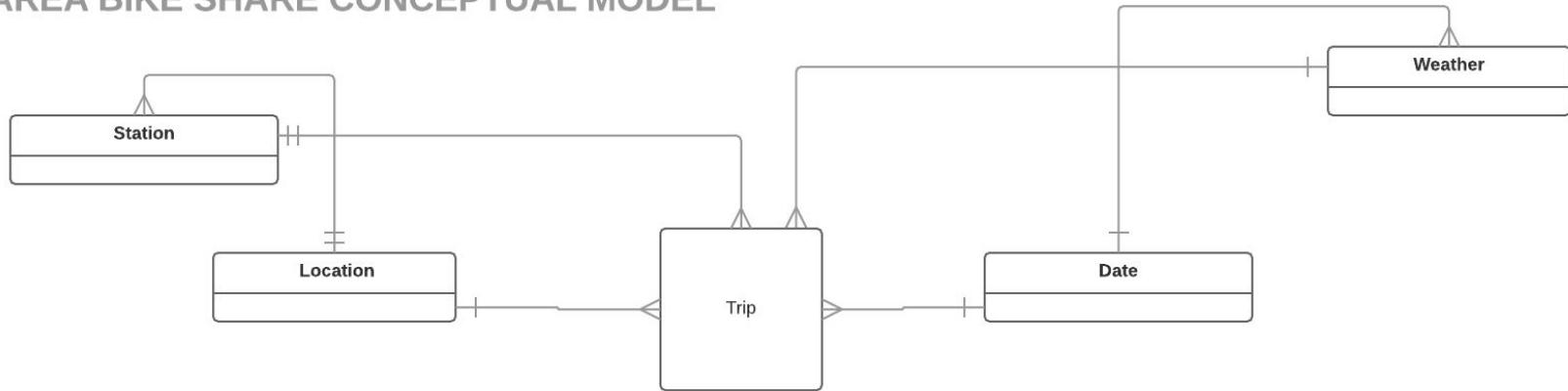


<http://www.bayareabikeshare.com/open-data>

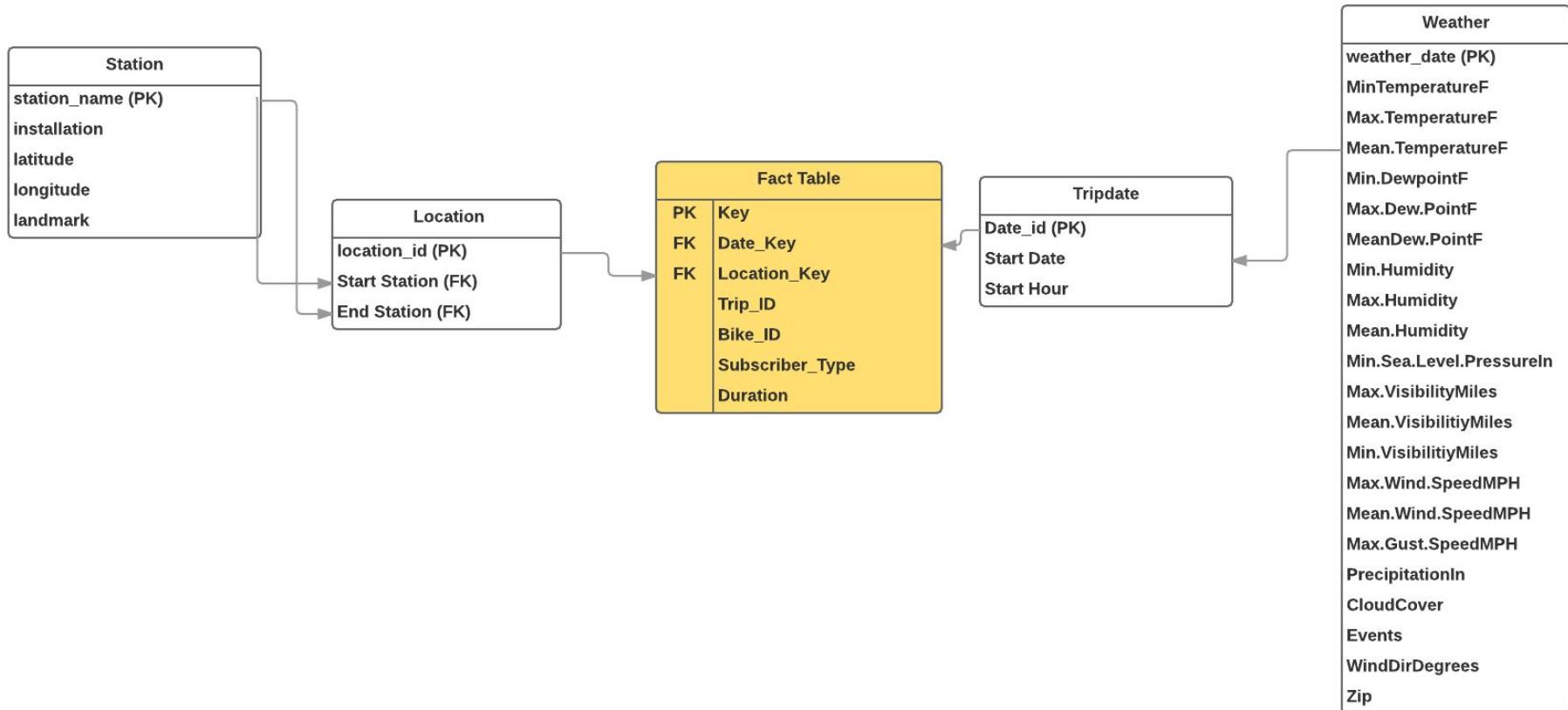
# Raw Data

- Raw data consisted of four Excel files
- Station, status, trip, weather
- Each trip is anonymized and includes: bike number, trip start/end day and time, start and end stations
- Plenty to work with, also can be easily understood

## BAY AREA BIKE SHARE CONCEPTUAL MODEL



# BAY AREA BIKE SHARE DIMENSIONAL MODEL



# The Grain

- A trip taken by one user (subscriber, customer) using Bike Share
- Transactional Grain

Fact Table	
PK	Key
FK	Date_Key
FK	Location_Key
	Trip_ID
	Bike_ID
	Subscriber_Type
	Duration



# Profiling - Excel

- Subscriber type, two unique identifiers of subscriber and customer ID, lots of nulls
- Station ID, unique but lacking completeness w/ lots of null values.

	A	B	C	D
1	station_id	bikes_avai	docks_ava	time
2	2	18	9	# #####
3	2	18	9	# #####
4	2	18	9	# #####
5	2	18	9	# #####
6	2	18	9	# #####
7	2	18	9	# #####
8	2	18	9	# #####
9	2	18	9	# #####
10	2	18	9	# #####
11	2	18	9	# #####
12	2	18	9	# #####
13	2	18	9	# #####
14	2	18	9	# #####
15	2	18	9	# #####
16	2	18	9	# #####
17	2	18	9	# #####
18	2	18	9	# #####
19	2	18	9	# #####

	A	B	C	D	E	F	G
1	station_id	name	lat	long	dockcount	landmark	installation
2	2	San Jose D	37.32973	-121.902	27	San Jose	8/6/2013
3	3	San Jose C	37.3307	-121.889	15	San Jose	8/5/2013
4	4	Santa Clar	37.33399	-121.895	11	San Jose	8/6/2013
5	5	Adobe on .	37.33142	-121.893	19	San Jose	8/5/2013
6	6	San Pedro	37.33672	-121.894	15	San Jose	8/7/2013
7	7	Paseo de S	37.3338	-121.887	15	San Jose	8/7/2013
8	8	San Salvad	37.33017	-121.886	15	San Jose	8/5/2013
9	9	Japantown	37.34874	-121.895	15	San Jose	8/5/2013
10	10	San Jose C	37.33739	-121.887	15	San Jose	8/6/2013
11	11	MLK Librar	37.33589	-121.886	19	San Jose	8/6/2013
12	12	SJSU 4th a	37.33281	-121.884	19	San Jose	8/7/2013
13	13	St James P	37.3393	-121.89	15	San Jose	8/6/2013
14	14	Arena Gre	37.33269	-121.9	19	San Jose	8/5/2013
15	16	SJSU - San	37.33396	-121.877	15	San Jose	8/7/2013
16	80	Santa Clar	37.3526	-121.906	15	San Jose	# #####
17	84	Ryland Par	37.34273	-121.896	15	San Jose	4/9/2014
18	89	S. Market :	37.3324	-121.89	19	San Jose	6/5/2016
19	88	5th S. at E.	37.33196	-121.882	19	San Jose	6/5/2016

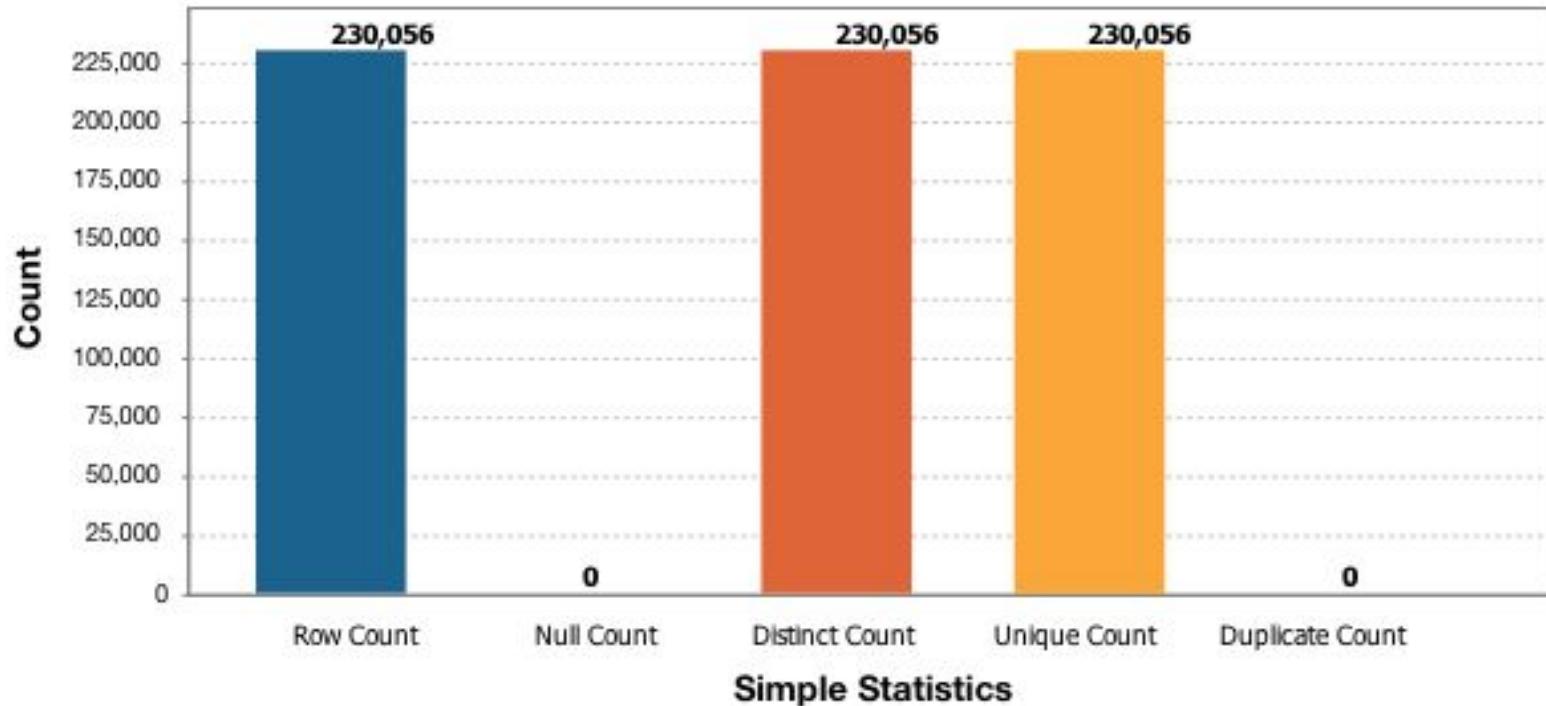
# Profiling - Excel

- Trip ID, all have uniqueness values, nulls galore

	A	B	C	D	E	F	G	H	I	J	K
1	Trip ID	Duration	Start Date	Start Station	Start Term	End Date	End Station	End Term	Bike #	Subscriber	Zip Code
2	913465	746	#####	San Francisco Caltrain 2 (330 Townsend)	69	#####	San Francisco City Hall	58	238	Subscriber	94107
3	913466	969	#####	Clay at Battery	41	#####	Washington at Kearny	46	16	Subscriber	94133
4	913467	233	#####	Davis at Jackson	42	#####	Commercial at Montgomery	45	534	Subscriber	94111
5	913468	213	#####	Clay at Battery	41	#####	Steuart at Market	74	312	Subscriber	94107
6	913469	574	#####	Steuart at Market	74	#####	San Francisco Caltrain 2 (330 Townsend)	69	279	Subscriber	94107
7	913470	623	#####	San Jose Diridon Caltrain Station	2	#####	Japantown	9	261	Subscriber	95112
8	913471	746	#####	Embarcadero at Bryant	54	#####	Powell Street BART	39	436	Subscriber	94103
9	913472	1038	#####	Townsend at 7th	65	#####	Howard at 2nd	63	607	Subscriber	94107
10	913473	424	#####	Market at 10th	67	#####	Townsend at 7th	65	259	Subscriber	94102
11	913474	633	#####	Embarcadero at Bryant	54	#####	Embarcadero at Sansome	60	613	Subscriber	94105
12	913475	174	#####	2nd at Folsom	62	#####	Market at Sansome	77	449	Subscriber	94107
13	913476	777	#####	Davis at Jackson	42	#####	San Francisco Caltrain (Townsend at 4th)	70	86	Subscriber	94111
14	913477	454	#####	Powell Street BART	39	#####	San Francisco Caltrain 2 (330 Townsend)	69	440	Subscriber	94107
15	913478	228	#####	Washington at Kearny	46	#####	Market at Sansome	77	351	Subscriber	94133
16	913479	445	#####	Civic Center BART (7th at Market)	72	#####	San Francisco Caltrain (Townsend at 4th)	70	315	Subscriber	94103
17	913480	649	#####	5th at Howard	57	#####	Steuart at Market	74	559	Subscriber	94112
18	913481	459	#####	San Francisco Caltrain (Townsend at 4th)	70	#####	2nd at Folsom	62	579	Subscriber	94002
19	913482	337	#####	Townsend at 7th	65	#####	2nd at Townsend	61	287	Subscriber	94107

# Profiling - Talend

**Trip\_ID**  
All  
contain  
unique  
values



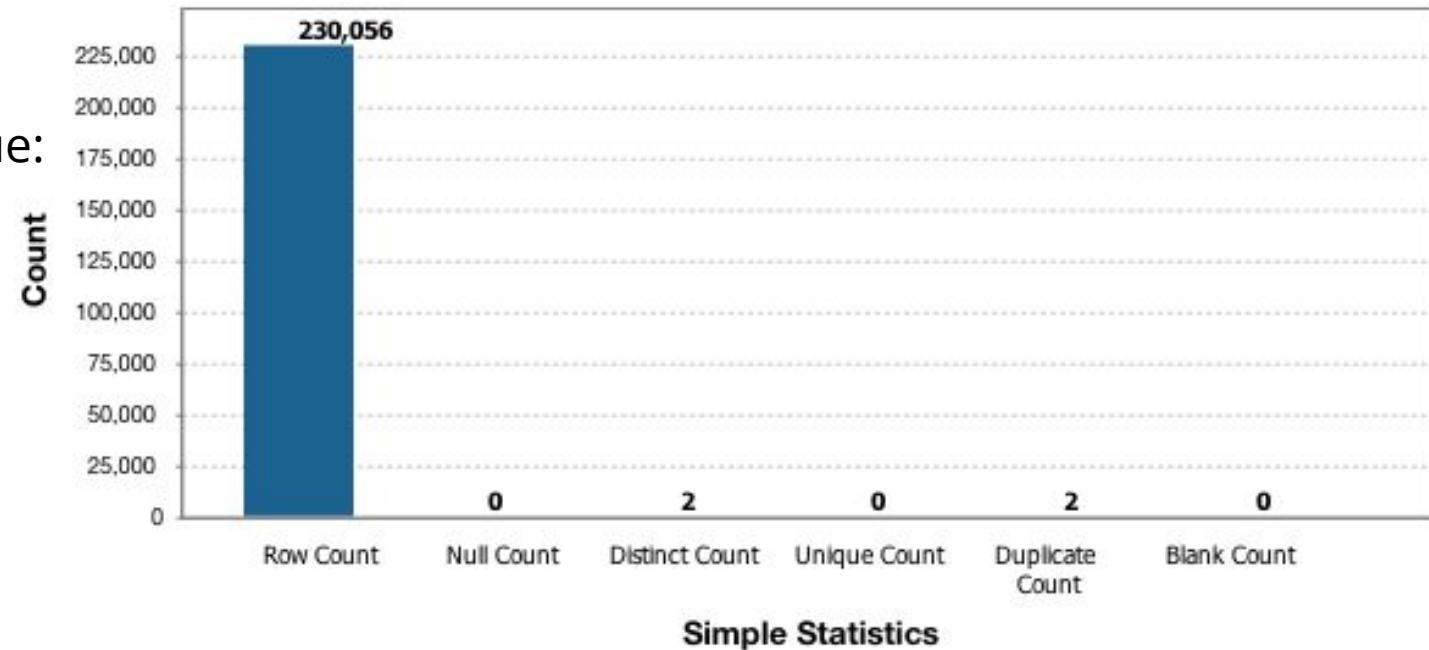
# Profiling - Talend

## Subscriber Type

2 Distinct, unique:

-Subscriber

-Customer

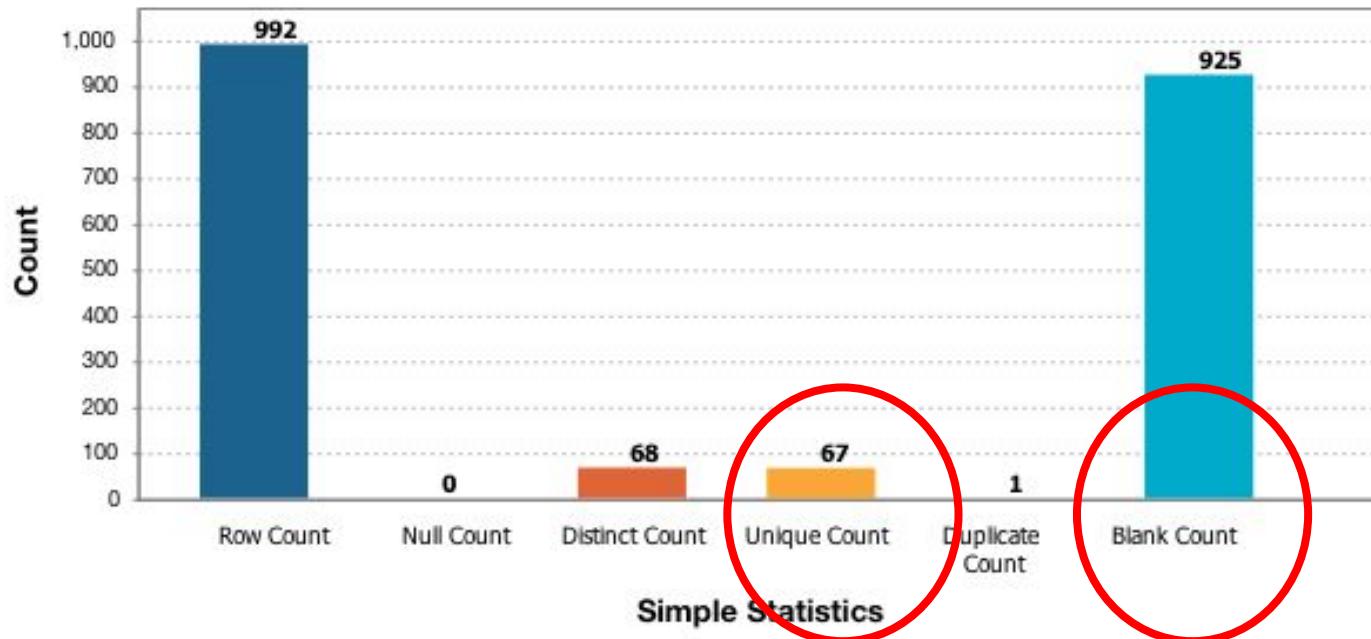


# Profiling - Talend

## Station ID

-Blank values  
confusing data

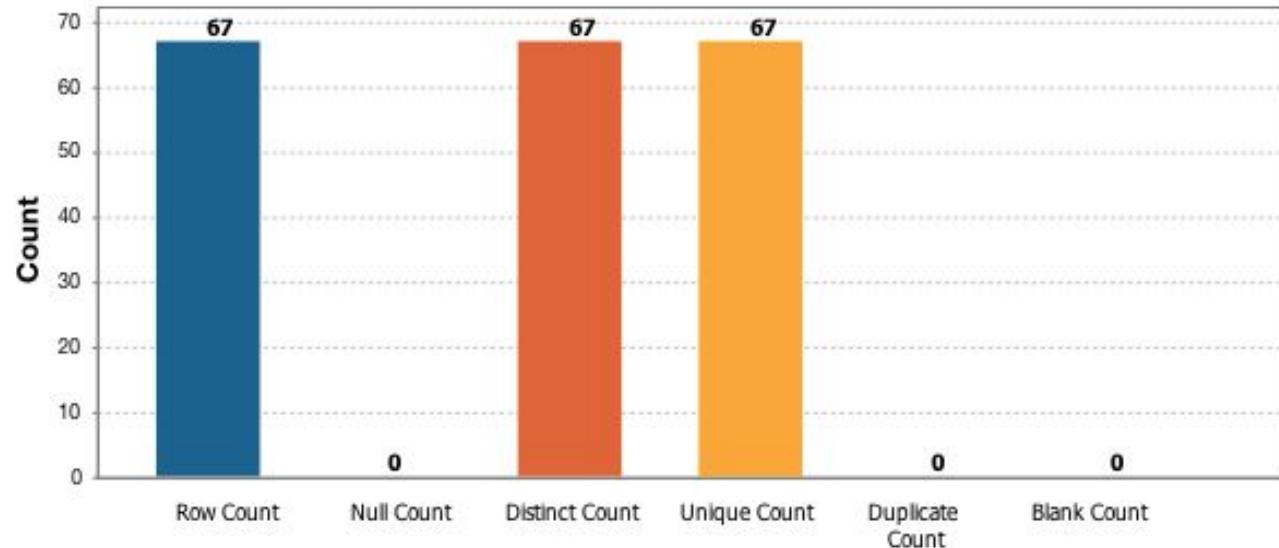
-67 unique  
stations



# Profiling - Talend

## Station ID

-Removed blank values in  
FileDelimited connection



# Data Quality Scorecard

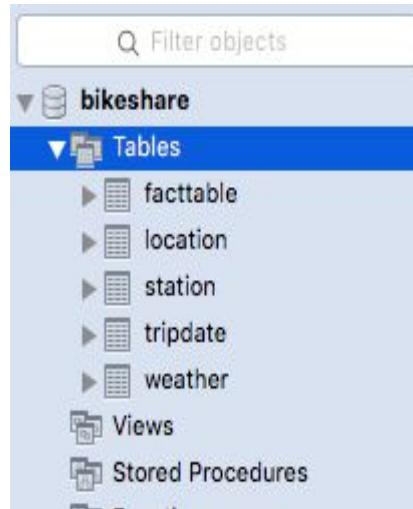
# Extract, Transform, Load

# 1. Create Database & Tables in MySQL

The screenshot shows the MySQL Workbench interface. On the left, the 'Query 1' tab contains the following SQL code:

```
1 • CREATE SCHEMA bikeshare;
2
3 • use bikeshare;
4
5 • CREATE TABLE weather(
6     weather_date VARCHAR(45),
7     Max_TemperatureF INT,
8     Mean_TemperatureF INT,
9     Min_TemperatureF INT,
10    Max_Dew_PointF INT,
11    MeanDew_PointF INT,
12    Min_DewpointF INT,
13    Max_Humidity INT,
14    Mean_Humidity INT,
15    Min_Humidity INT,
16    Max_Sea_Level_PressureIn FLOAT,
17    Mean_Sea_Level_PressureIn FLOAT,
18    Min_Sea_Level_PressureIn FLOAT,
19    Max_VisibilityMiles INT,
20    Mean_VisibilityMiles INT,
21    Min_VisibilityMiles INT,
22    Max_Wind_SpeedMPH INT,
23    Mean_Wind_SpeedMPH INT,
24    Max_Gust_SpeedMPH INT,
25    CloudCover INT,
26    WindDirDegrees INT,
27    PRIMARY KEY(weather_date)
28 );
29
30
31
32 • CREATE TABLE station(
33     station_name VARCHAR(45),
34     latitude FLOAT
```

A red circle highlights the first line of code, 'CREATE SCHEMA bikeshare;'. A large grey arrow points from the query results area to the object browser on the right.



## 2. Create Connection in talend (MySQL)

New Database Connection on repository - Step 1/2

⚠ It is inadvisable to leave the purpose blank.

Name	bikeshareexample
Purpose	
Description	
Author	user@talend.com
Locker	
Version	0.1
Status	
Path	

### New Database Connection on repository - Step 2/2

Define the connection parameters

DB Type MySQL

Db Version MySQL 5

String of Connection `jdbc:mysql://localhost:3306/bikeshare?noDatetimeStringSync=true`

Login root

Password \*\*\*\*

Server localhost

Port 3306

DataBase bikeshare

Additional parameters `noDatetimeStringSync=true`

**Check** (button circled in red)

Database Properties

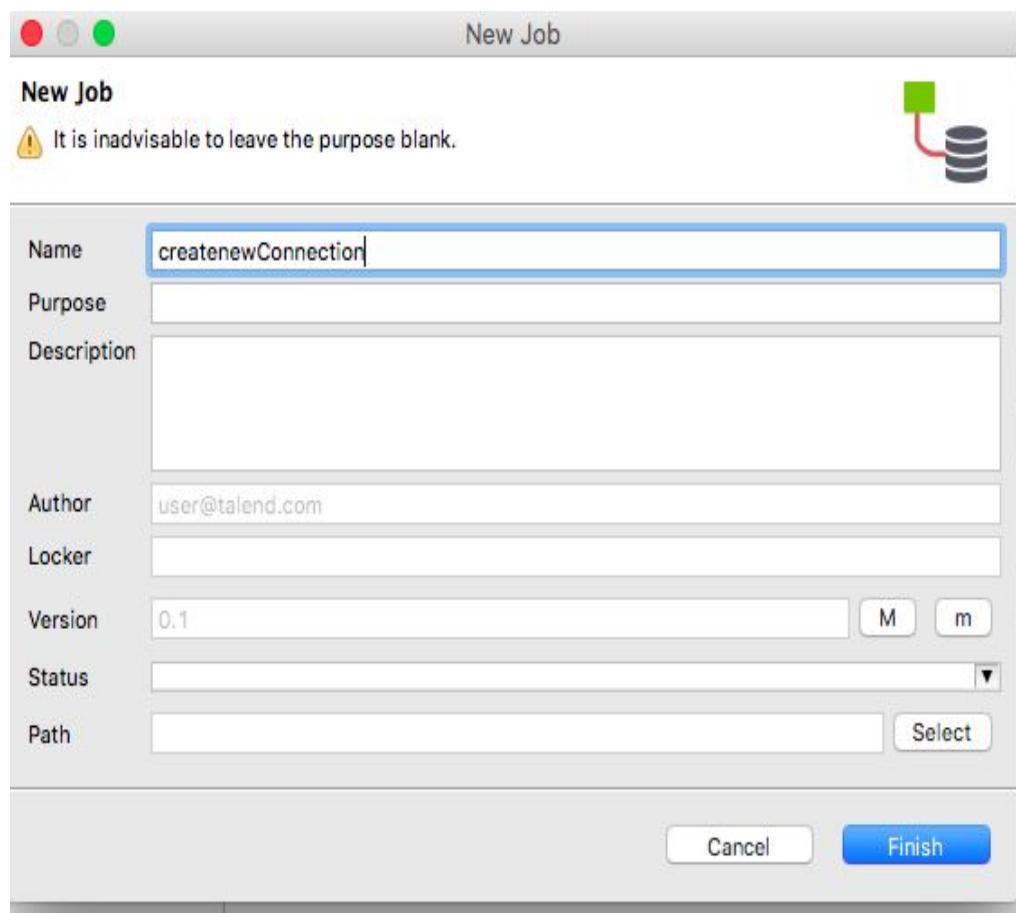
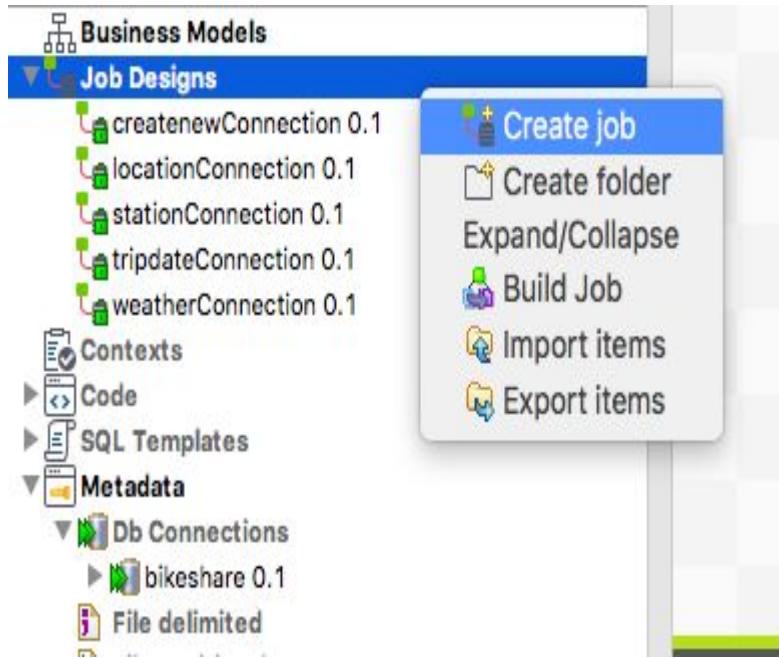
Check Connection

"bikeshareexample" connection successful.

SQL Syntax SQL

OK

# 3. Create workspace in Talend



# 4. Create Excel file

## File - Step 1 of 4

 It is inadvisable to leave the purpose blank.

Name	tripdateexample
Purpose	
Description	
Author	user@talend.com
Locker	
Version	0.1
Status	
Path	

File - Step 2 of 4  
Add a Metadata File on repository  
Define the path of the file and the format settings 

**File Settings**

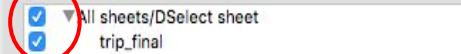
Server: Localhost 127.0.0.1

File: /Users/ann/Documents/GW life/data warehousing/final project/trip\_final.xlsx 

Read excel2007 file format(xlsx)

Generation mode: Memory-consuming(User mode) 

**File Viewer and Sheets setting**

Set sheets parameters 

Please select sheet (Sheet structure as schema guide) trip\_final 

All sheets/DSel sheet  
 trip\_final

A	B	C	D	E	F	G	H	I
	Trip.ID	Duration	Start.St...	Start.Ter...	End.Stat...	End.Ter...	Bike..	Subscri...
5.0	913469.0	574.0	Steuart...	74.0	San Fran...	69.0	279.0	Subscriber
7.0	913471.0	746.0	Embarca...	54.0	Powell S...	39.0	436.0	Subscriber
9.0	913473.0	424.0	Market a...	87.0	Townse...	65.0	259.0	Subscriber
10.0	913474.0	633.0	Embarca...	54.0	Embarca...	60.0	613.0	Subscriber
16.0	913480.0	649.0	5th at H...	57.0	Steuart...	74.0	559.0	Subscriber
19.0	913483.0	518.0	Townse...	65.0	South V...	66.0	457.0	Subscriber
20.0	913484.0	352.0	Tempora...	55.0	2nd at T...	61.0	589.0	Subscriber
22.0	913486.0	314.0	Rengsto...	33.0	San Ant...	29.0	118.0	Subscriber
24.0	913488.0	274.0	Washing...	46.0	Market a...	76.0	329.0	Subscriber
27.0	913491.0	509.0	Harry Br...	50.0	2nd at T...	61.0	413.0	Subscriber
39.0	913507.0	613.0	San Fran...	70.0	Tempora...	55.0	548.0	Subscriber
40.0	913510.0	270.0	2nd at T...	61.0	San Fran...	70.0	502.0	Subscriber
46.0	913518.0	390.0	Broadwa...	82.0	Embarca...	51.0	496.0	Subscriber

## File - Step 3 of 4

Add a Metadata File on repository  
Define the setting of the parse job

## File - Step 4 of 4

Add a Schema on repository  
Define the Schema



## File Settings

Encoding

Advanced separator(for number)

Thousands separator:

Decimal separator:

## Metadata column setting

First column:

Last column:

## Rows To Skip

If any rows must be ignored

Header  2

Footer

## Name

Comment

## Limit Of Rows

If the number of lines must be limited

Limit

## Schema

Click to update schema preview

Guess

## Preview

## Output

Set heading row as column names

[Refresh Preview](#)

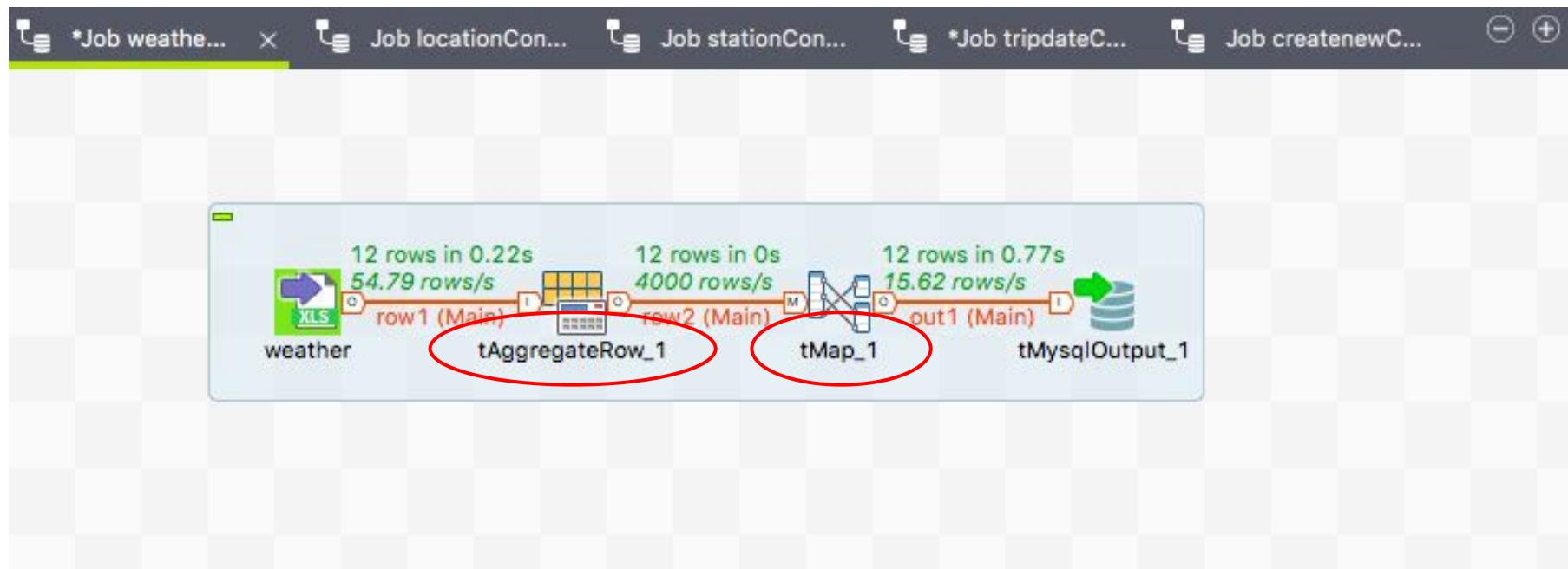
Column 0	Trip.ID	Duration	Start.Station	Start.Terminus	End.Station
5	913469	574	Steuart at Market	74	San Franci
7	913471	746	Embarcadero at Bryant	54	Powell Stri
9	913473	424	Market at 10th	67	Townsend
10	913474	633	Embarcadero at Bryant	54	Embarcadi
16	913480	649	5th at Howard	57	Steuart at
19	913483	518	Townsend at 7th	65	South Van
20	913484	352	Temporary Transbay Terminal (Howard at Beale)	55	2nd at Tov
22	913486	314	Rengstorff Avenue / California Street	33	San Anton
24	913488	274	Washington at Kearny	46	Market at

## Description of the Schema

Column	key	Type	Nullab	Date Pattern (Ctrl+Space)	Length	Precision	Default	Comment
Column0	<input checked="" type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		3	0		
_13469	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		6	0		
_74	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		4	0		
Steuart_at_Market	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		45	0		
_4	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		2	0		
San_Francisco_Caltrain_2__330_T...	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		45	0		
_9	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		2	0		
_79	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		3	0		
Subscriber	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		10	0		
_4107	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		5	0		
Wed Sep 09 00 00 00 EDT 2015	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		28	0		

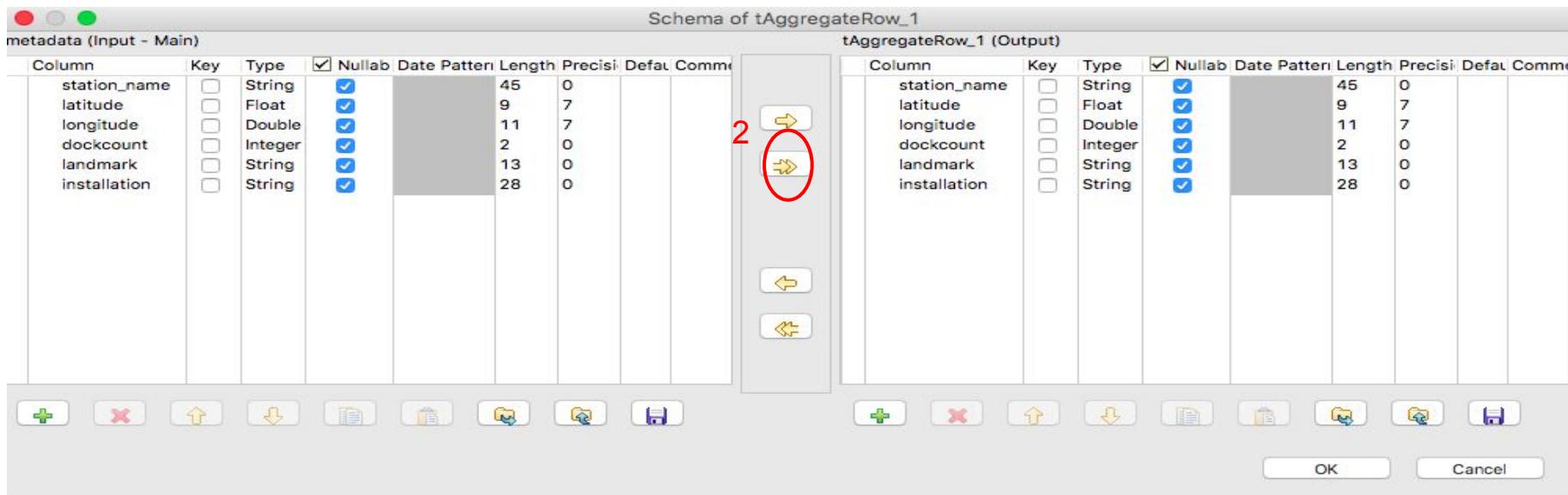
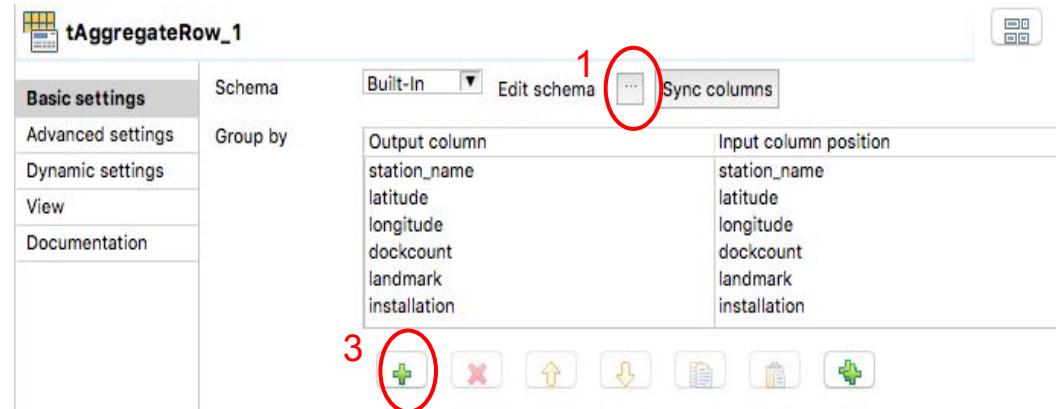


# 5. Drag to workspace



# ETL

## process-tAggregate



# ETL process-tMap

Screenshot of a tMap component in Talend ETL tool.

The interface shows three main sections:

- row2** (Input): Contains columns: station\_name, latitude, longitude, dockcount, landmark, installation.
- Var** (Mapping): A central area showing the mapping between input and output columns.
- out1** (Output): Contains columns: station\_name, latitude, longitude, dockcount, landmark, installation.

The "Auto map!" button in the Var section is circled in red.

Below the main sections, there are two schema editors:

- Schema editor** (row2): Shows the schema for the input rows.
- Expression editor** (out1): Shows the schema for the output rows.

Both schema editors include columns for:

Column	Type	Key	Nullab	Date Pattern (Ctrl+E)	Length	Precision	Default	Comment
station_name	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>		45	0		
latitude	Float	<input type="checkbox"/>	<input checked="" type="checkbox"/>		9	7		
longitude	Double	<input type="checkbox"/>	<input checked="" type="checkbox"/>		11	7		
dockcount	Integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>		2	0		
landmark	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>		13	0		
installation	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>		28	0		

At the bottom, there are several icons for managing the tMap component.

# Talend weather

The screenshot shows the Talend Studio interface with the following details:

- Repository View:** On the left, under "Job Designs", the "weatherConnection 0.1" job is selected (highlighted in blue).
- Job Designer View:** The main workspace displays a job flow:
  - Input: A green "XLS" component labeled "weather".
  - Process:
    - A "tAggregateRow\_1" component with two rows: "row1 (Main)" and "row2 (Main)".
    - An "tMap\_1" component.
    - An "tMysqlOutput\_1" component.
  - Performance Metrics: "12 rows in 0.22s", "54.79 rows/s", "12 rows in 0s", "4000 rows/s", "12 rows in 0.77s", and "15.62 rows/s".
- Job Execution Log:** Below the designer, the "Job weatherConnection" window shows the execution log. The "Execution" tab is active, featuring a red circle around the "Run" button. The log output is:

```
Starting job weatherConnection at 13:59 04/12/2016.  
[statistics] connecting to socket on port 3857  
[statistics] connected  
Duplicate entry '4/4/16' for key 'PRIMARY'  
[statistics] disconnected  
Job weatherConnection ended at 13:59 04/12/2016. [exit code=0]
```
- Component Catalog:** A sidebar on the right lists various components including Db Connections, File types (Excel, delimited, positional, regex, xml), and other Talend services like MDM, Web Service, and FTP.

# 6. Result in MySQL (weather)

The screenshot shows the MySQL Workbench interface with a query results grid. The left sidebar contains navigation links for Management, Instance, Performance, Schemas, and a central search bar. The main area has two tabs: 'Query 1' and 'SQL File 5\*'. The SQL tab displays the following code:

```
1 • select *from location;
2 • select *from weather;
3 • select *from station;
4 • select *from tripdate;
5
6 ✘ use bikeshare;
7 • select * from facttable JOIN tripdate JOIN weather ON facttable.date_key=tripdate.date_id AND tripdate.start_date=weather.date
8 • select * from facttable JOIN location JOIN station ON facttable.location_key=location.location_id AND location.start_
9 • select * from facttable;
```

The results grid shows data for the 'weather' table, with columns: weather\_date, Max\_TemperatureF, Mean\_TemperatureF, Min\_TemperatureF, Max\_Dew\_PointF, MeanDew\_PointF, Min\_DewpointF, Max\_Humidity, Mean\_Humidity, and Min\_Humidity. The data includes various dates from 2015 to 2016, along with their corresponding temperature and humidity values.

weather_date	Max_TemperatureF	Mean_TemperatureF	Min_TemperatureF	Max_Dew_PointF	MeanDew_PointF	Min_DewpointF	Max_Humidity	Mean_Humidity	Min_Humidity
1/1/16	52	44	35	30	23	15	64	47	29
10/10/15	72	67	61	61	60	58	93	81	68
11/11/15	61	50	39	47	42	39	89	72	55
12/12/15	57	48	39	49	43	37	86	71	55
2/2/16	58	50	42	46	41	37	93	77	61
3/3/16	65	62	59	57	53	49	84	70	56
4/4/16	68	61	54	52	50	48	86	68	49
5/5/16	68	63	57	52	50	49	80	68	56
6/6/16	68	62	55	54	53	51	86	75	63
7/7/16	65	61	57	53	52	51	80	72	63
8/8/16	73	65	57	56	55	54	93	75	57
9/9/15	93	77	60	58	50	44	73	47	21
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The right sidebar features a vertical stack of icons for different result types: Result Grid, Form Editor, Field Types, and Query Stats.

# Result in MySQL (location)

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

SCHEMAS

- Filter objects
- Tables
  - facttable
    - Columns
    - Indexes
    - Foreign Keys
    - Triggers
  - location
    - Columns

DIRECTORY

- Tables
  - facttable
    - Columns
    - Indexes
    - Foreign Keys
    - Triggers
  - location
    - Columns

Query 1 SQL File 5\*

Limit to 2000 rows

```
1 • select * from location;
2 • select * from weather;
3 • select * from station;
4 • select * from tripdate;
5
6 ✘ use bikeshare;
7 • select * from facttable JOIN tripdate JOIN weather ON facttable.date_key=tripdate.date_id AND tripdate.start_
8 • select * from facttable JOIN location JOIN station ON facttable.location_key=location.Location_id AND location_
9 • select * from facttable;
```

Result Grid

location_id	start_station	end_station
1	Steuart at Market	San Francisco Caltrain 2 (330 Townsend)
2	Embarcadero at Bryant	Powell Street BART
3	Market at 10th	Townsend at 7th
4	Embarcadero at Bryant	Embarcadero at Sansome
5	5th at Howard	Steuart at Market
6	Townsend at 7th	South Van Ness at Market
7	Temporary Transbay Terminal (Howard at Beale)	2nd at Townsend
8	Rengstorff Avenue / California Street	San Antonio Caltrain Station
9	Washington at Kearny	Market at 4th
10	Harry Bridges Plaza (Ferry Building)	2nd at Townsend
11	San Francisco Caltrain (Townsend at 4th)	Temporary Transbay Terminal (Howard...)
12	2nd at Townsend	San Francisco Caltrain (Townsend at 4th)
13	Broadway St at Battery St	Embarcadero at Folsom
14	Embarcadero at Sansome	Temporary Transbay Terminal (Howard...)
15	Davis at Jackson	Post at Kearny
16	San Francisco Caltrain 2 (330 Townsend)	Market at Sansome
17	San Francisco Caltrain (Townsend at 4th)	Embarcadero at Sansome

location 23

Apply

# Result in MySQL (station)

The screenshot shows the MySQL Workbench interface with the following details:

- Left Sidebar (Management):** Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore.
- Left Sidebar (Instance):** Startup / Shutdown, Server Logs, Options File.
- Left Sidebar (Performance):** Dashboard, Performance Reports, Performance Schema Setup.
- Left Sidebar (Schemas):** Filter objects, Tables, facttable (selected), Columns, Indexes, Foreign Keys, Triggers, location, Columns, location\_id.
- Query Editor:** Contains the following SQL code:

```
1 select * from location;
2 select * from weather;
3 select * from station;
4 select * from tripdate;
5
6 use bikeshare;
7 select * from facttable JOIN tripdate JOIN weather ON facttable.date_key=tripdate.date_id AND tripdate.start_time_key=weather.id;
8 select * from facttable JOIN location JOIN station ON facttable.location_key=location.location_id AND location_id=station.id;
9 select * from facttable;
```
- Result Grid:** Shows the results of the last query, displaying columns: station\_name, latitude, longitude, dockcount, landmark, installation. The data includes rows for various bike stations in San Francisco and San Jose, such as "2nd at Folsom", "2nd at South Park", "2nd at Townsend", etc., with their respective coordinates and details.

# Result in MySQL (tripdate)

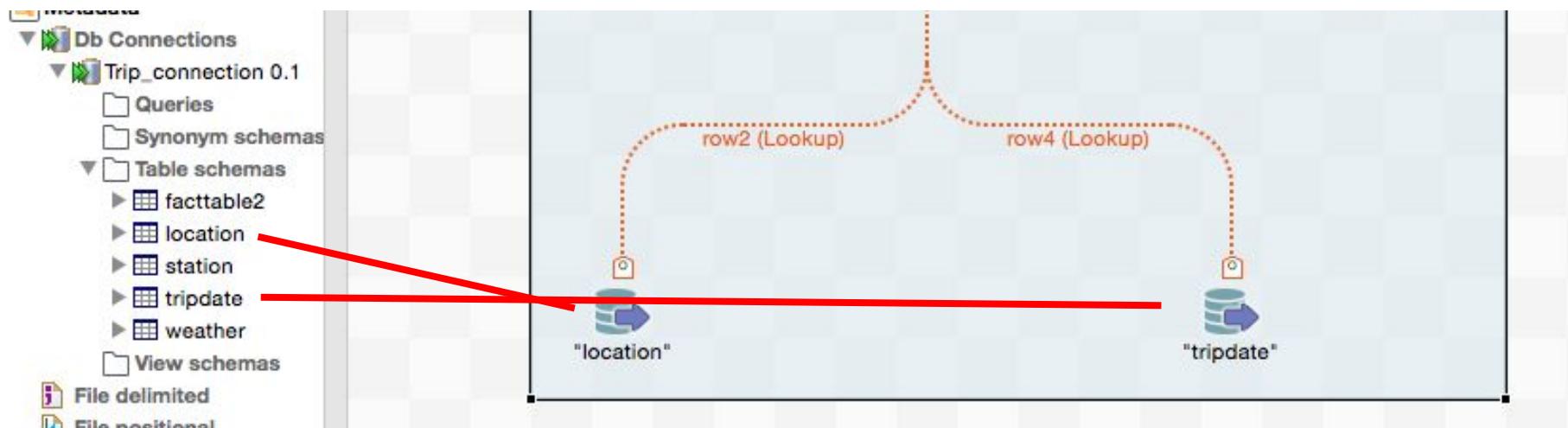
The screenshot shows a MySQL Workbench interface with the following details:

- Query Editor: The query `select * from tripdate;` is entered.
- Status Bar: Shows 100% completion, 24:33 execution time, and 1 error found.
- Result Grid: The results are displayed in a grid format with three columns: date\_id, start\_date, and start\_hour. The data is as follows:

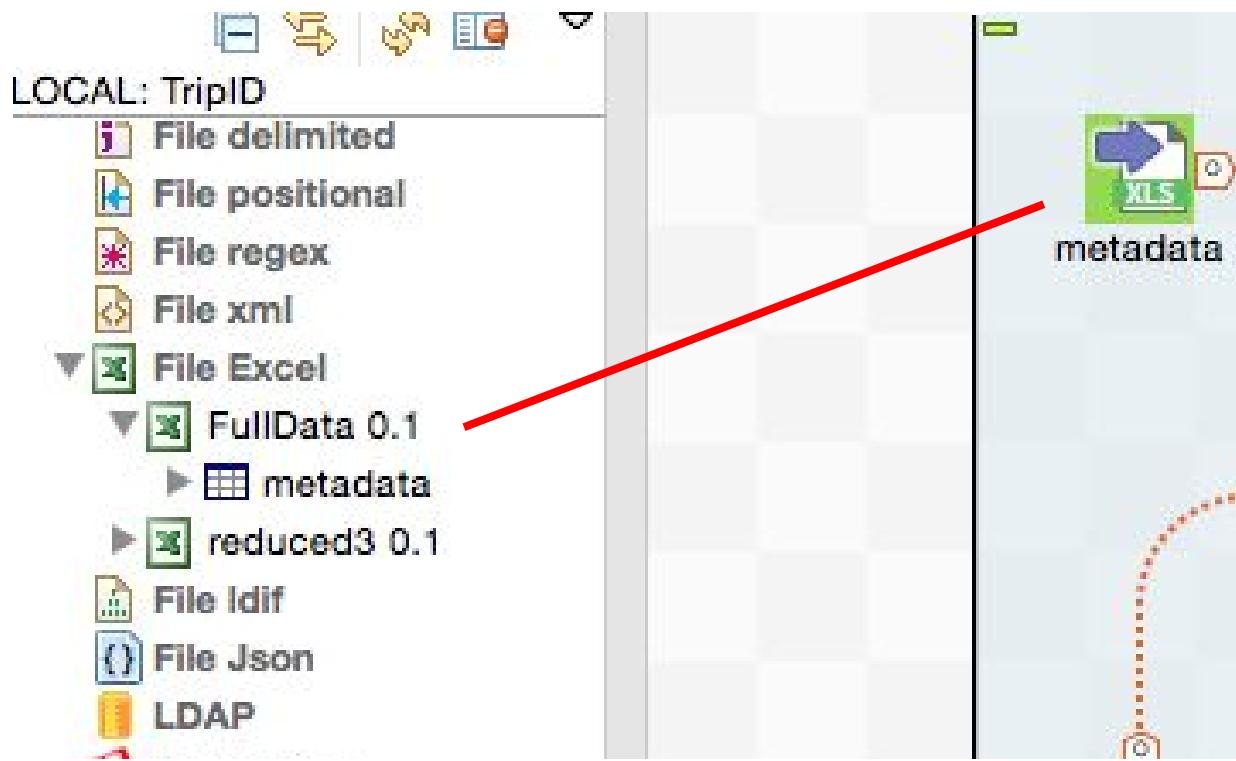
date_id	start_date	start_hour
1	9/9/15	0
2	9/9/15	1
3	9/9/15	2
4	9/9/15	3
5	9/9/15	4
6	9/9/15	5
7	9/9/15	6
8	9/9/15	7

# ETL for the Fact Table

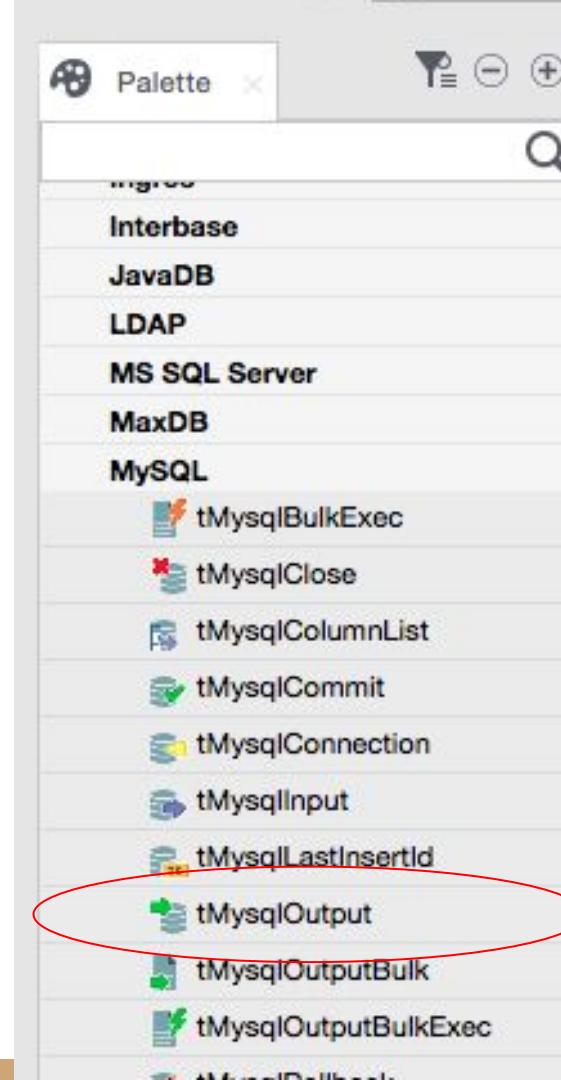
# Establish connection with MySQL db



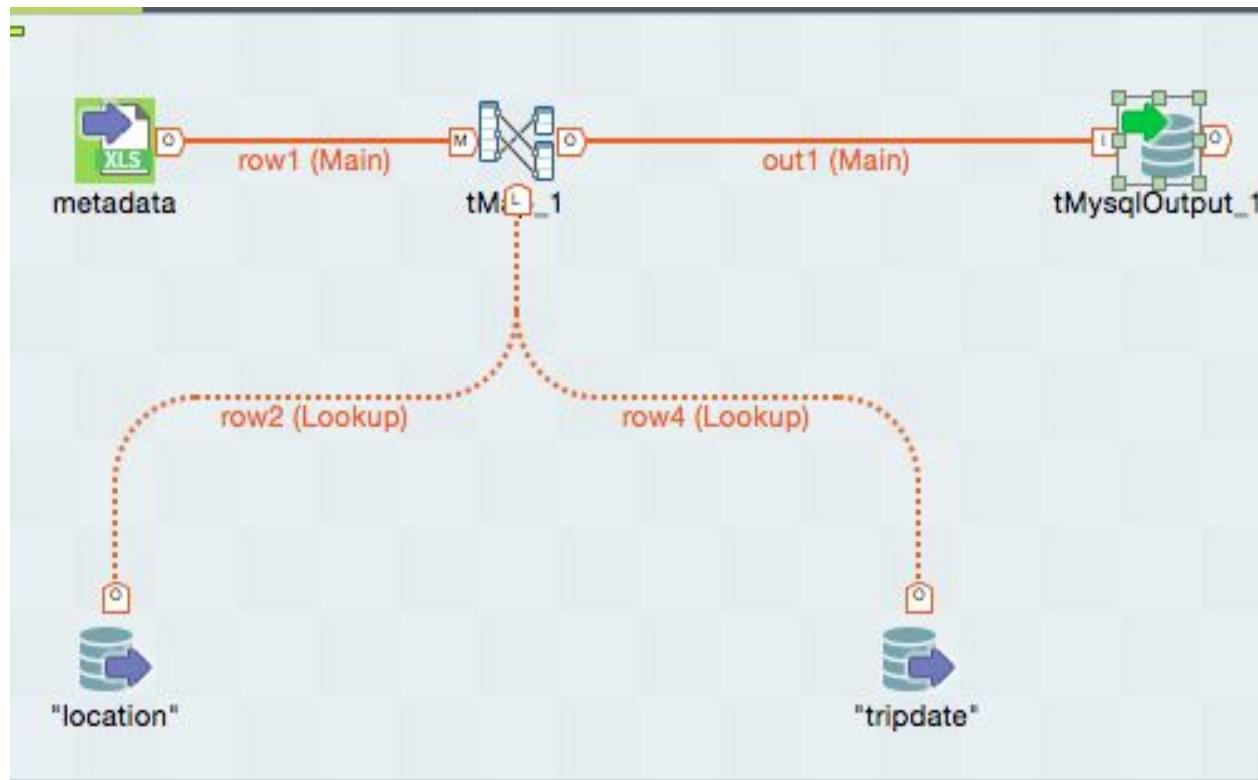
# Establish metadata connection



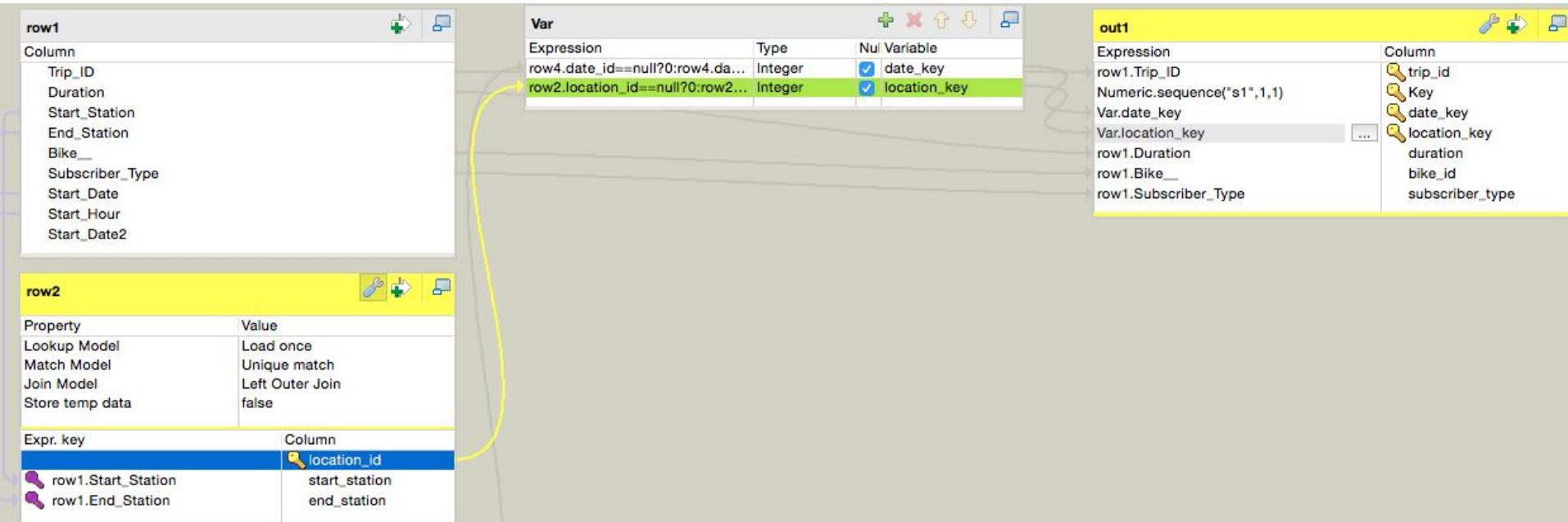
# MySQL output



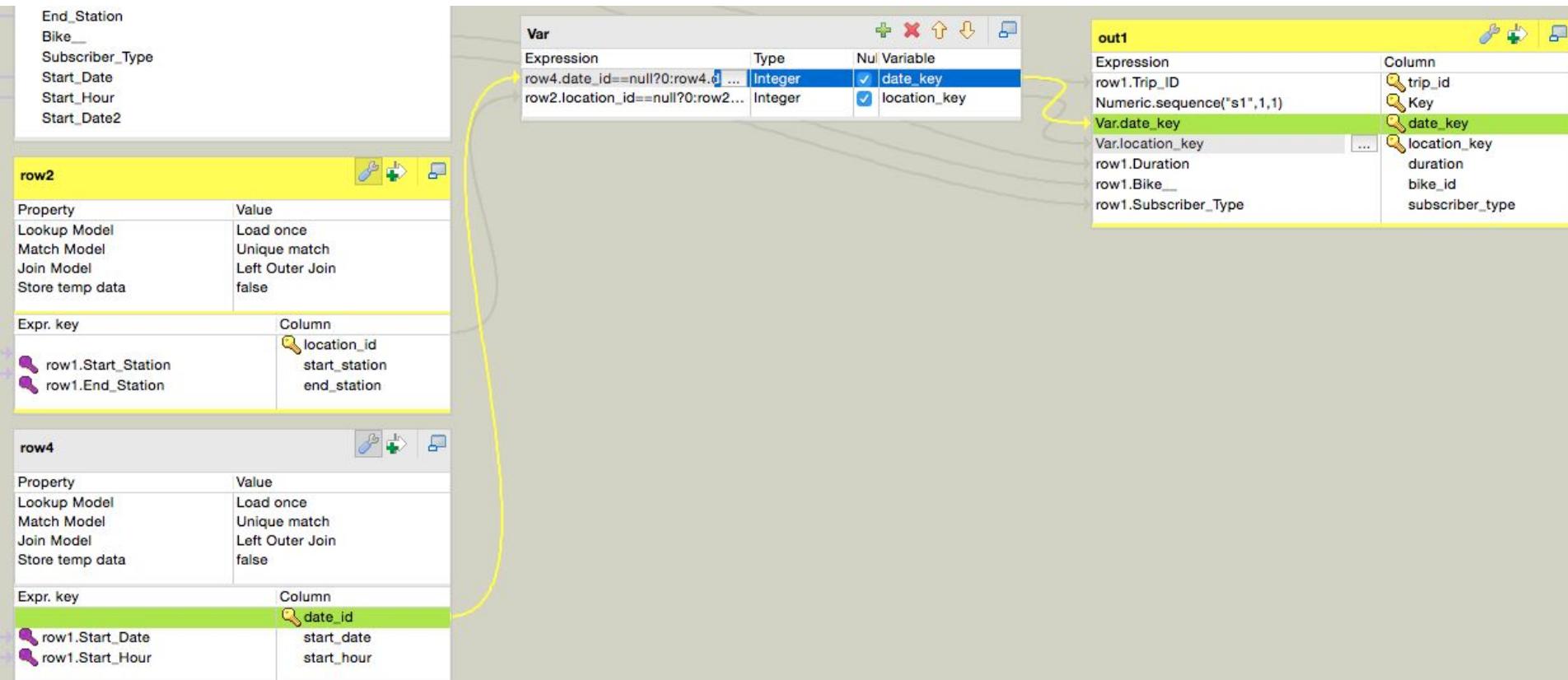
# Add tMap Processing Function



# Location key



# Date key



# Joining metadata and MySql database

The screenshot shows a data integration tool's configuration interface. It includes three main sections: **row1**, **row2**, and **Expr. key**.

**row1:** A list of columns. The "Start\_Station" column is highlighted with a green background.

Trip_ID
Duration
Start_Station
End_Station
Bike_
Subscriber_Type
Start_Date
Start_Hour
Start_Date2

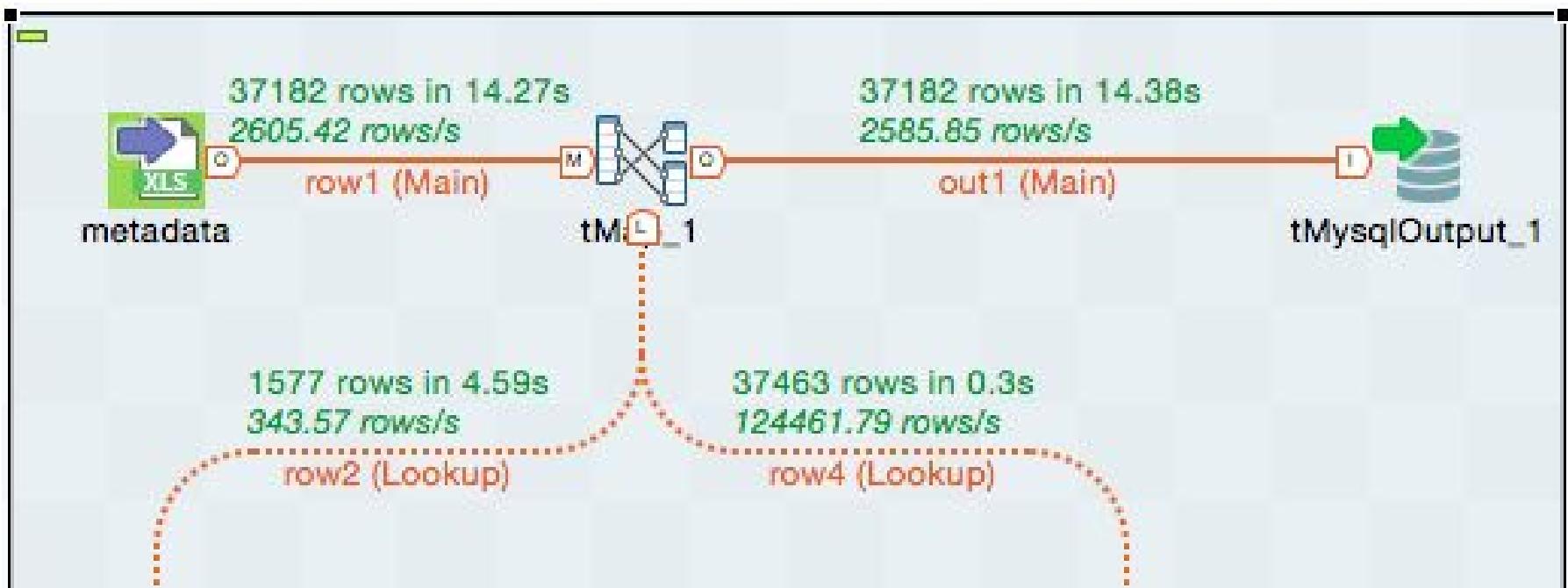
**row2:** A table of properties and their values.

Property	Value
Lookup Model	Load once
Match Model	Unique match
Join Model	Left Outer Join
Store temp data	false

**Expr. key:** A section for mapping expressions to columns. It contains two rows:

Expr. key	Column
row1.Start_Station	location_id
row1.End_Station	start_station
	end_station

# After running the job



# MySQL output

31 ✖ |select \* from facttable2;

100% ◇ 1:31 1 error found

Result Grid | Filter Rows: | Search | Edit: □

	trip_id	date_key	location_key	duration	bike_id	subscriber_type	Key
▶	913469	2	1	574	279	Subscriber	1
	913471	3	2	746	436	Subscriber	2
	913473	5	3	424	259	Subscriber	3
	913474	5	4	633	613	Subscriber	4
	913480	6	5	649	559	Subscriber	5
	913483	7	6	518	457	Subscriber	6
	913484	7	7	352	589	Subscriber	7
	913486	7	8	314	118	Subscriber	8
	913488	7	9	274	929	Subscriber	9

# Talend benefits and problems

## Benefits

- When we change the settings of MySQL tables, we can easily reload all the data into Talend.
- Talend handles large source files.
- Eventually, after debugging, we got the results we wanted.

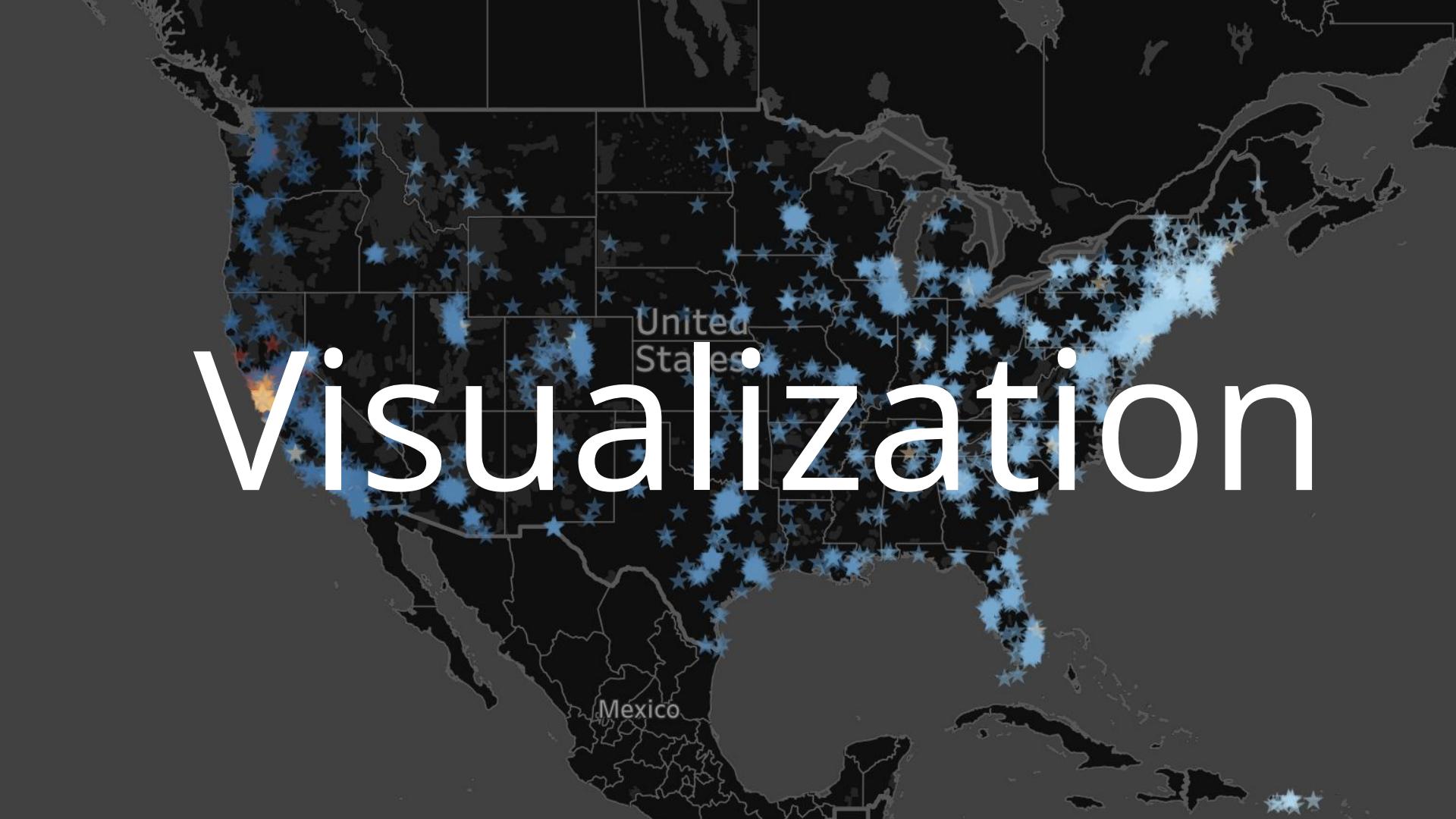
## Problems

- Not intuitive when attempting to configure
- Many errors during when running the job, and not easy to fix
  - Errors during ELT process are understandable only for someone experienced in Java
  - Troubleshooting process alone took 3 days

# Benefits and problems cont.

- Explains why relationship between Business and IT is so important (we were acting as both in this project)
- Those skilled in ETL can save hours for those in analytics, BI, etc.
- Worth investing in a configurable ETL tool
- Familiarity with SQL made the process move quicker

# Visualization



# Connect MySQL to Tableau

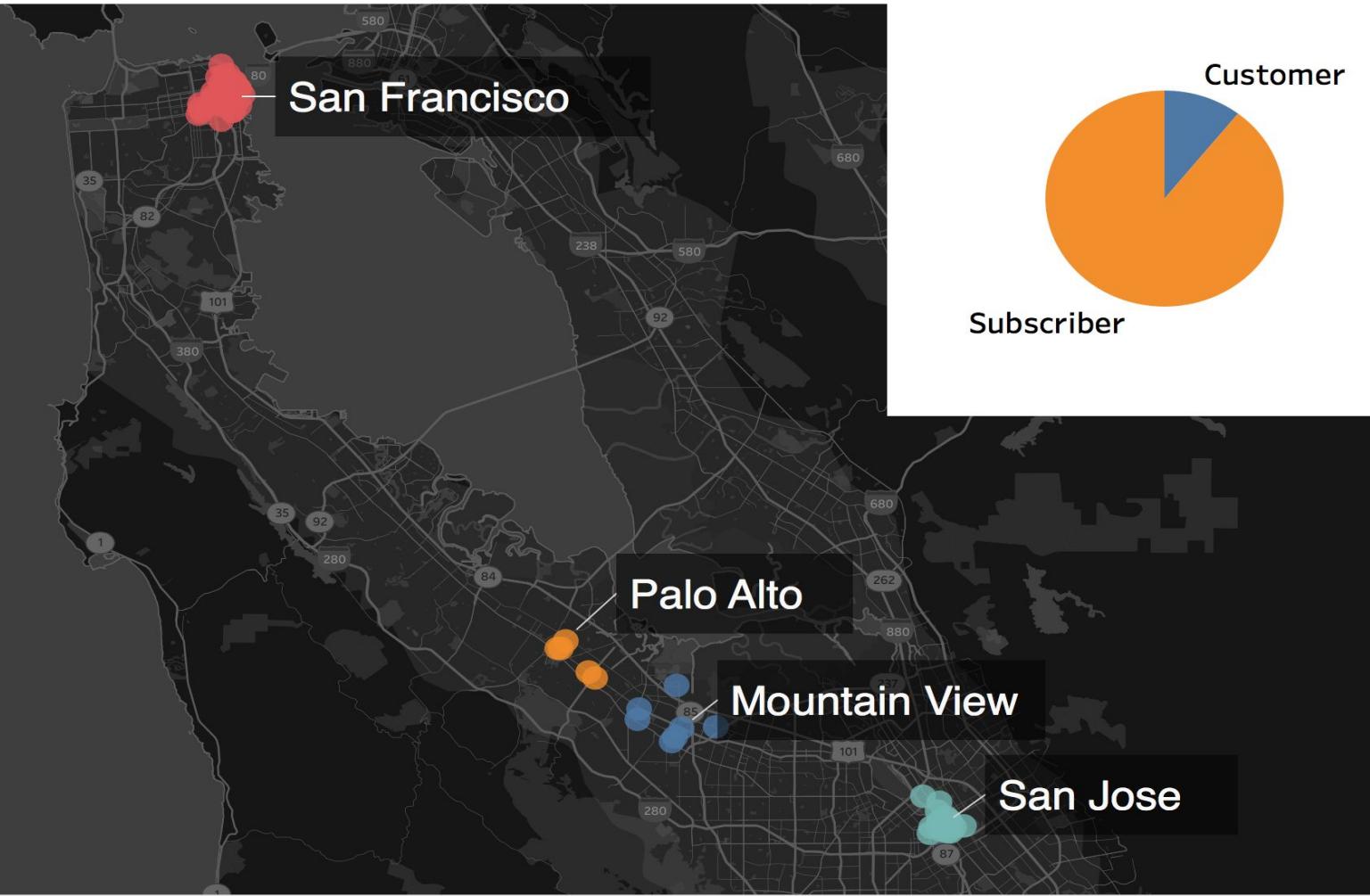
The screenshot shows the Tableau Data Source setup interface. On the left, the 'Connections' section lists a connection to 'localhost MySQL'. The 'Database' section is set to 'bikeshare'. The 'Table' section lists tables: facttable, location, station, tripdate, and weather. A search icon is also present.

The main area displays a data flow diagram for a query named 'facttable+ (bikeshare) (2)'. The flow starts with 'facttable', which joins with 'location' and 'tripdate'. 'location' then joins with 'station'. 'tripdate' joins with 'weather'. A 'Join' dialog box is open over the 'tripdate' join, showing four options: Inner (selected), Left, Right, and Full Outer. Below the dialog, the 'Date Key' from the 'facttable' table is joined with the 'Date Id' from the 'tripdate' table. The 'Data Source' dropdown in the dialog is set to 'tripdate'.

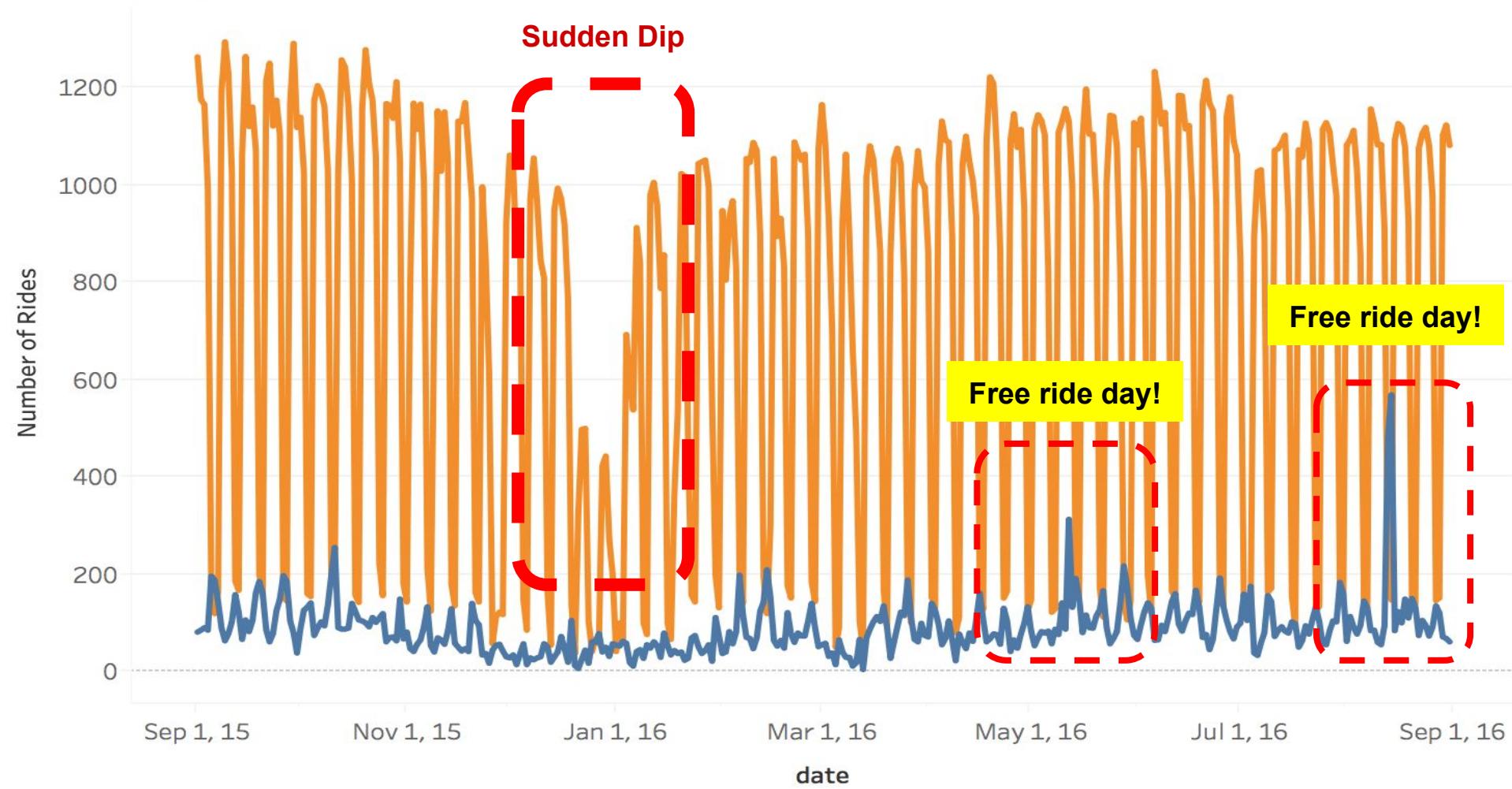
On the right, the query results are displayed as a table:

#	facttable	Date Key	=	Date Id	Table	Abc	#	location	#	Abc
	Trip Id		Add new join clause		Table	facttable	Location Id	location	Start Station	Start Station
1	913469	2	=	1	574	279	Subscriber	Subscriber	1	Steuart at Market
2	913471	3	=	2	746	436	Subscriber	Subscriber	2	Embarcadero at Market
3	913473	5	=	3	424	259	Subscriber	Subscriber	3	Market at 10th
4	913474	5	=	4	633	613	Subscriber	Subscriber	4	Embarcadero at Howard
5	913480	6	=	5	649	559	Subscriber	Subscriber	5	5th at Howard

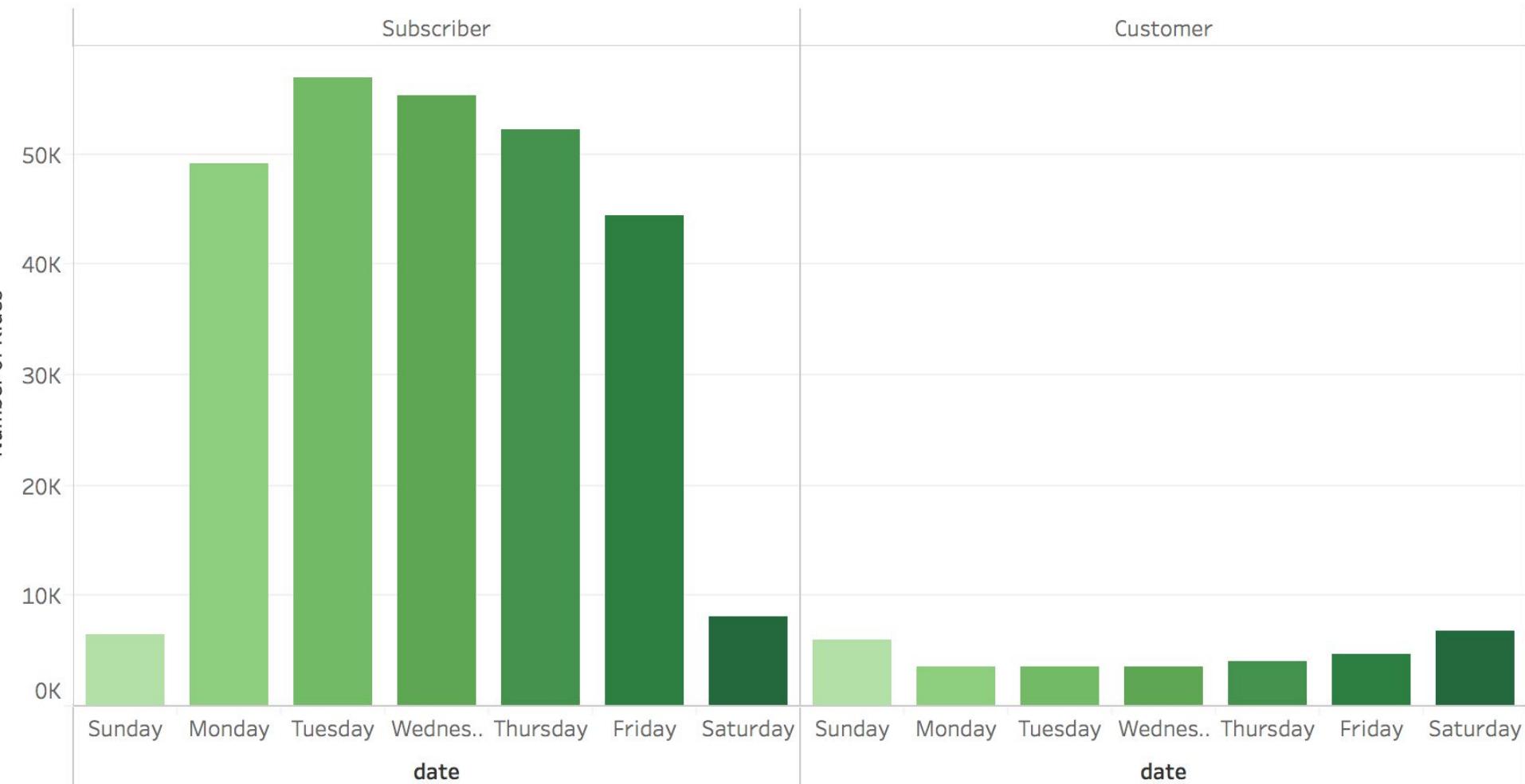
## Station location



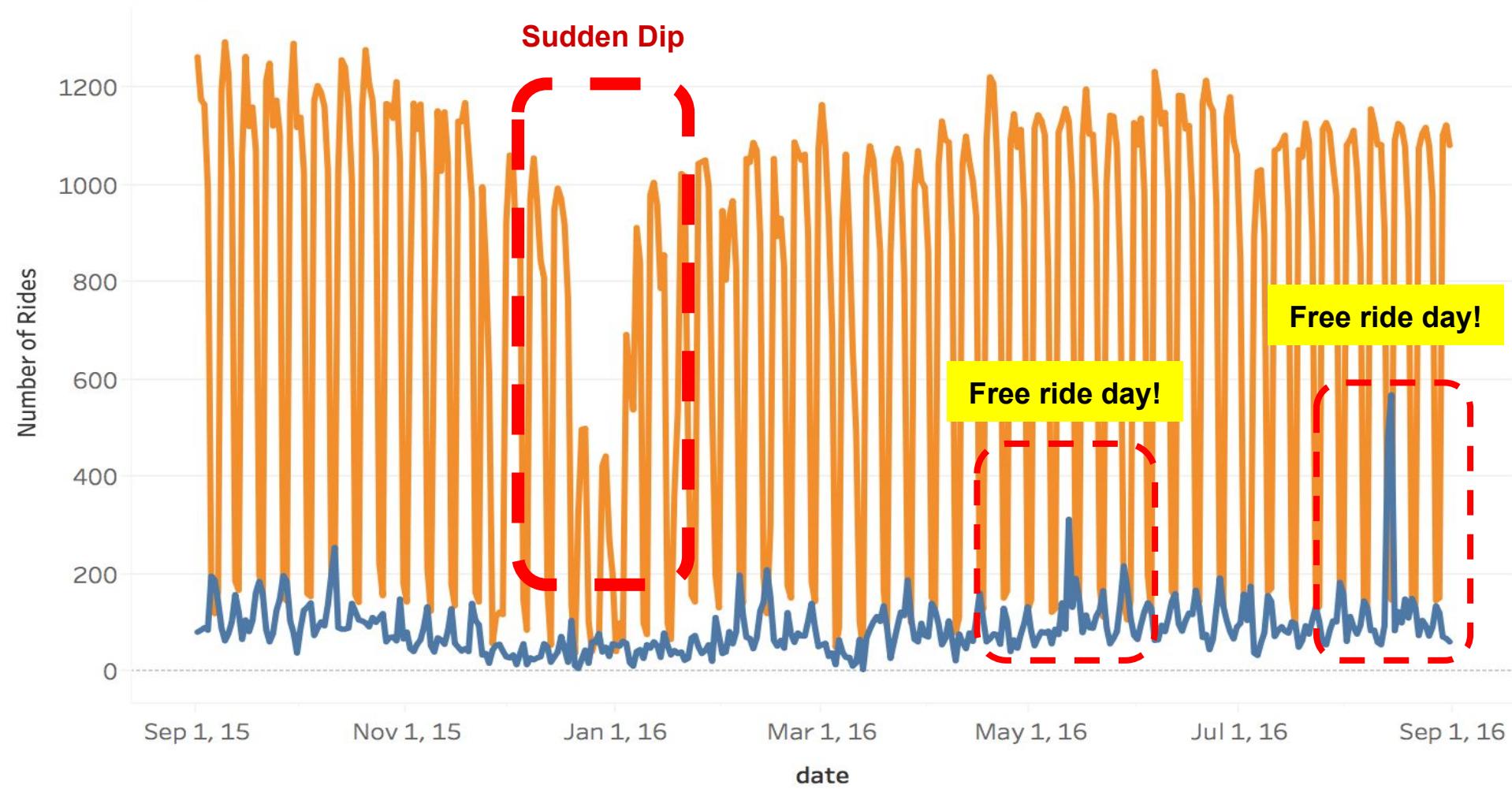
# Rides (9/1/15-9/1/16)



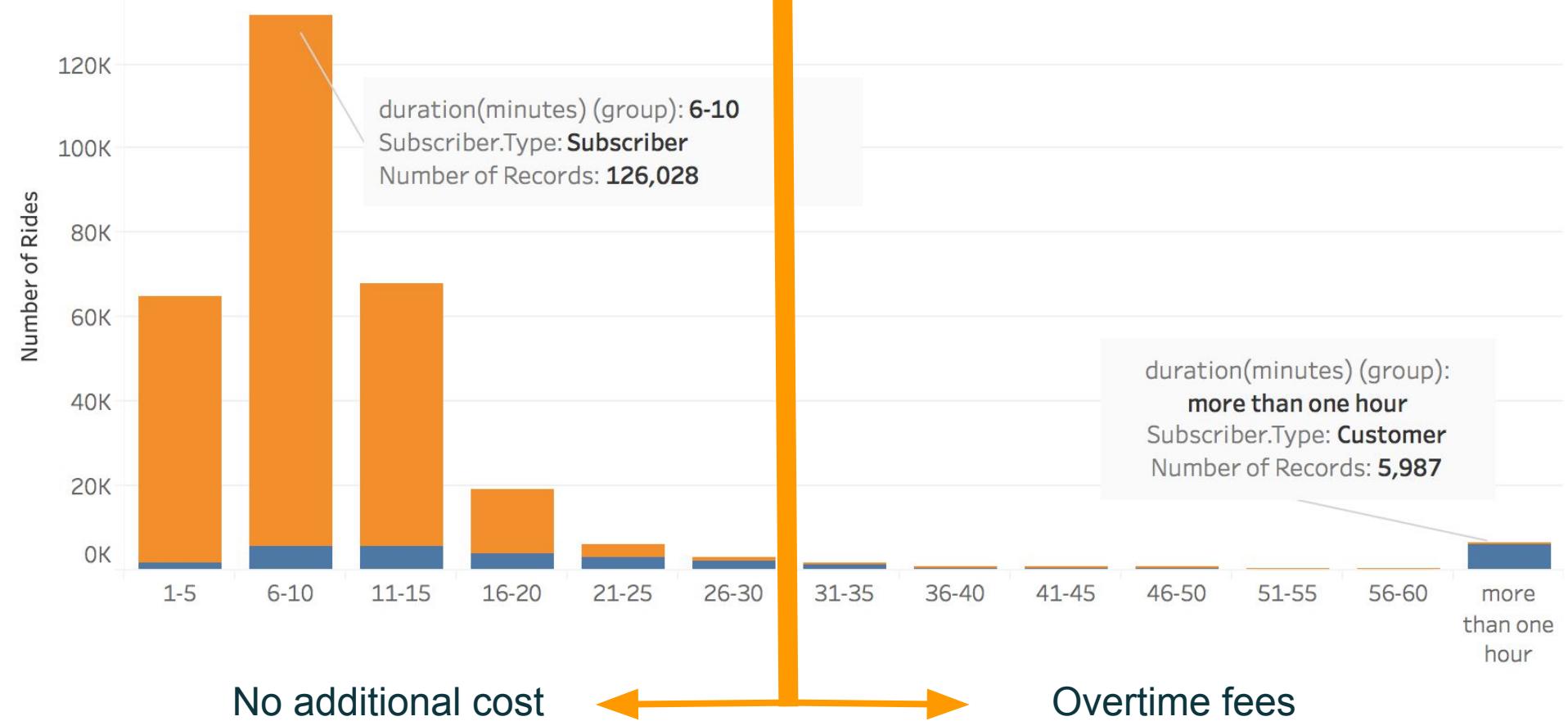
# Weekday analysis of subscribers & customers



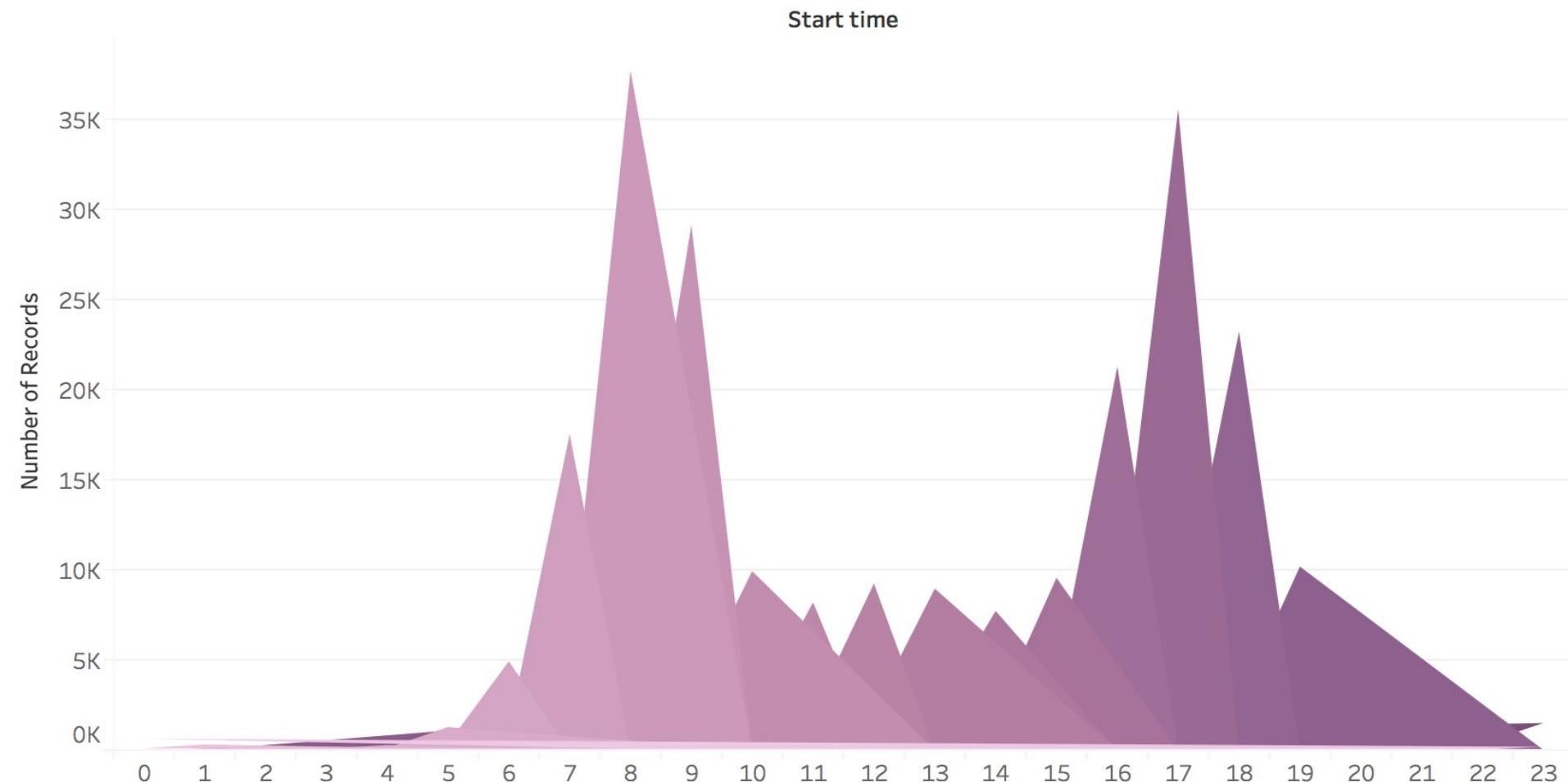
# Rides (9/1/15-9/1/16)



## Durations(miniutes)



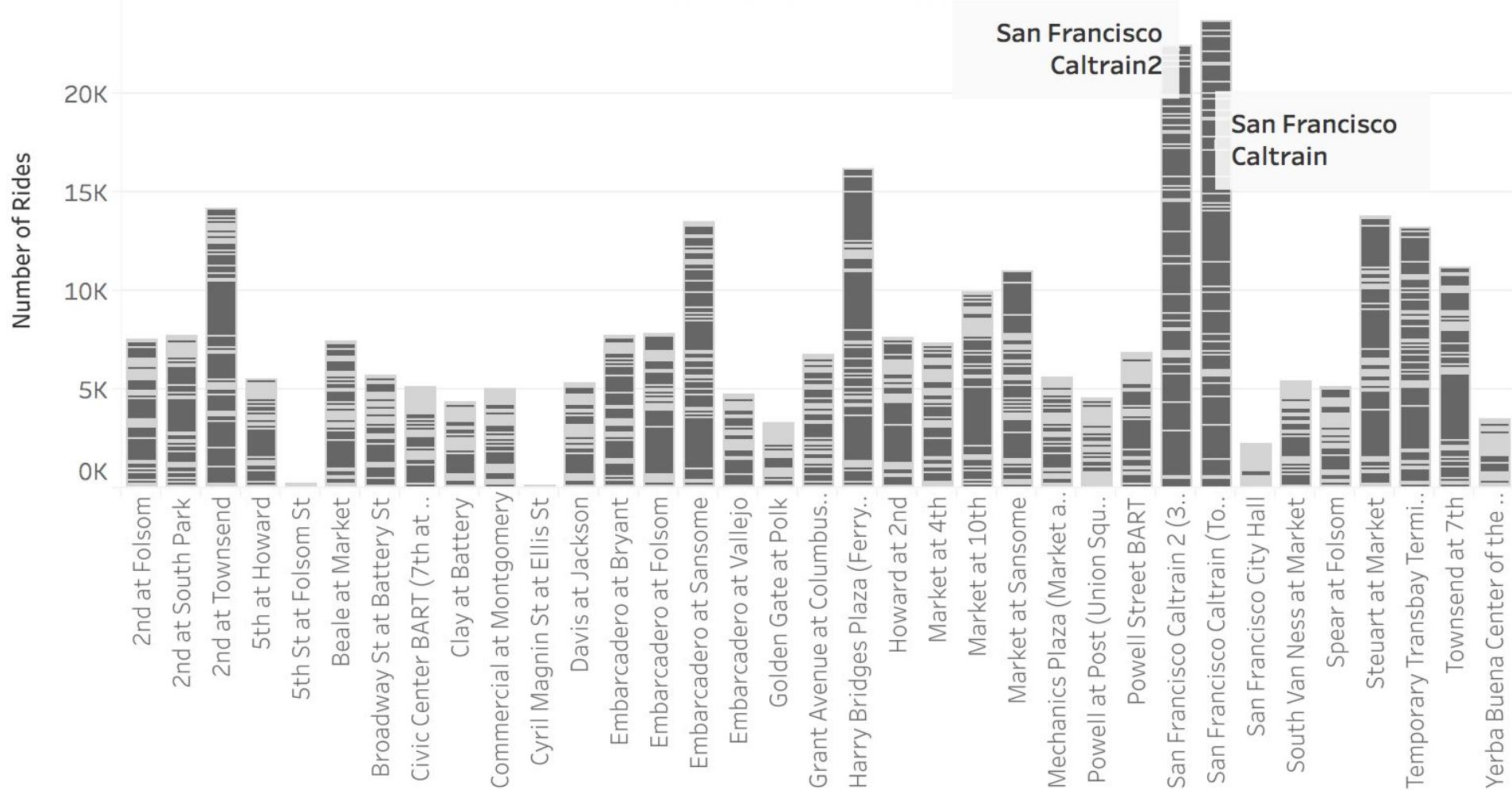
# Rides appear in rush hours



# San Francisco



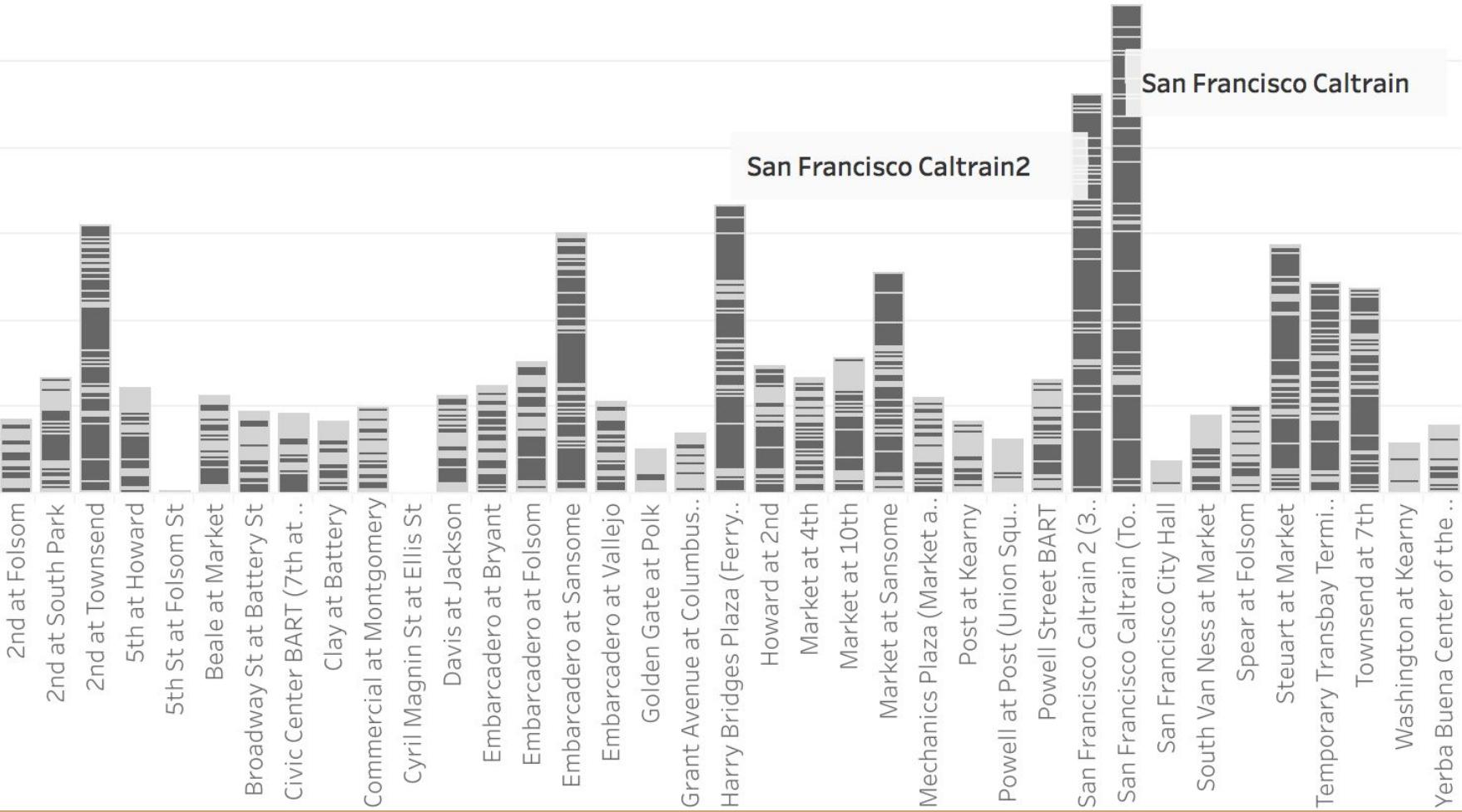
# Start.Station



# End.Station

Number of Rides

25K  
20K  
15K  
10K  
5K  
0K





# Key insights

Subscribers are mainly working population, who rides on the rush hours on weekdays for less than 20 min.

Customers are mainly tourists, who rides on the weekends for more than one hour.

Bike stations near caltrain station are the most popular stations. Many people may use bike-caltrain combinations in their routes between their houses and offices.

Thanks & Questions