

**IO-AVSTATS-DB - User Manual**

**Release 23.04.15**

**IO Aeronautical Autonomy Labs, LLC**



**© IO-Aero 2023**

Table of Contents

[Introduction 3](#_Toc131093369)

[Getting Started 4](#_Toc131093370)

[Requirements 4](#_Toc131093371)

[First Installation 4](#_Toc131093372)

[Update Installation 4](#_Toc131093373)

[Access credentials for **IO-AVSTATS-DB** 4](#_Toc131093374)

[Database Schema 5](#_Toc131093375)

[NTSB 5](#_Toc131093376)

[IO-Aero specific supplements 6](#_Toc131093377)

[Column **io\_last\_seen\_ntsb** 6](#_Toc131093378)

[Indexes 7](#_Toc131093379)

[Materialized View **io\_app\_ae1982** 8](#_Toc131093380)

[Table **events** 17](#_Toc131093381)

[Table **io\_aviation\_occurrence\_categories** 18](#_Toc131093382)

[Table **io\_countries** 20](#_Toc131093383)

[Table **io\_lat\_lng** 22](#_Toc131093384)

[Table **io\_md\_codes\_category** 26](#_Toc131093385)

[Table **io\_md\_codes\_eventsoe** 27](#_Toc131093386)

[Table **io\_md\_codes\_modifier** 28](#_Toc131093387)

[Table **io\_md\_codes\_phase** 29](#_Toc131093388)

[Table **io\_md\_codes\_section** 30](#_Toc131093389)

[Table **io\_md\_codes\_subcategory** 31](#_Toc131093390)

[Table **io\_md\_codes\_subsection** 32](#_Toc131093391)

[Table **io\_ntsb\_2002\_2021** 33](#_Toc131093392)

[Table **io\_processed\_files** 35](#_Toc131093393)

[Table **io\_sequence\_of\_events** 36](#_Toc131093394)

[Table **io\_states** 37](#_Toc131093395)

[Data Cleansing 41](#_Toc131093396)

[Decimal latitude and longitude 41](#_Toc131093397)

[Trimmed Database Columns 42](#_Toc131093398)

[Change Log 43](#_Toc131093399)

[References 44](#_Toc131093400)

# Introduction

The National Transportation Safety Board (NTSB) investigates all aviation accidents in the U.S. and makes the investigation results available on their website (data.ntsb.gov, n.d.) in Microsoft Access database files for public use. The NTSB provides aviation event data from 1982 through 2007 in the file **Pre2008.zip** (version 9/30/2020 12:51:56 PM). Data since 2008 are available first in the overall **avall.zip** file, which is updated monthly, and second with a quasi-weekly amendment file each month on the 1st, 8th, 15th, and 22nd, e.g., **up22JUN.zip**.

**IO-AVSTATS-DB** is IO-Aero’s database version of NTSB’s aviation event data. **IO-AVSTATS-DB** is based on a PostgreSQL database (The PostgreSQL Global Development Group, 2019) and contains, in addition to the data from NTSB IO-Aero specific extensions and data from the following sources:

* Aviation Occurrence Categories (AVIATION OCCURRENCE CATEGORIES DEFINITIONS AND USAGE NOTES, 2013)
* GeoDatos (Geodatos.net, 2019)
* opendatasoft (Opendatasoft, n.d.)
* simplemaps (simplemaps.com, n.d.)
* United States Zip Codes.org (UnitedStatesZipCodes, n.d.)

The **IO-AVSTATS-DB** is updated on the 8th, 15th, and 22nd of each month with the update file provided by NTSB. Every first of the month a new database version is created from the two files **Pre2008.zip** and **avall.zip**. This ensures that the deletions made by NTSB, which are not included in the update file, are reflected in the **IO-AVSTATS-DB** database.

# Getting Started

IO-Aero provides the database files in PostgreSQL format. The development of **IO-AVSTATS-DB** was done using PostgreSQL version 15 in the form of the official Docker images available on DockerHub (hub.docker.com. (n.d.)). This is also the recommended installation form for using **IO-AVSTATS-DB**.

## Requirements

Docker Desktop (Docker Documentation, 2022)

## First Installation

* If not already installed, install the latest version of Docker Desktop.
* From IO-Aero's member area (members.io-aero.com, n.d.) download the zipped **IO-AVSTATS-DB** database.
* Unzip the downloaded file - the default name of the new directory is **postgres**.
* Create and launch a PostgreSQL Docker container - for example, as follows:

docker create -e POSTGRES\_DB=io\_avstats\_db

-e POSTGRES\_HOST\_AUTH\_METHOD=password

-e POSTGRES\_PASSWORD=YL9CGphL58Rye3nUDopw

-e POSTGRES\_USER=guest

--name io\_avstats\_db

-p 5432:5432

-v postgres:/var/lib/postgresql/data postgres:latest

docker start io\_avstats\_db

## Update Installation

* Stop the Docker container that may be running.
* Delete the existing database directory **postgres**.
* From IO-Aero's member area (members.io-aero.com, n.d.) download the zipped **IO-AVSTATS-DB** database.
* Unzip the downloaded file.
* Restart the Docker container.

## Access credentials for **IO-AVSTATS-DB**

Username: **guest**

Password: **YL9CGphL58Rye3nUDopw**

# Database Schema

## NTSB

The documentation provided by NTSB on the data provided consists of the following two documents (data.ntsb.gov. (n.d.)):

* codman.pdf –-aviation coding manual
* eadmspub.pdf – database schema diagram

For the data provided by NTSB in the form of MS Access databases, the data dictionary was extracted using RazorSQL (razorsql.com, n.d.) and transferred to the PostgreSQL database **IO-AVSTATS-DB**. This includes the following tables:

* aircraft
* dt\_aircraft
* dt\_events
* dt\_flight\_crew
* engines
* events
* Events\_Sequence
* Findings
* Flight\_Crew
* flight\_time
* injury
* narratives
* NTSB\_Admin
* Occurrences
* seq\_of\_events

During the transfer to PostgreSQL, the spelling of all table and column names was unified to lowercase letters. Furthermore, the following adjustments were made to the data types:

|  |  |
| --- | --- |
| MS Access | PostgreSQL |
| bit | boolean |
| byte | char(1) |
| datetime | timestamp |
| double | float |
| integer | int |
| longchar | text |

## IO-Aero specific supplements

### Column **io\_last\_seen\_ntsb**

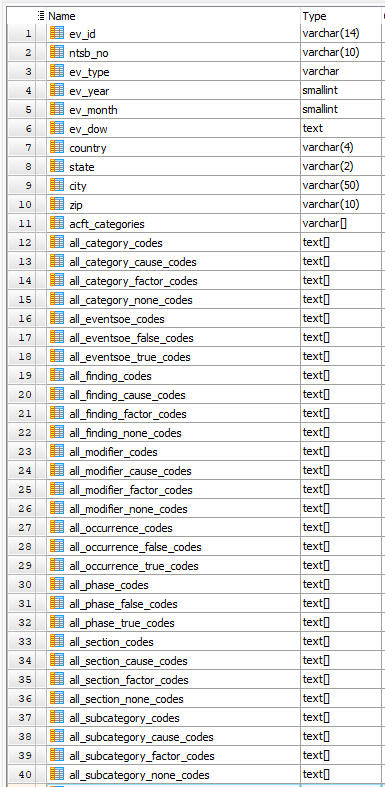
This database column of the timestamp data type has been added to all NTSB tables to document when a specific row was created or last modified.

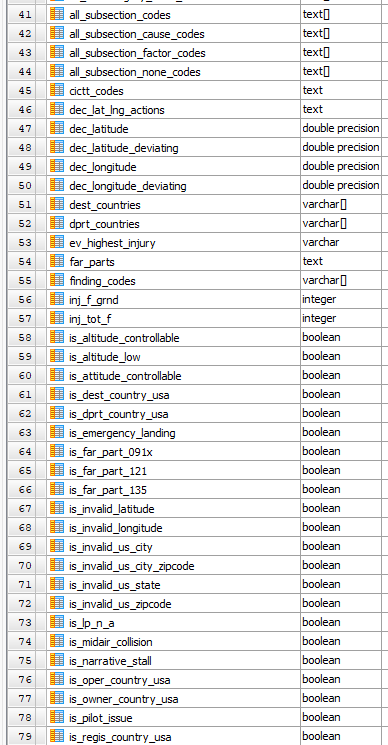
### Indexes

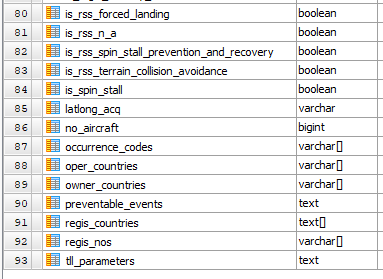
|  |  |
| --- | --- |
| table / view | column(s) |
| aircraft | acft\_category |
| events\_sequence | occurrence\_code |
| findings | finding\_code |
| io\_app\_ae1982 | cictt\_codes |
| io\_app\_ae1982 | ev\_highest\_injury |
| io\_app\_ae1982 | ev\_type |
| io\_app\_ae1982 | far\_parts |
| io\_app\_ae1982 | inj\_f\_grnd |
| io\_app\_ae1982 | inj\_tot\_f |
| io\_app\_ae1982 | latlong\_acq |
| io\_app\_ae1982 | no\_aircraft |
| io\_app\_ae1982 | preventable\_events |
| io\_app\_ae1982 | tll\_parameters |

### Materialized View **io\_app\_ae1982**

This materialized view provides all necessary data for **IO-Aero**'s data analysis applications.







#### Original Data - Unconditional

* **events.ev\_highest\_injury**
* **events.ev\_id**
* **events.ev\_month**
* **events.ev\_type**
* **events.ev\_year**
* **events.inj\_f\_grnd**
* **events.inj\_tot\_f**
* **events.lat\_lng\_acq**
* **events.ntsb\_no**

#### Original Data - with Conditions

* **events.ev\_dow** - capital letters
* **events.io\_city** - if empty **events.ev\_city**
* **events.io\_country** - if empty **events.ev\_country**
* **events.io\_dec\_latitude** - if empty **events.ev\_dec\_latitude**
* **events.io\_dec\_longitude** - if empty **events.ev\_dec\_longitude**
* **events.io\_site\_zipcode** - if empty **events.ev\_site\_zipcodey**
* **events.io\_state** - if empty **events.ev\_state**

#### Logical Variables

##### Level Base

* **ev\_type**: Categorization of missing data as '**n/a**'.
* **acft\_categories** - an array of **aircraft.acft\_category**
* **all\_category\_codes** - an array of **findings.category\_no**
* **all\_category\_cause\_codes** - an array of **findings.category\_no** where **cause\_factor** is ‘**C**’
* **all\_category\_factor\_codes** - an array of **findings.category\_no** where **cause\_factor** is ‘**F**’
* **all\_category\_none\_codes** - an array of **findings.category\_no** where **cause\_factor** is neither ‘**C**’ nor ‘**F**’
* **all\_eventsoe\_codes** - an array of **events\_sequence.eventsoe\_no**
* **all\_eventsoe\_false\_codes** - an array of **events\_sequence.eventsoe\_no** where **defining\_ev** is not true
* **all\_eventsoe\_true\_codes** - an array of **events\_sequence.eventsoe\_no** where **defining\_ev** is true
* **all\_finding\_codes** - an array of **findings.finding\_no**
* **all\_finding\_cause\_codes** - an array of **findings.finding\_no** where **cause\_factor** is ‘**C**’
* **all\_finding\_factor\_codes** - an array of **findings.finding\_no** where **cause\_factor** is ‘**F**’
* **all\_finding\_none\_codes** - an array of **findings.finding\_no** where **cause\_factor** is neither ‘**C**’ nor ‘**F**’
* **all\_modifier\_codes** - an array of **modifiers.modifier\_no**
* **all\_modifier\_cause\_codes** - an array of **modifiers.modifier\_no** where **cause\_factor** is ‘**C**’
* **all\_modifier\_factor\_codes** - an array of **modifiers.modifier\_no** where **cause\_factor** is ‘**F**’
* **all\_modifier\_none\_codes** - an array of **modifiers.modifier\_no** where **cause\_factor** is neither ‘**C**’ nor ‘**F**’
* **all\_occurrence\_codes** - an array of **events\_sequence.occurrence\_no**
* **all\_occurrence\_false\_codes** - an array of **events\_sequence.occurrence\_no** where **defining\_ev** is not true
* **all\_occurrence\_true\_codes** - an array of **events\_sequence.occurrence\_no** where **defining\_ev** is true
* **all\_phase\_codes** - an array of **events\_sequence.phase\_no**
* **all\_phase\_false\_codes** - an array of **events\_sequence.phase\_no** where **defining\_ev** is not true
* **all\_phase\_true\_codes** - an array of **events\_sequence.phase\_no** where **defining\_ev** is true
* **all\_section\_codes** - an array of **sections.section\_no**
* **all\_section\_cause\_codes** - an array of **sections.section\_no** where **cause\_factor** is ‘**C**’
* **all\_section\_factor\_codes** - an array of **sections.section\_no** where **cause\_factor** is ‘**F**’
* **all\_section\_none\_codes** - an array of **sections.section\_no** where **cause\_factor** is neither ‘**C**’ nor ‘**F**’
* **all\_subcategory\_codes** - an array of **subcategorys.subcategory\_no**
* **all\_subcategory\_cause\_codes** - an array of **subcategorys.subcategory\_no** where **cause\_factor** is ‘**C**’
* **all\_subcategory\_factor\_codes** - an array of **subcategorys.subcategory\_no** where **cause\_factor** is ‘**F**’
* **all\_subcategory\_none\_codes** - an array of **subcategorys.subcategory\_no** where **cause\_factor** is neither ‘**C**’ nor ‘**F**’
* **all\_subsection\_codes** - an array of **subsections.subsection\_no**
* **all\_subsection\_cause\_codes** - an array of **subsections.subsection\_no** where **cause\_factor** is ‘**C**’
* **all\_subsection\_factor\_codes** - an array of **subsections.subsection\_no** where **cause\_factor** is ‘**F**’
* **all\_subsection\_none\_codes** - an array of **subsections.subsection\_no** where **cause\_factor** is neither ‘**C**’ nor ‘**F**’
* **cictt\_codes** - an array of **events\_sequence.cictt\_code**
* **description\_main\_phase\_defining** - an array of **dewcription\_main\_phase** where **defining\_ev** is true
* **dest\_countries** - an array of **aircraft.dest\_country**
* **dprt\_countries** - an array of **aircraft.dprt\_country**
* **ev\_highest\_injury**: Categorization of missing data as '**n/a**'.
* **far\_parts** - an array of **aircraft.far\_part**
* **finding\_codes** - an array of encoded **findings.finding\_code**:
  + ‘**PARAMS\_ALT**’ if position 1 to 8 of **finding\_code** is ‘**01062012**’
  + ‘**PARAMS\_DEC\_RATE**’ if position 1 to 8 of **finding\_code** is ‘**01062037**’
  + ‘**PARAMS\_DEC\_APP**’ if position 1 to 8 of **finding\_code** is ‘**01062040**’
  + ‘**PARAMS\_AoA**’ if position 1 to 8 of **finding\_code** is ‘**01062042**’
  + ‘**ENV\_TER** if position 1 to 6 of **finding\_code** is ‘**030210**’
  + ‘**ENV\_OAS**’ if position 1 to 6 of **finding\_code** is ‘**030220**’
* **is\_attitide\_controllable** –if:
  + position 1 to 6 of **finding\_code** is one of
    - ‘**010355**’, ‘**010357**’, ‘**030320**’, ‘**030330**’
  + or position 1 to 8 of **finding\_code** is one of
    - ‘**01061040**’, ‘**03031025**’, ‘**03034020**’, ‘**03034030**’, ‘**03034050**’, ‘**03034060**’
  + or **eventsoe\_no** is one of
    - ‘**210**’, ‘**245**’, ‘**333**’, ‘**361**’,
  + or **eventsoe\_no** is one of ‘**337**’, ‘**338**’ and **defining\_ev**
* **is\_midair\_collision** –if:
  + **eventsoe\_no** is ‘**250**’ and **defining\_ev**
* **is\_narrative\_stall** –if:
  + **narratives.narr\_accp** is like ‘**%STALL%**
* **is\_pilot\_issue** –if:
  + **modifier\_no** is one of ‘**44**’, ‘**45**’, ‘**46**’
* **latlong\_acq** is ‘**none**’ if
  + **latlong\_acq** is null
  + and **io\_latlong\_acq** is null
  + and one of **dec\_latitude**, **dec\_longitude**, **io\_dec\_latitude**, **io\_dec\_longitude** is not null
* **no\_aircraft** –the number of aircraft involved in the event
* **occurrence\_codes** - an array of encoded **events\_sequence.occurrence\_code**:
  + ‘**INIT\_CLIMB**’ if **phase:no** is ‘**350**’
  + ‘**MAN\_LALT**’ if **phase:no** is ‘**452**’
  + ‘**FINAL\_APP**’ if **phase:no** is ‘**502**’
  + ‘**CFIT**’ if **eventsooe\_no** is ‘**120**’
  + ‘**LALT**’ if **eventsooe\_no** is ‘**220**’
  + ‘**LOC-I**’ if **eventsooe\_no** is ‘**240**’
  + ‘**STALL**’ if **eventsooe\_no** is ‘**241**’
  + ‘**MIDAIR**’ if **eventsooe\_no** is ‘**250**’
  + ‘**UIMC**’ if **eventsooe\_no** is ‘**401**’
  + ‘**CAA**’ if **eventsooe\_no** is ‘**420**’
  + ‘**BIRD**’ if **eventsooe\_no** is ‘**901**’
* **oper\_countries** - an array of **aircraft.oper\_country**
* **owner\_countries** - an array of **aircraft.owner\_country**
* **phase\_codes\_defining** - an array of **events\_sequence.phase\_no** where **defining\_ev** is true
* **regis\_countries** - an array of
  + ‘**USA**’ if **regis\_no** has one of the formats ‘**N99999**’, ‘**N9999A**’, ‘**N999AA**’,
  + ‘**NON-US**’ else
* **Regis\_nos** - an array of **aircraft.regis\_no**

##### Level 1

* **is\_dest\_country\_usa** –if ‘**USA**’ is in **dest\_countries**
* **is\_dprt\_country\_usa** –if ‘**USA**’ is in **dprt\_countries**
* **is\_emergency\_landing** –if:
  + **modifier\_no** is one of
    - ‘**01**’, ‘**02**’, ‘**03**’, ‘**06**’, ‘**22**’, ‘**23**’, ‘**24**’, ‘**25**’, ‘**26**’
  + or position 1 to 6 of **finding\_code** is one of
    - ‘**010224**’, ‘**010228**’, ‘**010230**’, ‘**010461**’, ‘**010462**’, ‘**010465**’, ‘**010467**’, ‘**010571**’, ‘**010572**’, ‘**010573**’, ‘**010574**’, ‘**010578**’, ‘**010579**’, ‘**010581**’, ‘**010585**’
  + or position 1 to 8 of **finding\_code** is one of
    - ‘**01050000**’, ‘**01061010**’, ‘**01061020**’, ‘**01062025**’, ‘**01071000**’, ‘**01071010**’
  + or **phase\_no** is ‘**600**’
  + or **eventsoe\_no** is one of
    - ‘**130**’, ‘**140**’, ‘**190**’, ‘**191**’, ‘**192**’, ‘**193**’, ‘**194**’, ‘**340**’, ‘**341**’, ‘**342**’, ‘**343**’, ‘**440**’, ‘**441**’, ‘**500**’, ‘**901**’,
  + or **eventsoe\_no** is one of ‘**337**’, ‘**338**’ and **defining\_ev**
* **is\_far\_part\_091x** –if **far\_parts** is any of
  + ‘**091**’
  + ‘**091F**’
  + ‘**091K**’
* **is\_far\_part\_121** –if ‘**121**’ is in **far\_parts**
* **is\_far\_part\_135** –if ‘**135**’ is in **far\_parts**
* **is\_oper\_country\_usa** –if ‘**USA**’ is in **oper\_countries**
* **is\_owner\_country\_usa** –if ‘**USA**’ is in **owner\_countries**
* **is\_regis\_country\_usa** –if ‘**USA**’ is in **regis\_countries**
* **is\_spin\_stall** –if
  + **(** ‘**PARAMS\_AoA**’ is in **findings\_codes**
  + or ‘**STALL**’ is in **occurrence\_codes**
  + or ‘**LOC-I**’ is in **occurrence\_codes** and **is\_narrative\_stall )**
  + and not **(** **occurrence\_codes** is any of ‘**CAA**’, ‘**CFIT**’ **)**

##### Level 2

* **is\_altitude\_controllable** –if one of the following logical variables is true:
  + **is\_attitude\_controllable**
  + **is\_emergency\_landing**
  + **is\_spin\_stall**
* **is\_altitude\_low** –if:
  + **( occurrence\_codes** is one of
    - ‘**CAA**’,
    - ‘**CFIT**’,
    - ‘**FINAL\_APP**’,
    - ‘**INIT\_CLIMB**’,
    - ‘**LALT**’,
    - ‘**MAN\_LALT**’
  + or **findings\_codes** is ‘**ENV\_OAS**’ and **occurrence\_codes** is not ‘**BIRD**’
  + or **findings\_codes** is one of
    - ‘**ENV\_TER**’
    - ‘**PARAMS\_ALT**’
    - ‘**PARAMS\_DEC\_APP**’
    - ‘**PARAMS\_DEC\_RATE**’ **)**
  + and not **( occurrence\_codes** is ‘**MIDAIR**’ and **is\_spin\_stall** is false **)**

##### Level 3

* **is\_rss\_forced\_landing** –if one of the following logical variables is true:
  + **is\_attitude\_controllable**
  + **is\_emergency\_landing**
* **is\_rss\_spin\_stall\_prevention\_and\_recovery** –if one of the following logical variables is true:
  + **is\_attitude\_controllable**
  + **is\_spin\_stall**
* **is\_rss\_terrain\_collision\_avoidance** –if one of the following logical variables is true:
  + **is\_altitude\_controllable**
  + **is\_altitude\_low**
  + **is\_attitude\_controllable**

##### Level 4

* **cictt\_codes**: Categorization of missing data as '**no data**'.
* **far\_parts**: Categorization of missing data as '**no data**'.
* **is\_lp\_n\_a** – a logical parameter is not available if none of the following logical variables is true:
  + **is\_altitude\_low**
  + **is\_attitude\_controllable**
  + **is\_emergency\_landing**
  + **is\_midair\_collision**
  + **is\_pilot\_issue**
  + **is\_rss\_forced\_landing**
  + **is\_rss\_spin\_stall\_prevention\_and\_recovery**
  + **is\_rss\_terrain\_collision\_avoidance**
  + **is\_spin\_stall**
* **is\_rss\_n\_a** – a required safety system is not available if none of the following logical variables is true:
  + **is\_midair\_collision**
  + **is\_rss\_forced\_landing**
  + **is\_rss\_spin\_stall\_prevention\_and\_recovery**
  + **is\_rss\_terrain\_collision\_avoidance**
* **preventable\_events**:
  + **Airborne collision** if **is\_midair\_collision**
  + **Forced landing** if **is\_rss\_forced\_landing**
  + **Spin / stall** if **is\_rss\_spin\_stall\_prevention\_and\_recovery**
  + **Terrain collision** if **is\_rss\_terrain\_collision\_avoidance**
  + **n/a** if none of the above
* **tll\_parameters** – top logical parameters:
  + **Aerodynamic spin / stall** if **is\_spin\_stall**
  + **Aircraft can climb** if **is\_altitude\_controllable**
  + **Aircraft has degraded control failure** if **is\_emergency\_landing**
  + **Altitude too low** if **is\_altitude\_low**
  + **Attitude is controllable** if **is\_attitude\_controllable**
  + **Pilot is able to perform maneuver** if **is\_pilot\_issue**
  + **n/a** if none of the above

Sample data:

**Name |Value |**

-----------------------------+-------------------+

ev\_id |20221112106276 |

ntsb\_no |CEN23MA034 |

ev\_type |ACC |

ev\_year |2022 |

ev\_month |11 |

ev\_dow |SA |

country |USA |

state |TX |

city |Dallas |

zip |75237 |

acft\_categories |{AIR,AIR} |

all\_defining\_occurrence\_codes|{452250,452250} |

all\_finding\_codes |{} |

all\_occurrence\_codes |{452250,452250} |

cictt\_codes |{MAC} |

dec\_latitude |32.673779 |

dec\_longitude |-96.862801 |

dest\_countries |{} |

dprt\_countries |{} |

ev\_counter |1 |

ev\_highest\_injury |FATL |

far\_parts |{091,091} |

finding\_codes |{} |

inj\_f\_grnd |0 |

inj\_tot\_f |6 |

is\_altitude\_controllable |false |

is\_altitude\_low | True |

is\_atttitude\_controllable |false |

is\_dest\_country\_usa |false |

is\_dprt\_country\_usa |false |

is\_emergency\_landing | True |

is\_far\_part\_091x | True |

is\_far\_part\_121 |false |

is\_far\_part\_135 |false |

is\_midair\_collision | True |

is\_narrative\_stall |false |

is\_oper\_country\_usa | True |

is\_owner\_country\_usa | True |

is\_pilot\_issue |false |

is\_regis\_country\_usa | True |

is\_spin\_stall |false |

latlong\_acq |MEAS |

no\_aircraft |2 |

occurrence\_codes |{MAN\_LALT,MAN\_LALT}|

oper\_countries |{USA,USA} |

owner\_countries |{USA,USA} |

regis\_countries |{NON-US,USA} |

regis\_nos |{N6763,N7227C} |

### Table **events**

#### Column **io\_latlong\_acq**

Extending the categories used in the **latlong\_acq** column:

* **LOLA**: Correction based on swapped latitude and longitude
* **LALO**: Correction based on latitude and longitude
* **ERRA**: Invalid latitude string
* **ERRO**: Invalid longitude string

#### Corrections with csv files

The values of the columns **city**, **country**, **latitude**, **longitude**, **site\_zipcode** and **state** can be corrected via csv files and the changed value is used in all subsequent processing. The changed values are stored in the following IO-Aero specific columns:

* **io\_city**
* **io\_country**
* **io\_latitude**
* **io\_longitude**
* **io\_site\_zipcode**
* **io\_state**

#### Determination of the missing values of **dec\_latitude** / **dec\_longitude**

* **io\_dec\_lat\_lng\_actions** - shows the actions performed to find the missing values in the **dec\_latitude** and **dec\_longitude** columns.
* **io\_dec\_latitude** - the found value for the **dec\_latitude** column.
* **io\_dec\_latitude\_deviating** - Difference between latitude and decimal latitude if a predefined threshold value is exceeded.
* **io\_dec\_longitude** - the found value for the **dec\_longitude** column.
* **io\_dec\_longitude\_deviating** - Difference between longitude and decimal longitude if a predefined threshold value is exceeded.

#### Error Flags

* **io\_invalid\_latitude** - invalid value in column **latitude**.
* **io\_invalid\_longitude** - invalid value in column **longitude**.
* **io\_invalid\_us\_city** - unknown value in **ev\_city** column.
* **io\_invalid\_us\_city\_zipcode** - unknown combination of values in columns **ev\_site\_zipcode** and **ev\_city**.
* **io\_invalid\_us\_state** - unknown value in **ev\_state** column.
* **io\_invalid\_us\_zipcode** - unknown value in **ev\_site\_zipcode** column.

### Table **io\_aviation\_occurrence\_categories**

This table contains CICTT codes from the document Aviation Occurrence Categories (AVIATION OCCURRENCE CATEGORIES DEFINITIONS AND USAGE NOTES):



#### Sample data:

**Name |Value**

---------------+-----------------------------------------------------------

cictt\_code |LOC-I

identifier |LOSS OF CONTROL–INFLIGHT

definition |Loss of aircraft control while, or deviation from inten ...

first\_processed|2023-01-18 11:07:38.756

last\_processed |

last\_seen |2023-01-18 10:07:37.837

#### Data source:



The data from this document (AVIATION OCCURRENCE CATEGORIES DEFINITIONS AND USAGE NOTES, 2013) has been extracted into the MS Excel file

**Aviation\_Occurrence\_Categories/aviation\_occurrence\_categories.xlsx**

and can be loaded from there into the database table.



### Table **io\_countries**

This table mainly contains data on latitudes and longitudes of countries. Currently, only the values of the USA are stored.

The latitude and longitude stored here will be applied to US events that do not themselves contain decimal latitude and longitude and which cannot be determined at the zip code, city or state level.



#### Sample data:

**country|country\_name |dec\_latitude|dec\_longitude|**

-------+-------------+------------+-------------+

USA |United States| 37.09024| -95.712891|

#### Data source:



The data must be provided in a JSON file with the following format:



### Table **io\_lat\_lng**

This is the main table for determining missing decimal latitude or longitude for US events. The table is fed by the following sources:

* simplemaps United States Cities Database (simplemaps.com. (n.d.))
* simplemaps US Zip Codes Database (simplemaps.com. (n.d.))
* Zip Codes.org ZIP Code Database (UnitedStatesZipCodes. (n.d.))

In addition, the missing values for cities are averaged from the values of the matching zip codes.



#### Sample data:

**Name |Value |**

---------------+--------------------------------+

id |13620 |

type |ZIPCODE |

country |USA |

state |SC |

city |ABBEVILLE |

zipcode |29620 |

dec\_latitude |34.1813 |

dec\_longitude |-82.42804 |

source |simplemaps US Zip Codes Database|

first\_processed|2022-11-29 13:11:16.347 |

last\_processed |2022-11-29 13:28:49.520 |

#### Data source simplemaps

Data from the simplemapsflat files **uscities.xlsx** and **uszips.xlsx** is loaded into the PostgreSQL database using the following processing logic:

1. the existing data with column **source** equal to ‘**simplemaps United States Cities Database’** or ‘**simplemaps US Zip Codes Database**’ are deleted,
2. the database United States Cities is processed, whereby for each zip code in the column **zips** an entry in the PostgreSQL table **io\_lat\_lng** is created (not updated) - as latitude and longitude the corresponding values of the affected city are used, with column **type** equal to ‘**ZIPCODE**’ and column **source** equal to ‘**simplemaps United States Cities Database**’,
3. the database US Zip Codes is processed, whereby for each zip code an entry in the PostgreSQL table **io\_lat\_lng** is created or updated with column **type** equal to ‘**ZIPCODE**’ and column **source** equal to ‘**simplemaps US Zip Codes Database**’,
4. the database United States Cities is processed, whereby for each city an entry in the PostgreSQL table **io\_lat\_lng** is created or updated with column **type** equal to ‘CITY’ and column **source** equal to ‘**simplemaps United States Cities Database**’,
5. the existing rows of the database table **io\_lat\_lng** with column **source** equal to ‘**average**’ are deleted,
6. from the lines of the database table **io\_lat\_lng** of type ‘**ZIPCODE**’ the average of latitude and longitude per city is calculated and this value is stored as latitude and longitude of this city in the database table **io\_lat\_lng** with **source** equal to ‘**average**’.

##### Resulting database table **io\_lat\_lng**:



##### Data source **United States Cities Database**



##### Data source **US Zip Codes Database**



#### Data source Zip Codes.org

Data from the Zip Codes.org flat file **zip\_code\_database.xls** is loaded into the PostgreSQL database using the following processing logic:

1. the existing data with column source equal to ‘Zip Codes.org ZIP Code Database’ are deleted,
2. the Excel rows of **type** equals ‘**STANDARD**’ and **country** equals ‘**US**’ are inserted or updated in the database table **io\_lat\_lng** with column **source** equal to ‘**Zip Codes.org ZIP Code Database**’,
3. the existing rows of the database table **io\_lat\_lng** with column **source** equal to ‘**average**’ are deleted,
4. from the lines of the database table **io\_lat\_lng** of **type** ‘**ZIPCODE**’ the average of latitude and longitude per city is calculated and this value is stored as latitude and longitude of this city in the database table **io\_lat\_lng** with **source** equal to ‘**average**’.

##### Resulting database table **io\_lat\_lng**:



##### Data source **ZIP Code Database**



### Table **io\_md\_codes\_category**

This table contains the category codes extracted from the **findings** table.



#### Sample data:

**Name |Value |**

-------------+--------+

category\_code|01 |

description |Aircraft|

### Table **io\_md\_codes\_eventsoe**

This table contains the eventsoe codes extracted from the **events\_sequence** table.



#### Sample data:

**Name |Value |**

-------------+-----------------------------+

eventsoe\_code|490 |

description |Collision during takeoff/land|

### Table **io\_md\_codes\_modifier**

This table contains the modifier codes extracted from the **findings** table.



#### Sample data:

**Name |Value |**

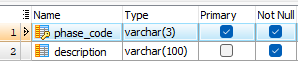
-------------+-------+

modifier\_code|01 |

description |Failure|

### Table **io\_md\_codes\_phase**

This table contains the phase codes extracted from the **events\_sequence** table.



#### Sample data:

**Name |Value |**

-----------+-----------------+

phase\_code |453 |

description|Maneuvering-hover|

### Table **io\_md\_codes\_section**

This table contains the section codes extracted from the **findings** table.



#### Sample data:

**Name |Value |**

----------------+---------+

category\_code |01 |

subcategory\_code|01 |

section\_code |00 |

description |(general)|

### Table **io\_md\_codes\_subcategory**

This table contains the subcategory codes extracted from the **findings** table.



#### Sample data:

**Name |Value |**

----------------+-------------+

category\_code |02 |

subcategory\_code|02 |

description |Psychological|

### Table **io\_md\_codes\_subsection**

This table contains the subsection codes extracted from the **findings** table.



#### Sample data:

**Name |Value |**

----------------+--------------------------+

category\_code |01 |

subcategory\_code|02 |

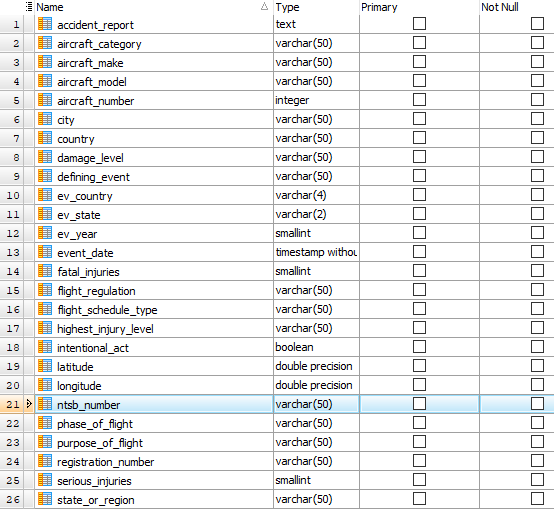
section\_code |37 |

subsection\_code |10 |

description |Vacuum distribution system|

### Table **io\_ntsb\_2002\_2021**

In the November 17, 2021, news release U.S. Civil Aviation Fatalities and Flight Activity Decreased in 2020 (www.ntsb.gov, n.d.), the NTSB published, among other things, the US Civil Aviation Accident Statistics (Ntsb.gov, 2023). These are available on this website as an MS Excel file. This file contains 28 worksheets (Table 1 through Table 28) of accumulated data, and in Worksheet 29 (Table 29. Accident Aircraft, 2002 through 2021, US Civil Aviation) (most likely) the underlying event and aircraft details. The data from Worksheet 29 were loaded unchanged into this database table. To simplify the use of the database table, only the columns **ev\_year**, **ev\_state** and **ev\_country**, derived from the columns **event\_date**, **state**\_or\_**region** and **country**, were added. The header lines and the empty lines for the loading process were ignored.



#### Sample data:

**Name |Value**

--------------------+------------------------------------------------

ntsb\_number |MIA02LA056

accident\_report |=HYPERLINK("http://data.ntsb.gov/...","Report")|

event\_date |2002-01-20 00:00:00.000

ev\_year |2002

city |Sarasota

state\_or\_region |Florida

ev\_state |FL

country |United States

ev\_country |USA

latitude |27.395277

longitude |-82.553886

fatal\_injuries |

serious\_injuries |

highest\_injury\_level|None

aircraft\_number |1

damage\_level |Substantial

registration\_number |N235AL

aircraft\_category |Airplane

aircraft\_make |Maule

aircraft\_model |M-7-235B

flight\_regulation |Part 91: General aviation

flight\_schedule\_type|

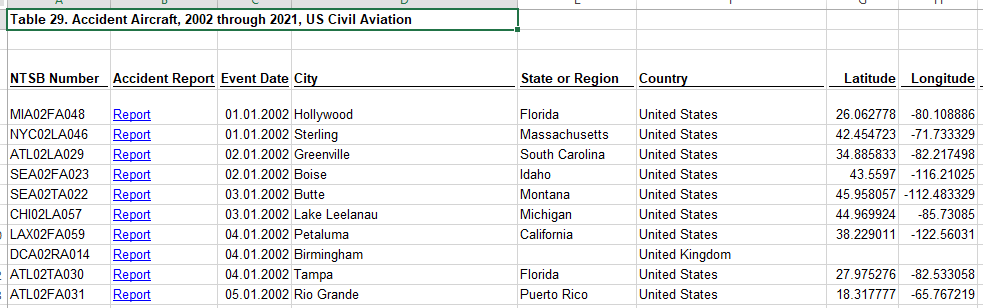
purpose\_of\_flight |Personal

intentional\_act |

defining\_event |

phase\_of\_flight |

#### Data source:



### Table **io\_processed\_files**

This table documents which external files were processed to create and maintain the database content.



#### Sample data:

**Name |Value |**

---------------+-----------------------+

file\_name |up22NOV |

first\_processed|2022-11-27 13:40:18.094|

last\_processed |2022-12-12 12:08:15.915|

counter |2 |

### Table **io\_sequence\_of\_events**

This table enables the assignment of CICTT codes via the event sequence number.



#### Sample data:

**soe\_no|cictt\_code|meaning**

------+----------+-----------------------------------------

0|UNK |Unknown or undetermined

10|RAMP |Aircraft loading event

20|RAMP |Aircraft servicing event

30| |Preflight or dispatch event

40| |Aircraft maintenance event

50| |Aircraft inspection event

60| |Attempted remediation/recovery

70|ADRM |Airport occurrence

80|RAMP |Ground handling event

81|RAMP |AC/prop/rotor contact w person

82|RAMP |Prop/jet/rotor blast/suction

#### Data source CICTT\_SOE\_MAP.csv (file given by NTSB):



### Table **io\_states**

This table mainly contains data on latitudes and longitudes of US states.

The latitude and longitude stored here will be applied to US events that do not themselves contain decimal latitude and longitude and which cannot be determined at the zip code or city level.



#### Sample data:

**country|state|state\_name |dec\_latitude |dec\_longitude |**

-------+-----+--------------------+-------------+--------------+

USA |AK |Alaska |63.7431630974|-151.594035116|

USA |AL |Alabama |32.7570463396| -86.844525962|

#### Data source:



Same JSON file as with table io\_countries.

View **io\_lat\_lng\_issues**

This view shows the related data from the tables **events**, **io\_countries**, **io\_lat\_lng** and **io\_states** for those events in the USA where either the decimal latitude or longitude is missing from the NTSB data.



#### Sample data:

**Name |Value**

--------------------------+-----------------------------------------------

ev\_id |20030618X00903

ev\_country |USA

ev\_state |AK

ev\_city |Anchorage

ev\_site\_zipcode |99515

ev\_dec\_latitude |

ev\_dec\_longitude |

ev\_latitude |610624N

ev\_longitude |1495152W

io\_country |

io\_state |

io\_city |

io\_site\_zipcode |

io\_dec\_latitude |61.10666666666667

io\_dec\_longitude |-149.86444444444444

io\_latitude |

io\_longitude |

io\_dec\_lat\_lng\_actions |INFO.00.037 Correction based on latitude and l…

io\_dec\_latitude\_deviating |

io\_dec\_longitude\_deviating|

io\_invalid\_latitude |

io\_invalid\_longitude |

io\_invalid\_us\_city |

io\_invalid\_us\_city\_zipcode|

io\_invalid\_us\_state |

io\_invalid\_us\_zipcode |

country |USA

state |AK

city |Anchorage

site\_zipcode |99515

latitude |610624N

longitude |1495152W

state\_name |Alaska

zipcode\_dec\_latitude |61.11733

zipcode\_dec\_longitude |-149.88894

city\_dec\_latitude |61.1508

city\_dec\_longitude |-149.1091

state\_dec\_latitude |63.7431630974

state\_dec\_longitude |-151.594035116

country\_dec\_latitude |37.09024

country\_dec\_longitude |-95.712891

# Data Cleansing

## Decimal latitude and longitude

The values in the following columns will be filled when executing task **c\_l\_l** (Correct decimal US latitudes and longitudes):

* **io\_dec\_latitude**
* **io\_dec\_longitude**
* **io\_dec\_lat\_lng\_actions**

The columns **dec\_latitude** and **dec\_longitude** were added by NTSB in 2008. For data from previous years, the two columns remained empty (NULL). **IO-Aero** pursues the following two goals with the **c\_l\_l** task:

* to check the validity of the existing values in the columns **latitude** and **longitude** and
* to find the missing values in the columns **dec\_latitude** and **dec\_longitude** for rows with **ev\_country** equal to **'USA'**.

The underlying algorithm logs its processing steps in the **io\_dec\_lat\_lng\_actions** column - see the following examples:

**io\_dec\_lat\_lng\_actions |**

---------------------------------------------------------------------------------+

ERROR.00.916 Unknown US state and city & INFO.00.035 Correction based on US state|

ERROR.00.922 Invalid US state id & INFO.00.033 Correction based on US zip code |

---------------------------------------------------------------------------------+

INFO.00.033 Correction based on US zip code |

INFO.00.034 Correction based on US state and city |

INFO.00.037 Correction based on latitude and longitude |

## Trimmed Database Columns

The following database columns are trimmed because they contain different whitespace elements:

|  |  |
| --- | --- |
| Table | Column |
| aircraft | acft\_category |
| aircraft | dest\_country |
| aircraft | dprt\_country |
| aircraft | far\_part |
| aircraft | oper\_country |
| aircraft | owner\_country |
| aircraft | regis\_no |
| events | ev\_city |
| events | ev\_site\_zipcode |
| events | latitude |
| events | longitude |
| events\_sequence | occurrence\_code |

# Change Log

#### Release 23.04.15

* Database column added:
  + **io\_app\_ae1982.description\_main\_phase\_ defining**
  + **io\_md\_codes\_phase.description\_main\_phase**
* NTSB files included:
  + **Up15APR.zip** - 4/15/2023 3:01:20 AM
* simplemaps files included:
  + simplemaps\_uscities\_basicv1.76
  + simplemaps\_uszips\_basicv1.82
* United States Zip Codes.org file included:
  + zip\_code\_database.xls (42735 entries)

#### Release 23.04.08

* NTSB files included:
  + **Pre2008.zip** - 9/30/2020 12:51:56 PM
  + **avall.zip** - 4/ 3/2023 8:13:22 AM
  + **up08APR.zip** - 4/ 8/2023 3:01:20 AM

#### Release 23.04.01

* Database column added:
  + **io\_app\_ae1982.phase\_codes\_defining**
* NTSB files included:
  + **Pre2008.zip** - 9/30/2020 12:51:56 PM
  + **avall.zip** - 3/ 1/2023 6:01:23 AM
  + **up08MAR.zip** - 3/ 8/2023 3:00:22 AM
  + **up15MAR.zip** - 3/15/2023 3:00:11 AM
  + **up22MAR.zip** - 3/22/2023 3:00:14 AM
  + **up01APR.zip** - 4/ 1/2023 3:00:31 AM

# References

**AVIATION OCCURRENCE CATEGORIES DEFINITIONS AND USAGE NOTES. (2013)**. Available at: <https://www.ntsb.gov/safety/data/Documents/datafiles/OccurrenceCategoryDefinitions.pdf> [Accessed 28 Mar. 2023].

**data.ntsb.gov. (n.d.)**. *MDB Download Directory - NTSB.ADMS.DataTransfer.Web*. [online] Available at: <https://data.ntsb.gov/avdata>.

**Docker Documentation. (2022)**. *Get Docker*. [online] Available at: <https://docs.docker.com/get-docker/>.

**Geodatos.net. (2019)**. *GeoDatos - Information and maps of the world*. [online] Available at: <https://www.geodatos.net/en>.

**hub.docker.com. (n.d.)**. *Docker Hub*. [online] Available at: <https://hub.docker.com/_/postgres>.

**members.io-aero.com. (n.d.)**. *Streamlit*. [online] Available at: <http://members.io-aero.com/> [Accessed 28 Mar. 2023].

**Ntsb.gov. (2023)**. Available at: <https://www.ntsb.gov/safety/data/Documents/AviationAccidentStatistics_2002-2021_20221208.xlsx> [Accessed 29 Mar. 2023].

**Opendatasoft. (n.d.)**. *Create the best data experiences*. [online] Available at: <https://www.opendatasoft.com/?hsLang=en> [Accessed 28 Mar. 2023].

The **PostgreSQL Global Development Group** (2019). *PostgreSQL: The world’s most advanced open source database*. [online] Postgresql.org. Available at: <https://www.postgresql.org/>.

**razorsql.com. (n.d.)**. *RazorSQL - SQL Query Tool and SQL Editor for Mac, Windows, and Linux*. [online] Available at: <https://razorsql.com/>.

**simplemaps.com. (n.d.)**. *Interactive HTML5 and JavaScript Maps for Websites | Simplemaps.com*. [online] Available at: <https://simplemaps.com> [Accessed 28 Mar. 2023].

**UnitedStatesZipCodes. (n.d.)**. *U.S. ZIP Codes: Free ZIP code map and zip code lookup*. [online] Available at: <https://www.unitedstateszipcodes.org>.

**www.ntsb.gov. (n.d.)**. *U.S. Civil Aviation Fatalities and Flight Activity Decreased in 2020*. [online] Available at: <https://www.ntsb.gov/news/press-releases/Pages/NR20211117.aspx>.