A Python tool and database of Amtrak departure and arrival times with weather information

Author(s):
Lapamonpinyo, Pipatphon; Derrible, Sybil; Corman, Francesco

Publication date:
2021-10

Permanent link:
https://doi.org/https://doi.org/10.3929/ethz-b-000511371

Originally published in:
engrXiv, https://doi.org/10.31224/osf.io/rhxcj

A Python tool and database of Amtrak departure and arrival times with weather information

Pipatphon Lapamonpinyo[1]*, Sybil Derrible[1], Francesco Corman[2]

1. Department of Civil and Materials Engineering, The University of Illinois at Chicago, USA
2. Department of Civil, Environmental and Geomatic Engineering, ETH Zürich, Zurich, Switzerland
*corresponding author(s): Pipatphon Lapamonpinyo (plapam2@uic.edu)

Abstract

This article proposes a Python-based Amtrak and Weather Underground (PAWU) tool to collect data on Amtrak (the main passenger train operator in the United States) departure and arrival times with weather information. In addition, this article offers a database, developed with PAWU, of the operating characteristics of 16 Amtrak routes from 2008 to 2019. More specifically, PAWU enables users to retrieve Amtrak departure and arrival times of any train number throughout the United States. It then automatically retrieves weather information from Weather Underground for each rail station and stores the data collected in a local MySQL database. Users can easily select any desired train number(s) and date range(s) without dealing with the code and the raw data from those sources that are in different formats. The database itself can be used, in part, to develop, apply, and benchmark models that assess the performance of rail services such as the one offered by Amtrak.

Keywords: Amtrak Departure and Arrival Time Data, Weather Information, Big Data.

1. Background & Summary

Around the world, rail system services are essential to move people and goods within and between cities. In fact, rail transport is often considered as a cheaper, more efficient, and more environment-friendly way of transport. As a direct result, the use of virtually every rail system service has increased since the early 2000s. In the United States (US), for example, ridership of the main passenger train operator, Amtrak, has increased by 13.24%, from 28.7 million to 32.5 million trips, over the last decade (2008-2018) [1]. Amtrak expects its ridership to continue to increase by up to 12%, or 36.4 million trips, by 2023 [1, 2].

In parallel, in this era of Big Data, the advancement of new sensors, computer technology, and advanced automated data collection systems enables the collection of massive amounts of data from various sources that have transformed the transport sector [3]. The availability of massive amounts of data is not without problems, however. In general, Big Data is characterized by the 5 Vs—volume, velocity, variety, value and veracity—that result in databases with different characteristics, sometimes structured, semi-structured, or even unstructured [4]. From cleaning and sampling to merging, processing these databases to turn them into workable formats can be extremely difficult [5].

Train departure and arrival times as well as station information are essential inputs in many works; for example, to predict time arrivals in real time and to assess service performance of specific lines or of the entire system in general. Furthermore, weather information is also normally used as an additional data to improve model accuracy. Therefore, the objective of this work is twofold. First, it proposes a tool to collect this type of data for Amtrak and store them into a relational database via MySQL. Second, the tool was applied and a database was built that is openly accessible and that can be used for multiple purposes, including as a benchmark to test the performance of new and existing models.

Specifically, this work proposes the Python-based Amtrak and Weather Underground (PAWU) tool that integrates data on Amtrak's train departure and arrival times and weather information for each Amtrak train station retrieved from the Amtrak Status Maps Archive Database (ASMAD), with weather information from Weather Underground (WU). Technically, merging these two datasets is not trivial since the raw departure and arrival time data from ASMAD are in the form of HTML tables provided via PHP scripts, whereas the raw data from Weather Underground provided by its API service are in the form of JSON objects with nested dictionary

structure. Once collected and converted, we leverage the simplicity of the relational data structure of MySQL to store the final data.

In addition, PAWU was applied to 16 Amtrak routes over the decade 2008-2019, and the Amtrak's Departure and Arrival times with Weather information (ADAW) dataset was built and is offered as part of this work. As mentioned, this database can serve multiple purposes, including as a benchmark to compare the accuracy of models that assess the performance of the rail service from Amtrak.

The details of the proposed PAWU tool and the ADAW dataset are provided in the following sections.

2. Methods and data

2.1 A Python-based Amtrak and Weather Underground (PAWU) data collecting tool

The main contribution of this work is a Python-based Amtrak and Weather Underground (PAWU) data collecting tool. A high-level process flow diagram of PAWU is shown in Figure 1.

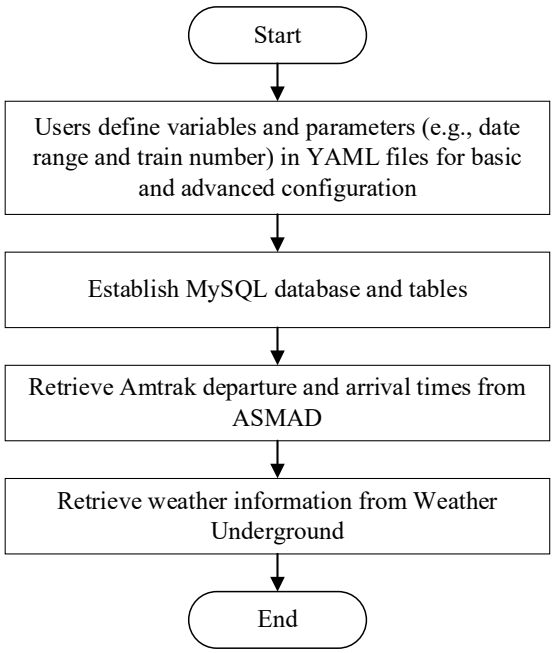

Figure 1 A high-level process flow diagram of PAWU

PAWU requires user-defined variables and parameters to configure basic and advanced features of the program needed to create a new database and tables as well as to be able to retrieve

data from ASMAD and WU. These features need to be input in files in YAML format—the YAML format was selected since it can easily be created / edited in a text editor. Specifically, one YAML file is needed to define the basic configurations and another one is needed to define the advanced configurations. For the basic configurations, users need to define a *searching option*, one or multiple *Amtrak train numbers*, one or multiple *station codes*, a *start date* and an *end date*, and a WU API key—users have two searching options: train numbers (to download all the data regarding specific train numbers) or station codes (to download all the data regarding all train numbers visiting specific stations). For the advanced configurations, users need to define the MySQL database admin information such as the *database host*, *username*, *password*, and a *database name*. Examples of basic and advanced configuration files are shown in tables 2 and 3, respectively.

After writing the two YAML files, users are also able to change the basic configuration file anytime by changing the Amtrak train number(s) or station code(s) with a specific date range before executing PAWU so as to retrieve more data from ASMAD and WU into the local database determined by the advanced configuration file. The working process in more details of PAWU will be provided in the data acquisition section.

Table 2 Basic configuration YAML file

Field Name	Description	Example
searching_option	1) searching by train number 2) searching by station code	1 or 2
train_number_list	Amtrak train number	train_number_list: - 337 - 244
station_list	This field will be automatically filled if it is searching by train number (1)	station_list: '337': - CHI - MKA - SVT - GLN - MKE '244': - ALB - HUD - RHI - POU - CRT - NYP
station_list_for_searching	Users need to provide a list of station codes for retrieving departure and arrival time from ASMAD	station_list_for_searching: - CHI - NYP

Field Name	Description	Example
		- BOS
start_date	A start date for retrieving data from ASMAD and WU	start_date: day: 2 month: 5 year: 2019
end_date	An end date for retrieving data from ASMAD and WU	end_date: day: 2 month: 5 year: 2019
wu_api_key	A user's api key for retrieving data from WU	"xxxxxxxxxxxxxxx"

Table 3 An advanced configuration YAML file

Field Name	Description	Example
db_admin		
host	MySQL host	"localhost"
user	MySQL user's name	"root"
passwd	MySQL password	"XXXX"
db_name	MySQL database name	"amtack_project"

2.2 An Amtrak's Departure and Arrival times with Weather information (ADAW) dataset

Using PAWU, the Amtrak's Departure and Arrival times with Weather information (ADAW) dataset of 16 routes was compiled. The routes were selected because they are among the most popular routes in the Amtrak system (in terms of ridership) and they vary in terms of: (a) route length and (b) performance in terms of delay in minutes per 1,000 train miles based on Amtrak's 4-year host railroad report card [6]. Table 1 contains details about the characteristics of the 16 Amtrak routes selected.

Table 1 Details about the 16 Amtrak's routes selected

No	Performance Grade	Delay (minutes per 1,000 train miles)	Distance (miles)	Railroad Host	Routes
1	A	< 600	86	Hiawatha	Chicago-Milwaukee
2	A	< 600	244	Ethan Allen Express	New York-Albany-Saratoga-Rutland VT
3	A	< 600	383	Adirondack	New York-Albany-Plattsburgh-Montreal

No	Performance Grade	Delay (minutes per 1,000 train miles)	Distance (miles)	Railroad Host	Routes
4	A	< 600	2623	Empire Builder	Chicago-Minneapolis-North Dakota-Montana-Seattle/Portland
5	B-	< 1200	170	Capitol Corridor	Sacramento, CA - San Jose, CA
6	B-	< 1200	348	Pacific Surfliner	Los Angeles, CA - San Diego, CA
7	B-	< 1200	1413	Coast Starlight	Seattle, WA - Los Angeles, CA
8	B-	< 1200	2759	Texas Eagle	Chicago, IL–Union Station - Joliet-Dallas, Temple-San Antonio - Los Angeles, CA
9	C	< 1300	194	Hoosier State	Indianapolis, IN - Chicago, IL
10	C	< 1300	725	Carolinian	New York, NY - Washington, DC - Richmond, VA - Norfolk, VA - Charlotte, NC
11	C	< 1300	1159	Cardinal	New York, NY - Charlottesville, VA - Huntington, WV - Cincinnati, OH - Indianapolis, IN - Chicago, IL
12	C	< 1300	1854	Palmetto	Washington-Richmond-Selma-Charleston-Savannah
13	F	> 1500	165	Wolverine	Chicago, IL - Pontiac, MI
14	F	> 1500	318	Blue Water	Chicago, IL - Grand Rapids, MI - Port Huron, MI
15	F	> 1500	463	Pennsylvanian	New York-Philadelphia-Harrisburg-Pittsburgh
16	F	> 1500	1161	Lake Shore Limited	New York/Boston-Albany-Cleveland-Chicago

2.3 Database structure

The PAWU database structure (used in the ADAW dataset) consists of four main tables: (1) station, (2) departure, (3) arrival, and (4) weather. Figure 2 shows an entity relation (ER) diagram of the database structure, which can be mapped into four relational tables of the same names as shown in tables 7, 8, 9, and 10.

The station table possesses station id (STN_ID) as a primary key with seven other attributes as shown in Table 7.1. Information about all Amtrak stations is extracted from geospatial metadata provided by Federal Railroad Administration (FRA) and the U.S. Department of Transportation (US DOT). Nonetheless, as the original Amtrak station metadata does not have station latitude and longitude information, the tool includes the file “amtrak_station.csv” that contains the geographical coordinates of all Amtrak stations, combining the original Amtrak station metadata

with latitude and longitude information calculated using Geographic Information Systems (GIS) software. When running PAWU, all the combined Amtrak station data with latitudes and longitudes from the csv file are automatically imported into the station table in MySQL by PAWU.

As for departure table shown in table 8, departure date (dp_date) and train number (train_number) are its composite primary keys with station id (STN_ID) as its foreign key from station table. Similarly, the arrival table has arrival date (ar_date) and train number (train_number) as its composite primary key with station id (STN_ID) referred from the station table as its foreign key as shown in Table 9. Regarding the weather table shown in Table 10, weather date (w_date) is its primary key, and station id (STN_ID) referred from the station table is its foreign key.

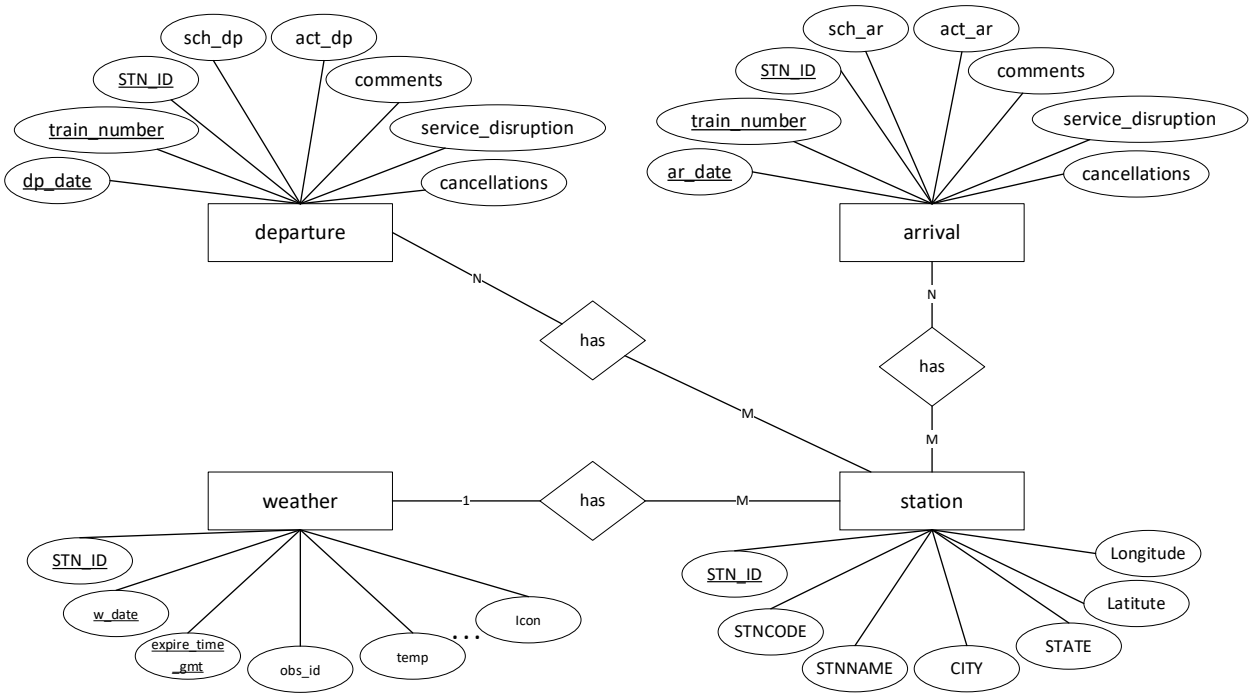

Figure 2 Entity Relationship Diagram (ERD) of the database produced with PAWU. Underlined terms indicate the primary keys used in the database

Table 4 Station

Field Name	Key	Description	Example
STN_ID	PK	Amtrak station ID	"4"
STNCODE		Amtrak station code	"CHI"
STNNAME		Amtrak station name	"Chicago (Union Station), Illinois"

CITY2		City code where a station is located	“Chicago”
STATE		State code where a station is located	“IL”
Latitude		Latitude of a rail station	-87.6394
Longitude		Longitude of a rail station	41.8787

Table 5 Departure

Field Name	Key	Description	Example
dp_date	PK	Actual departure date of a train at a station	2009-12-30
train_number	PK	Amtrak train number	337
STN_ID	FK	Amtrak station ID	4
sch_dp		Scheduled departure time	07:45:00
act_dp		Actual departure time	07:45:00
comments		Whether the train departure is on time, late, or early	Dp: on time.
service_disruption		“SD” is filled if service is disrupted	SD
Cancellations		“CT” is filled if service is cancelled	CT

Table 6 Arrival

Field Name	Key	Description	Example
ar_date	PK	Actual departure date of a train at a station	2009-12-30
train_number	PK	Amtrak train number	337
STN_ID	FK	Amtrak station ID	4
sch_ar		Scheduled arrival time	15:20:00
act_ar		Actual arrival time	14:53:00
comments		Whether the train arrival is on time, late, or early	Ar: 27 min early.
service_disruption		“SD” is filled if service is disrupted	SD
cancellations		“CT” is filled if service is cancelled	CT

Table 7 Weather

Field Name	Key	Description	Example
w_date	PK	Weather information date	2010-01-01
expire_time_gmt	PK	Expiration time in UNIX seconds	1369252800
valid_time_gmt		Valid time of observation as a Unix epoch value (seconds since start of 1970, UTC)	1369252800
STN_ID	FK	Amtrak station ID	4
obs_id		Observation weather station ID	KMDW
obs_name		Observation weather station name	Chicago/Midway
day_ind		Daytime or nighttime of the local apparent time of the location - D = Day	D

Field Name	Key	Description	Example
		- N = Night, - X = Missing	
wx_icon		The two-digit number representing the observed weather conditions	33
wx_phrase		A text description of the observed weather conditions at the reporting station	Fair
temp		The temperature of the air, at the time of the observation, measured by a thermometer 1.5 meters (4.5 feet) above the ground	55
max_temp		Highest temperature in the last 24 hours	65
min_temp		Lowest temperature in the last 24 hours	52
feels_like		An apparent temperature which represents what the air temperature “feels like” on exposed human skin due to the combined effect of the wind chill or heat index.	55
heat_index		Heat index is the apparent temperature considering both the temperature and relative humidity	55
wc		Wind chill is the apparent temperature considering both the temperature and the wind speed	55
pressure_tend		The change in the barometric pressure reading over the last hour expressed as an integer - 0 = Steady - 1 = Rising or Rapidly Rising - 2 = Falling or Rapidly Falling	1
pressure_desc		A phrase describing the change in the barometric pressure reading over the last hour	Rising
pressure		Barometric pressure exerted by the atmosphere at the earth's surface, due to the weight of the air. The units are expressed in millibars (equivalent to Hectopascals)	29.13
dewPt		The temperature in which the air needs to cool down to be totally saturated or reach 100% of relative humidity form (when a cloud would form). Once the air temperature cools below its dew point, water vapor in the atmosphere will condense. The Dew Point is also an indirect measure of the humidity of the air.	28
rh		The relative humidity of the air. It is defined as the ratio of the amount of water vapor in the air to the amount of vapor required to bring the air to saturation (100% of relative humidity) at a constant temperature. Relative humidity is always expressed as a percentage.	48
vis		The horizontal visibility at the observation point. Visibilities can be reported as fractional values particularly when visibility is less than	10

Field Name	Key	Description	Example
		2 miles. Visibilities greater than 10 statute miles (16.1 kilometers) which are considered “unlimited” are reported as “999”. Visibility values can also be equal to zero, which is not wrong as dense fogs and heavy snows can produce the values near zero.	
wdir		The direction from which the wind blows expressed in degrees. The magnetic direction varies from 1 to 360 degrees, where 360° indicates the North, 90° the East, 180° the South, 270° the West, and so forth. A ‘null’ value represents no determinable wind direction	260
wdir_cardinal		The cardinal direction from which the wind blows in an abbreviated form. Wind directions are always expressed as “from whence the wind blows” meaning that a North wind blows from North to South. If you face North in a North wind, the wind is at your face. Face southward and the North wind is at your back	W
gust		Sudden or brief variations in the wind speed	38
wspd		Wind Speed	29
precip_total		Precipitation amount in the last rolling 24-hour period	0.3
precip_hrly		Precipitation for the last hour	0.5
snow_hrly		Snow increasing rapidly in inches or centimeters per hour depending on whether the snowfall is reported by METAR or TECCI (synthetic observations). METAR snow accumulation is in inches and TECCI is in centimeters	1
uv_desc		Ultraviolet index description (e.g., Extreme, High, Low, Minimal, Moderate, No Report, Not Available)	Low
uv_index		Ultraviolet index (0 - 11, and 999)	0
qualifier		Weather description qualifier code	QQ0063
qualifier_svrty		Weather description qualifier severity (1 (low) to 6 (high)	1
clds		Cloud cover description code (SKC, CLR, SCT, FEW, BKN, OVC)	CLR
water_temp		Water temperature	80
primary_wave_period		Primary wave period (0 - 99)	13
primary_wave_height		Primary wave height (0 - 99.99)	3.28
primary_swell_period		Primary swell period (0 - 99)	13
primary_swell_height		Primary swell height (0 – 99.99)	1.64

Field Name	Key	Description	Example
primary_swell_direction		Primary swell direction (0 - 359)	190
secondary_swell_period		Secondary swell period (0 - 99)	13
secondary_swell_height		Secondary swell height (0 – 99.99)	1.64
secondary_swell_direction		Secondary swell direction (0 - 359)	190

2.4 Data acquisition

The overall working process of PAWU is shown in figure 3. As PAWU stores data into a relational database, users need to install MySQL before using the tool. After that, PAWU automatically creates four tables—station, departure, arrival, and weather—the first time it is run or if tables with the same names do not exist.

For this process, users need to have, in the same folder, the two previously discussed YAML files: *basic_configuration.yaml* and *advanced_configuration.yaml.* The *advanced_configuration.yaml* file is first read (see Table 3 for an example). Next, all Amtrak station information are imported to station table on MySQL. Then, PAWU retrieves historical departure and arrival time data from ASMAD and stores them in the departure and arrival tables via MySQL. For this step, users need to define whether they want to search the data by train number ("1") or by station code ("2") in the *basic_configuration.yaml* file.

In addition, users need to provide the start and end dates in the *basic_configuration.yaml* file as well. Moreover, users also need to provide the API key for retrieving historical weather data in the *basic_configuration.yaml* file. Table 5 shows a sample *basic_configuration.yaml* file.

Following these steps, PAWU automatically retrieves weather information for all train stations included in the station table and stores the data into the weather table in MySQL.

Code availability

Both PAWU and the amtrak_stations.csv file are available for download at https://github.com/csunlab/PAWU and https://csun.uic.edu/codes/PAWU.html. PAWU is compatible with Python version 3.5 or higher. As for MySQL on Windows, it is also available for download at https://dev.mysql.com/downloads/installer/. It is recommended to download and install via MySQL installer community together with MySQL Workbench. As for installing MySQL on OS X, instructions and guidelines are provided at

https://dev.mysql.com/doc/refman/5.6/en/osx-installation.html. For Linux, instructions and guidelines are available at https://dev.mysql.com/doc/refman/8.0/en/linux-installation.html.

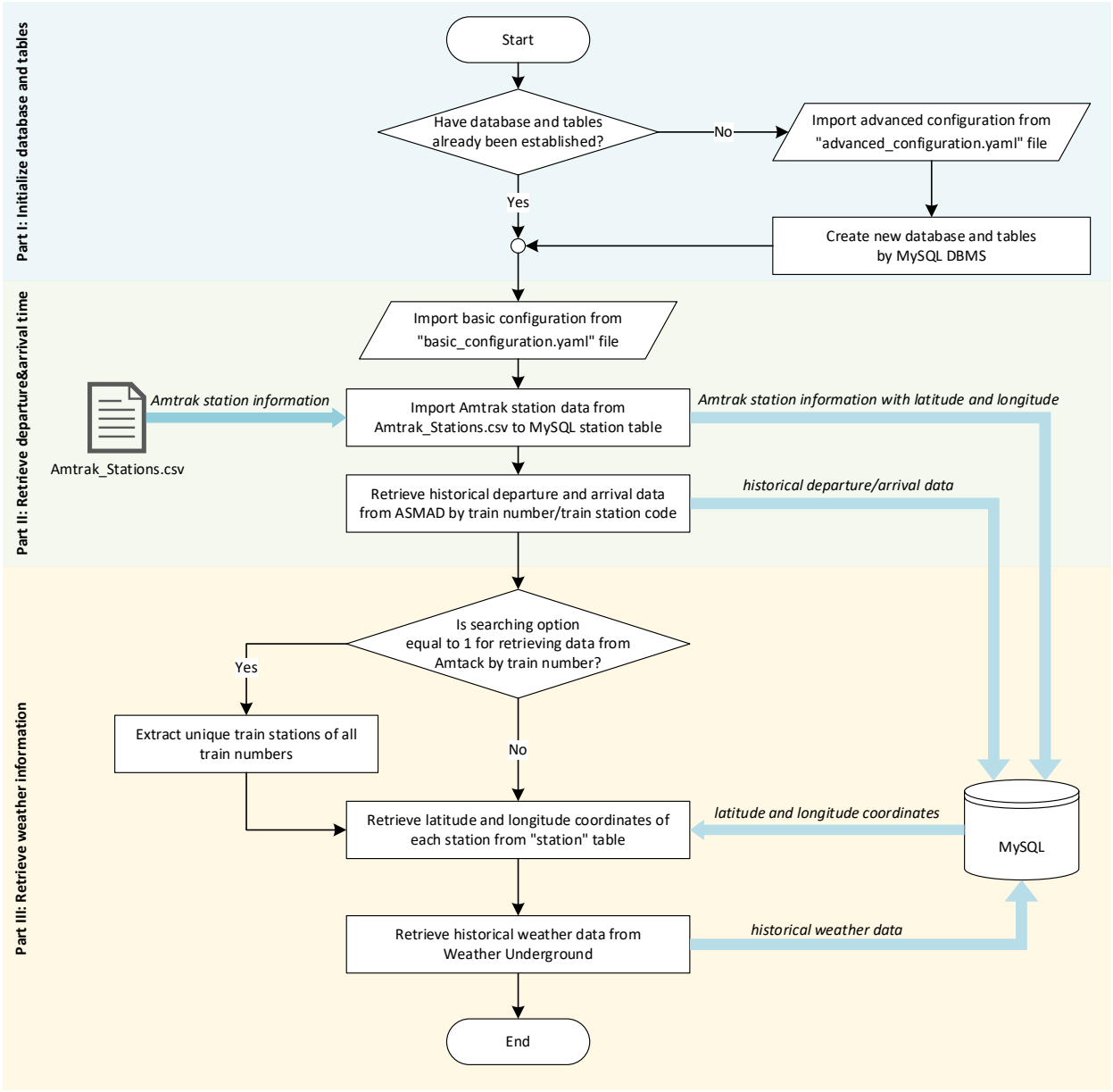

Figure 3 A workflow diagram of PAWU

References

1. Amtrak: Amtrak Five Year Service Line Plans (FY 2019–2023). (2018)
2. Amtrak: Amtrak Annual Report 2015. (2016)
3. Cottrill, C.D., Derrible, S.: Leveraging Big Data for the Development of Transport Sustainability Indicators. Journal of Urban Technology. 22, 45–64 (2015). https://doi.org/10.1080/10630732.2014.942094
4. Thaduri, A., Galar, D., Kumar, U.: Railway Assets: A Potential Domain for Big Data Analytics. Procedia Computer Science. 53, 457–467 (2015). https://doi.org/10.1016/j.procs.2015.07.323
5. Philip Chen, C.L., Zhang, C.-Y.: Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. Information Sciences. 275, 314–347 (2014). https://doi.org/10.1016/j.ins.2014.01.015
6. Zappi, C.M.: Amtrak Host Railroad Report Card 2018. (2019)
7. IBM: Weather Company Data - PWS - Historical - v2, https://docs.google.com/document/d/1w8jbqfAk0tfZS5P7hYnar1JiitM0gQZB-clxDfG3aD0/edit#, (2018)