Group 03 - Project Phase 1 and Phase 2 Aviation Accident Data Integration

Tommaso Tragno
Faculdade de Ciências
Universidade de Lisboa
Lisboa, Portugal
fc64699@alunos.fc.ul.pt

Manuel Cardoso
Faculdade de Ciências
Universidade de Lisboa
Lisboa, Portugal
fc56274@alunos.fc.ul.pt

Chen Cheng
Faculdade de Ciências
Universidade de Lisboa
Lisboa, Portugal
fc64872@alunos.fc.ul.pt

Cristian Tedesco
Faculdade de Ciências
Universidade de Lisboa
Lisboa, Portugal
fc65149@alunos.fc.ul.pt

***Abstract*—In this project, we investigate the correlation between U.S. aviation accident rates, passenger traffic volumes (a proxy for tourism), environmental conditions, and aircraft production details. We integrate four primary data sources: (1) aviation accident records from the National Transportation Safety Board (NTSB), (2) nationwide airline traffic data from a Kaggle dataset, (3) historical weather information from Open-Meteo's API, and (4) an aircraft production dataset (Kaggle) containing manufacturer and model details. Our integration pipeline addresses challenges such as inconsistent identifiers (e.g., aircraft codes), differing temporal resolutions, and missing or incomplete records across data sources. We employ schema integration, identity-resolution strategies (including blocking rules for aircraft data), and data-profiling/cleaning methods to create a unified dataset spanning the years 2003–2023. By leveraging this integrated dataset, we explore whether higher travel demand coincides with increased accident frequency, how meteorological factors amplify risks, and whether certain aircraft models or production eras exhibit higher accident rates in adverse conditions. The insights aim to inform policy and operational decisions by regulatory authorities, airlines, and airport management teams seeking data-driven approaches to enhancing aviation safety.**

***Index Terms*—Aviation accidents, Data integration, Identity resolution, Weather data, Airline traffic, Tourism, Data profiling, Data cleaning, Schema integration, Correlation analysis.**

I. Project Motivation and Questions

Civil aviation is one of the safest modes of transportation; however, any aviation accident or incident draws significant public attention and can lead to changes in regulatory oversight, operational practices, and technological innovation. Traditional analyses of aviation safety frequently emphasize mechanical reliability or human factors, yet comparatively fewer studies explore how environmental conditions, seasonal tourism fluctuations, and even aircraft manufacturing details might influence safety outcomes. Understanding these correlations could help better anticipate and mitigate risks under varying conditions of demand and environment.

Our project integrates four real-world datasets—spanning accident data, airline passenger traffic, weather conditions, and aircraft production details—to investigate the following key questions:

- **Tourism and Safety:** Does higher passenger volume—often associated with peak travel seasons—correlate with an increase in aviation accidents or incidents?
- **Weather Influence:** Do weather conditions, such as temperature extremes, precipitation, or high wind speeds, significantly coincide with higher accident rates or more severe accident outcomes?
- **Regional Variations:** Are certain locations or airports more susceptible to the combined effects of weather patterns and elevated passenger volumes, leading to a higher incidence of accidents?
- **Aircraft Manufacturing and Age:** Does the make, model, or production era of an aircraft correlate with an elevated accident rate, especially in times of high passenger volume or under poor weather conditions?

In addressing these questions, we will employ a range of data-processing methods, including schema integration, identity resolution, data profiling, and cleaning. Ultimately, our goal is to provide actionable insights for aviation stakeholders—airlines, airports, and regulators—seeking data-driven strategies to enhance safety protocols and resource allocation under conditions of fluctuating demand and variable weather.

II. Data Sources and Collections

To investigate the aforementioned questions, we draw on four real-world data sources covering overlapping time periods, each contributing complementary information about aviation accidents, passenger traffic, weather conditions and aircraft characteristics:

1) **NTSB Aviation Accident Database** - Contains detailed records of accidents, including location, date/time, flight phase, and possible contributing factors [5]. It spans a broader time period and includes global accident reports, but for our scope it has been filtered to U.S.-based accidents between 2003 and 2023, in order to match the scope of the Kaggle airline-traffic dataset. It has attribute like NtsbNumber, EventDate, City, State, Make, Model, RegistrationNumber, HighestInjury, AirportId, Latitude, Longitude, Narrative, ecc...
2) **U.S. Airline Traffic Data (Kaggle)** - Provides monthly flight volumes, passenger counts, and other tourism-related indicators for both domestic and international flights [9]. It spans a 2003–2023 coverage, domestic U.S. flights only, representing the shortest time range among the chosen datasets. For this reason, it is used in full.

Some attribute example are Year, Month, AirportCode, Passengers, Flights, AvailableSeats, LoadFactor, ecc...

3) **Open-Meteo API** - Returns hourly or daily weather data (temperature, precipitation, wind speed, wind direction, etc.) for specified coordinates and dates [6]. It has a worldwide coverage, including a broad historical range if requested, but for our scope it has been filtered to U.S. accident site coordinates from 2003–2023, to capture hourly weather conditions near each accident or flight origin/destination. Some attribute example are Latitude, Longitude, Temperature_2m, Precipitation, CloudCover, WindSpeed_10m, WindDirection_10m
4) **Aircraft Production Data (Kaggle)** Contains details on various aircraft manufacturers, models, and production volumes [1]. Includes worldwide production data with no strict time boundary for each aircraft type. It is used in full. Some attribute example are ManufacturerName, ModelName, FirstProductionYear, LastProductionYear, ProductionCount, AircraftCategory.

Although the NTSB and Open-Meteo data can extend to broader geographic regions and longer timeframes, we focus on U.S. data between 2003 and 2023 to align all three sources to a common spatiotemporal scope. The Kaggle airline-traffic dataset is the limiting factor here, as its coverage is restricted to U.S. domestic flights over that same 20-year window. Consequently, we constrain the NTSB accident data and the Open-Meteo queries to U.S. airports within the 2003–2023 timeframe to maintain consistency and facilitate a direct comparison of accidents, travel volumes, and weather variables.

III. Data Profiling & Cleaning

The datasets used found themselves to be of good quality, not needing much cleaning:

1) For **NTSB Aviation Accident Database** key aspects in cleaning were dropping columns we were not going to use, convert data types (for instance to standardized date/time formats ISO 8601), make all appropriate values lowercase and removing entries that had ”EventDate” column empty, as this column is essential for us.
2) For **U.S. Airline Traffic Data (Kaggle)**, similar to the previous, we just had to drop unnecessary columns and convert types, but, before that, we also had to remove commas from the integer values, in order to allow a proper casting. To check the data distribution, we also plotted an histogram that compares the number of flights per month throughout all the years, as shown in Figure 1.
3) For **Open-Meteo API** since we used the API, we only gathered the data we wanted, so there was no need for cleaning, at least for now, because we believe we may have gathered more data then we will use, which we will find out in the future.
4) For **Aircraft Database** we had to drop a single column, ”retired” and convert the aircraft models names’ to lowercase. We noticed som aircraft had incorrect date values, as shown in Figure 2. Since there were few of this mismatch, we manually searched for their production years (start and end years) and updated the dataset.

In the future, we believe the NTSB dataset will provide us some challenges because in the process of flattening it, we induced duplicates, that although needed, may become problems, as shown in Figure 3. Also in this dataset, some entries had null values in the ”Longitude” and ”Latitude” columns, as shown in Figure 4, which aren’t a problem itself but will prevent us from correlating those entries to entries on the weather dataset.

IV. Schema Integration

A. Dataset-level conceptual models

Each of the datasets conceptual models can be found in the Appendix, in Figure 5, aswell as the characterization of their attributes, in Tables II, III, IV and V.

B. Integrated model

For our Integrated model, we chose which data was important to keep in order to answer our question, and then we added all the information from the different tables to the integrated model, adding relationships with the highest cardinality while ensuring data consistency, integrity, and efficiency, making it easier for querying and analysis. Our proposed schema, as shown in Table VI, encompasses different types of correspondences between the data sources, like many-to-one correspondences, also using binding methods described in the section IV-C. We also joined ”FatalInjuryCount”, ”SeriousInjuryCount” and ”MinorInjuryCount” columns into a single one: ”TotalInjuryCount”, being the sum of the three columns.

C. Identity Resolution

To address ambiguities in identifying aircraft models between the NTSB dataset and the Aircraft Production Data, we implemented a text-based identity resolution strategy. Specifically, the *Model* field from each NTSB incident record was compared with the *aircraft* field from the external dataset, which contains manufacturing and specification data. Given that model names may differ due to formatting, abbreviations, or data entry inconsistencies, we adopted a two-step approach combining blocking with string similarity computation. **Blocking** was performed using q-gram overlap (substrings of length 3), retaining only pairs that share at least two q-grams or include one another as substrings. Additionally, we applied a numerical consistency filter, requiring that numeric substrings in both model names match (e.g., avoiding false matches like *”B737”* vs *”B747”*). For the similarity phase, we computed three distinct string similarity metrics using the *py_stringmatching* library: **Jaro-Winkler** (for phonetic similarity), **Levenshtein** (edit distance), and **Jaccard** (token-based, effective for multi-word models). These were combined into a weighted linear rule, and model pairs exceeding a threshold score (0.75) were accepted as matches. This approach produced 38 valid matches, allowing us to associate incident records with real-world aircraft production data. These matches provide a basis for future analysis of potential correlations between aircraft type, production volume or age, and safety outcomes, particularly under specific environmental or operational conditions. As a next step, we aim to optimize the algorithm by reducing its time complexity, improving scalability and efficiency when applied to larger datasets or real-time scenarios.

V. Tools & Libraries Phase I

During this project, we used a combination of open-source libraries in a Python/Jupyter environment to perform data extraction, cleaning, integration, and exploratory analysis. Below is an overview of the key tools employed: - Jupyter Notebook provided an interactive environment for writing and running Python code, documenting data explorations, and visualizing results [4]. - Pandas offered flexible and efficient data structures (DataFrame) for data manipulation and analysis. It has been employed extensively to import CSV/JSON files, handle missing values, and summarize or filter large datasets [7]. - NumPy for fast statistical calculations, and type conversions to/from Pandas DataFrames [2]. - The Open-Meteo historical weather API was leveraged to fetch meteorological observations using Python's built-in requests library to query the API endpoints, receiving weather data in JSON form, which was then parsed and appended to local data frames [6]. - Matplotlib used to create histograms, scatterplots, and time-series charts for outlier detection, correlation checks between accident rates, flight volumes, and weather conditions [3]. - Py_StringMatching is a specialized library for computing a variety of string similarity and distance metrics. It was used to support the identity resolution process by comparing aircraft models and operator names across heterogeneous datasets, enabling robust string comparisons despite inconsistent formatting or minor spelling variations [8].

VI. Changes and Improvements

After the work in Phase 1, we took the time to revise and improve both the integrated schema and the entity-relationship model to better reflect the structure and relationships within our data, as also suggested by the peer reviews. The updated versions can be seen in Table VII and Figure 6. We replaced the original Aircraft Production Data with the ICAO API Data Service, which provides comprehensive, up-to-date official type designators (DOC 8643) and offers richer attributes—such as manufacturer_code, model_name, engine_type, and wtc—to gain deeper insights and build a much more robust data model. The matching and blocking strategy was accordingly adapted between the NTSB and ICAO datasets to accurately align corresponding aircraft models. The new research question is: "Is the aircraft's engine_type a determining factor in the likelihood or severity of an aviation accident?"

VII. Data Fusion

As previously mentioned, our datasets are of good quality and share few common attributes. For this reason, in this section, we had to duplicate the NTSB dataset to demonstrate how certain data fusion strategies work. In addition to these simulated examples, several advanced data fusion strategies were applied within the actual scope of the project:

- Temporal Fusion for the Weather Integration was integrated by selecting the closest weather observation (within a ±3-hour window) for each accident.
- Record Linkage (Weather Location Match), performed with coarse spatial filtering and string parsing to extract and match against the NTSB accident location.
- Tiered Conflict Resolution (Engine Count) to fix discrepancies between NTSB-reported engine counts and the authoritative aircraft database.
- Temporal Fusion (Airline Integration), to incorporate contextual information about aviation activity,

To simulate data fusion, we began by duplicating the NTSB dataset twice: one copy served as the target dataset, and the other to be fused into it. In the target dataset, we intentionally set 30% of the 'State' values to 'NaN' to later address them using slot filling. In the second duplicate, we renamed three columns and aimed to modify 30% of the 'AirportName' values by replacing them with abbreviations, in order to generate conflicts. For example, 'Chickasha Municipal Airport' became 'C. M. A.'. So on this part, the strategies implemented were:

- **Slot Filling:** After introducing 30% missing values into the 'State' column, it contained a total of 7050 'NaN' entries. Following the data fusion process between the two datasets, only 47(0.67% of total records) of these missing values remained. We believe this residual number is due to the presence of 'NaN' values that already existed in the original dataset prior to our induced modifications.
- **Schema Matching:** Since not all column names matched, we had to rename those containing equivalent information to ensure they could be properly aligned and fused later.
- **Conflict Resolution:** With the goal of abbreviating 30% of the values, we generated a total of 5125 abbreviated entries. As before, we believe this number was slightly reduced due to existing missing values in the dataset, since the entries were selected at random. After the data fusion process, only 20 abbreviated values remained (0.39% of total records). In addition, a dedicated conflict resolution step was applied for numeric fields in the integrated aircraft data. Specifically, when `Vehicles.NumberOfEngines` conflicted with `engine_count`, we applied a three-rule strategy:
  1) Fill null values in `Vehicles.NumberOfEngines` using `engine_count`.
  2) Replace zero values with `engine_count` when the latter is positive.
  3) Overwrite mismatches when both values were non-zero but inconsistent, prioritizing the `engine_count` values, because more accurate.

  This approach resolved all engine count conflicts programmatically, avoiding the need for manual inspection.
- **Record Linkage and Temporal Fusion (Weather Integration):** To enrich the accident records with meteorological context, we performed record linkage and temporal fusion using the Open-Meteo dataset. A two-step strategy was implemented:

- **Record Linkage:** We extracted latitude and longitude from the weather data's `AccidentID` field and matched it to NTSB coordinates using a spatial threshold ($= 0.10^\circ$, $\approx 11km$).
- **Temporal Fusion:** For each accident, we selected the weather entry closest in time (within a ±3-hour window) using `abs(weather`$_t ime$ – $accident_t ime), and assigned it to the accident record.$

This approach resulted in a high-quality fusion:
- 18,255 of 20,000+ accidents were successfully matched to a weather record ($\approx 91.2\%$).
- Mean time delta between matched entries was within 30 minutes.
- Spatial deltas were within the expected geographic tolerance.

- **Temporal Fusion (Airline Data):** Another application of temporal fusion involved integrating U.S. airline traffic data with the accident records. While accident-level matching was not feasible due to the lack of direct identifiers, a monthly-level temporal fusion was applied using the accident date. This allows aggregate-level insights (e.g., passenger volumes, travel trends) to be linked to aviation safety events.
- **Deduplication:** As the final strategy applied, we performed deduplication to ensure there were no repeated records after merging both datasets. This process revealed only 3 duplicated records (0.01% of total records).

The table I show the summary for the fusion strategy.

VIII. Data Pipelines

Our project supports any input datasets, provided they share the same format as the ones we used. To run the project, simply select "Run All" in the Jupyter Notebook. The only steps performed manually are the 'Bert text classification,' due to its high computational demands, and the OpenMeteo API calls, which have a limited number of daily calls, and may require days of execution. For this, we include an additional script called "Bert_text_classification.py", which generates the file "ntsb_with_zero_shot.csv". This file is then used later for statistical analysis.

IX. Results

At the start of this project, we formulated four main questions, presented in Section I. Through the datasets we collected and the data-related tasks we implemented, we were able to answer some of them. However, not all questions could be fully addressed, as we had to revise or restructure certain ones due to limitations in the available data.

For the first question – "Does higher passenger volume—often associated with peak travel seasons—correlate with an increase in aviation accidents or incidents?" – we calculated the correlation between the total number of flights per month (Flt) and the number of recorded accidents, obtaining a coefficient of 0.56. This indicates a moderate positive relationship, suggesting that higher air traffic volumes may be associated with an increased number of incidents, although not strongly. To address the second and third questions—on the influence of weather and regional variation on accident outcomes—we applied a BERT-based zero-shot classifier to the ProbableCause field, assigning each incident to a structured category: Collision / Obstacle, Loss of Situational Awareness, Human Error – Control, Human Error – Procedural, Environmental Conditions, Fuel Management, and Mechanical Failure. This allowed us to reformulate the research focus as: "How are different types of accident causes associated with both varying weather conditions and geographic regions at the time of the incident?"

We estimated a multinomial logistic regression model using variables such as temperature, precipitation, wind gusts, cloud cover, season, time of day, and region. VIII Collision / Obstacle was selected as the baseline category, as it is the most frequent and structurally distinct class—serving as a stable reference to assess deviations driven by environmental or human factors.

The results show that environmental causes are significantly associated with precipitation, low temperatures, strong wind gusts, and cloud cover. Technical failures tend to occur in warmer, windier months, particularly in Central and Southern regions. Human error is linked to windy but clear conditions, more frequently in spring and during daylight hours. Loss of situational awareness is significantly associated with high temperatures, winter conditions, and again, with incidents in Central and Southern areas.

For the fourth question – "Does the make, model, or production era of an aircraft correlate with an elevated accident rate, especially in times of high passenger volume or under poor weather conditions?" – we restructured the question due to data limitations. Instead, we asked: "What is the most common engine type among aircraft involved in accidents?". To answer this, we constructed a sample through a matching strategy between the NTSB accident records and the ICAO Aircraft dataset, which provided the engine type information not originally available in the NTSB data. To evaluate whether the matched sample was representative of the full NTSB dataset in terms of aircraft characteristics, we conducted a Chi-squared test comparing the distribution of Vehicles.Make in the matched sample against that in the complete NTSB dataset. The test returned a p-value close to zero, indicating a statistically significant difference in manufacturer distributions. Therefore, the sample cannot be considered representative of the full dataset. Despite this limitation, the analysis shows that piston engines are by far the most common engine type among aircraft in the matched sample, accounting for over 85% of the total. Turboprop/turboshaft and jet engines appear much less frequently, suggesting that most incidents in this subset involve general aviation aircraft, rather than commercial jets or turbine-powered platforms.

X. Tools & Libraries Phase II

Seaborn was used alongside Matplotlib to improve the aesthetics and clarity of plots. It facilitated the creation of cor-

TABLE I: Summary of Data Fusion Strategies and Results

Strategy	Description	Matching Rules	Result
Weather Temporal Fusion	Assign nearest weather record by time and location	$\lvert\text{lat} - \text{lat}'\rvert < 0.10°$, $\lvert\text{lon} - \text{lon}'\rvert < 0.10°$, $\lvert\Delta t\rvert \leq 3h$	91.2% (18,255 / 20,000)
Weather Spatial Linkage	Parse coordinates from `AccidentID` and match by spatial proximity	Absolute differences in latitude and longitude	Same as above
Aircraft Schema Matching	Join aircraft info via composite key on vehicle fields	Make, Model, SerialNumber, RegistrationNumber, EventDate	25.0% (5,003 / 20,000)
Conflict Resolution	Resolve conflicting engine counts via 3 rules	Fill, replace zero, overwrite mismatch	3,090 resolved
Airline Temporal Fusion	Link airline data to accident month	Month-level temporal fusion	Monthly aggregate
Deduplication	Delete duplicated records after matching	Delete equal records	3 deletions
Slot Filling	Fill missing values	When matching 2 datasets, if one has a missing value, use the other one	7003 filled
Conflict Resolution	Resolve conflicting airport names	Choose the name with higher length	5,106 resolved

relation heatmaps and distribution plots, enhancing exploratory data analysis and pattern recognition.

TQDM was employed to provide dynamic progress bars for long-running loops and batch operations. This helped monitor and debug data processing stages more effectively.

Statsmodels supported advanced statistical analysis, including regression modeling and statistical testing, enabling a deeper understanding of relationships between key variables.

IPython.display was used to render HTML and improve the visual layout of outputs directly within the notebook environment, especially when formatting tables or embedding interactive elements.

Pathlib and os were used for file system operations and dynamic file handling. This was critical to automate the processing of new data files placed in the `datasource/` directory.

The calendar and re (regular expressions) modules were utilized for text parsing and temporal preprocessing, allowing for efficient date formatting, filtering, and extraction of structured patterns from raw text.

XI. Conclusion

This project gave us a meaningful look into how tourism, weather, and aircraft-related factors may influence aviation accident patterns in the U.S. from 2003 to 2023. By integrating multiple real-world datasets, we were able to apply data fusion and identity resolution techniques to uncover connections that wouldn't be obvious from any single source.

We found a moderate correlation between higher passenger volumes and increased accident counts. Weather had a noticeable impact too—conditions like low temperatures, high winds, and heavy precipitation were linked to higher accident rates.

We also looked into how aircraft type and production year might relate to accident rates. The identity resolution process helped a lot, especially once we switched to the ICAO aircraft dataset, though there's still room to improve. Investigating engine types is another promising area for future work.

In the end, the project showed how combining diverse datasets, when done carefully, can lead to useful and interesting insights. There's more to explore, but this was a solid step toward understanding how different factors affect aviation safety.

References

[1] Alvaroibrain. *Aircraft Production Data*. Kaggle Dataset, Accessed April 5, 2025. 2023. URL: https://www.kaggle.com/datasets/alvaroibrain/aircraft-production-data (visited on 04/05/2025).

[2] Charles R. Harris, K. Jarrod Millman, et al. *NumPy: The fundamental package for scientific computing with Python*. 2020. URL: https://numpy.org/.

[3] John D. Hunter, Thomas A. Caswell, et al. *Matplotlib: Visualization with Python*. Version 3.7. 2023. URL: https://matplotlib.org/.

[4] Ragan-Kelley Kluyver et al. *Jupyter Notebook*. 2016. URL: https://jupyter.org/.

[5] National Transportation Safety Board. *National Transportation Safety Board (NTSB)*. Accessed April 5, 2025. 2025. URL: https://www.ntsb.gov/ (visited on 04/05/2025).

[6] Open-Meteo. *Open-Meteo Historical Weather API Documentation*. Accessed April 5, 2025. 2025. URL: https://open-meteo.com/en/docs/historical-weather-api? (visited on 04/05/2025).

[7] The pandas development team. *pandas: Powerful Python data analysis toolkit*. Version 2.0. 2023. URL: https://pandas.pydata.org.

[8] Version 0.4.2. 2016. URL: https://anhaidgroup.github.io/py_stringmatching/v0.4.2/index.html.

[9] Yyxian. *U.S. Airline Traffic Data*. Kaggle Dataset, Accessed April 5, 2025. 2023. URL: https://www.kaggle.com/datasets/yyxian/u-s-airline-traffic-data/ (visited on 04/05/2025).

Appendix

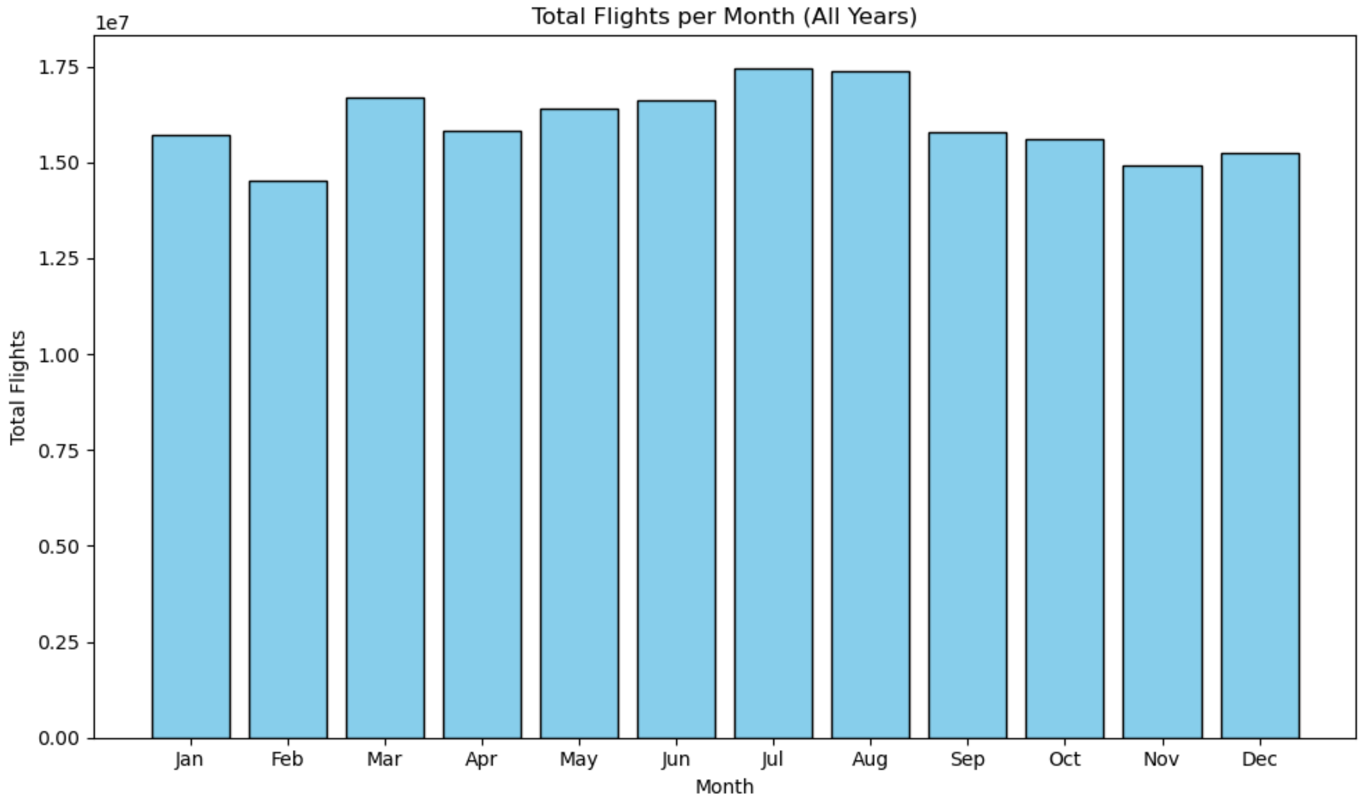

Fig. 1: Histogram comparing the number of flights per month throughout all years

```python
df_filtered = df_aircraft[(df_aircraft['startDate'] < 1000) | (df_aircraft['endDate'] < 1000)]
df_filtered.style.map(
    lambda val: 'background-color: red' if val < 1000 else '',
    subset=['startDate', 'endDate']
)
```

	aircraft	nbBuilt	startDate	endDate
82	lockheed c-5 galaxy	131	5	5
86	british aerospace nimrod aew3	8	11	11
171	schneider es-57 kingfisher	11	2	
190	bell 222	230	222	1991
284	flitfire	49	10	10
308	grumman c-2 greyhound	58	2	2
498	chu hummingbird	2	2	2
514	embraer legacy 500	500	500	
518	lockheed martin f-22 raptor	195	22	22
536	gallaudet d-4	2	2	2
637	fleet canuck	225	198	198
668	bell ah-1 supercobra	1271	1	1
688	chu cjc-3	1	1	1
896	dallach sunrise	0	5	5
951	sukhoi su-30mki	200	30	30
1049	boeing kb-29 superfortress	282	92	
1089	myasishchev m-4	2	93	93
1200	yakovlev yak-100	2	115	115

Fig. 2: Aircraft with wrong start and end dates

```python
df_ntsb.loc[df_ntsb['NtsbNumber']=='ops24la011']
```

Python

Oid	MKey	HighestInjury	NtsbNumber	ProbableCause	City	Country	EventDate	State	Agency	EventType	AirportId	AirportName	Latitude	Lor
67ee2dab017de3d12ee03758	193529	NaN	ops24la011	None	north las vegas	usa	2023-12-09 13:06:00	nv	ntsb	occ	vgt	north las vegas	36.211268	-115
67ee2dab017de3d12ee03758	193529	NaN	ops24la011	None	north las vegas	usa	2023-12-09 13:06:00	nv	ntsb	occ	vgt	north las vegas	36.211268	-115

Fig. 3: Duplicate entries

	Column	DataType	TotalCount	NonNullCount	NumMissing	MissingPerc	Cardinality
0	Vehicles.VehicleNumber	int64	23403	23403	0	0.00	3
1	Vehicles.DamageLevel	category	23403	23400	3	0.01	6
2	Vehicles.ExplosionType	category	23403	21880	1523	6.51	6
3	Vehicles.FireType	category	23403	23321	82	0.35	7
4	Vehicles.SerialNumber	object	23403	23283	120	0.51	21514
5	Vehicles.Make	object	23403	23402	1	0.00	1098
6	Vehicles.Model	object	23403	23398	5	0.02	3362
7	Vehicles.NumberOfEngines	int64	23403	23403	0	0.00	5
8	Vehicles.RegistrationNumber	object	23403	23397	6	0.03	22386
9	Vehicles.FlightOperationType	object	23403	21593	1810	7.73	22
10	Vehicles.OperatorName	object	23403	11290	12113	51.76	9289
11	Oid	object	23403	23403	0	0.00	22992
12	MKey	int64	23403	23403	0	0.00	22992
13	HighestInjury	category	23403	23307	96	0.41	5
14	NtsbNumber	object	23403	23403	0	0.00	22992
15	ProbableCause	object	23403	23205	198	0.85	20890
16	City	object	23403	23403	0	0.00	6092
17	Country	object	23403	23403	0	0.00	1
18	EventDate	datetime64[ns]	23403	23403	0	0.00	22717
19	State	object	23403	23356	47	0.20	58
20	Agency	object	23403	22495	908	3.88	3
21	EventType	category	23403	23403	0	0.00	3
22	AirportId	object	23403	17179	6224	26.59	5359
23	AirportName	object	23403	17208	6195	26.47	8774
24	Latitude	float64	23403	23107	296	1.26	17297
25	Longitude	float64	23403	23106	297	1.27	18108
26	TotalInjuryCount	int64	23403	23403	0	0.00	33

Fig. 4: Part of NTSB Data Profile

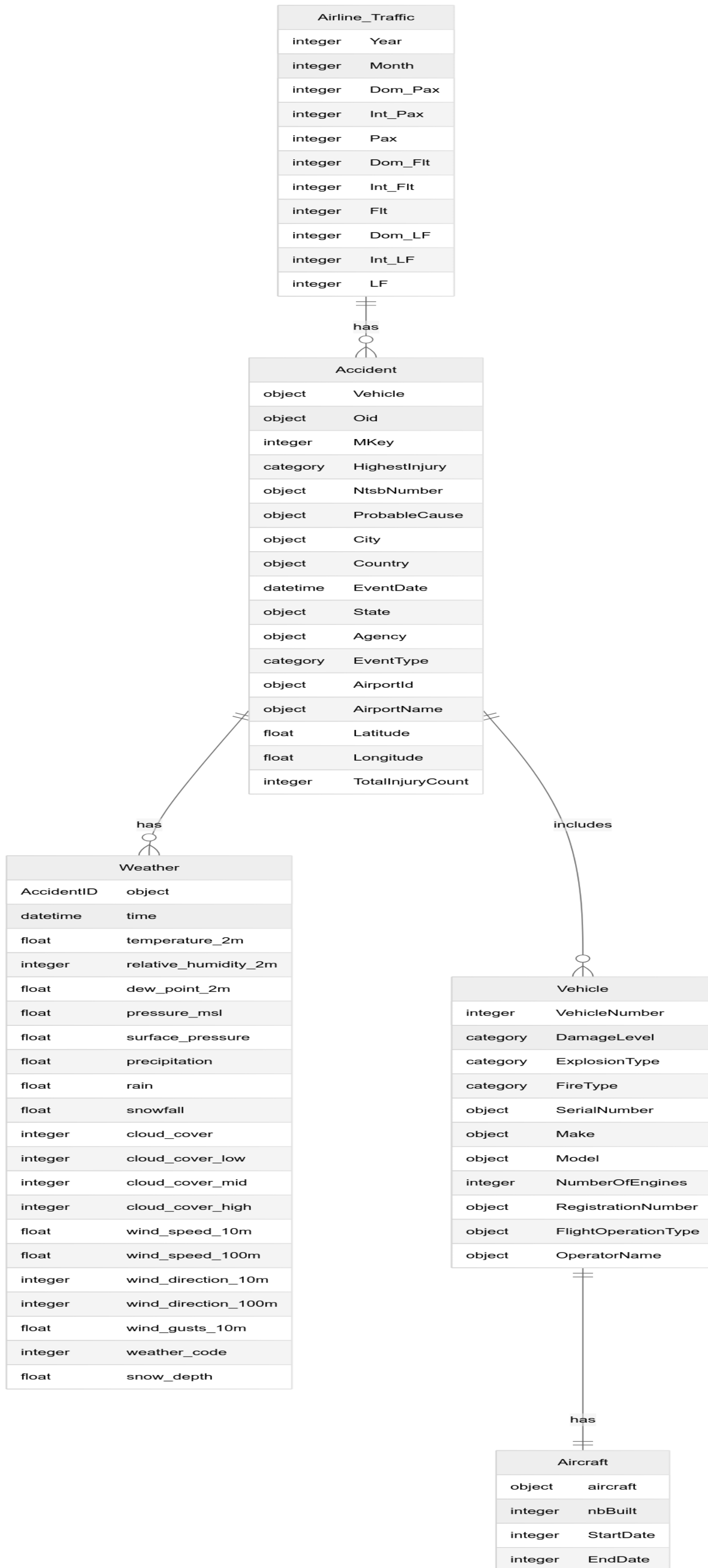

Fig. 5: Conceptual models

NTSB Aviation Accident Database model			
Attribute	Type	Constraints	Relevant
Vehicles.VehicleNumber	int64	>0	no
Vehicles.DamageLevel	category	only four possible values	no
Vehicles.ExplosionType	category	only two possible values	no
Vehicles.FireType	category	only two possible values	no
Vehicles.SerialNumber	object	none	no
Vehicles.Make	object	none	yes
Vehicles.Model	object	none	yes
Vehicles.NumberOfEngines	int64	>0	no
Vehicles.RegistrationNumber	object	none	no
Vehicles.FlightOperationType	object	none	no
Vehicles.OperatorName	object	none	no
Oid	object	none	no
MKey	int64	none	no
HighestInjury	category	only four possible values	yes
NtsbNumber	object	none	yes
ProbableCause	object	none	no
City	object	none	yes
Country	object	none	no
EventDate	datetime64[ns]	date time	yes
State	object	none	yes
Agency	object	none	no
EventType	category	only three possible values	no
AirportId	object	none	no
AirportName	object	none	yes
Latitude	float64	-90 to +90	yes
Longitude	float64	-180 to +180	yes
TotalInjuryCount	int64	none	yes

TABLE II: NTSB Aviation Accident Database model

TABLE VIII: MNLogit Regression Results

Variabile	**coef**	**std err**	**z**	**P > $\|z\|$**	**[0.025,**	**0.975]**
y = 1						
const	−2.6564	0.237	−11.208	0.000	−3.121	−2.192
precipitation	0.1358	0.053	2.567	0.010	0.032	0.239
temperature_2m	−0.0190	0.005	−4.093	0.000	−0.028	−0.010
wind_gusts_10m	0.0268	0.003	9.280	0.000	0.021	0.032
cloud_cover	0.0054	0.001	5.891	0.000	0.004	0.007
Season_Spring	0.1163	0.103	1.134	0.257	−0.085	0.317
Season_Summer	0.1072	0.108	0.994	0.320	−0.104	0.319
Season_Winter	−0.0310	0.121	−0.255	0.798	−0.269	0.207
TimeOfDay_Afternoon	−0.3618	0.204	−1.774	0.076	−0.762	0.038

TABLE IX –

Variabile	coef	std err	z	P > $\|z\|$	[0.025,	0.975]
TimeOfDay_Evening	−0.4793	0.213	−2.254	0.024	−0.896	−0.063
TimeOfDay_Morning	−0.2559	0.211	−1.211	0.226	−0.670	0.158
Region_Central	0.3120	0.096	3.255	0.001	0.124	0.500
Region_South	0.1505	0.093	1.609	0.108	−0.033	0.334
y = 2						
const	−2.2043	0.224	−9.820	0.000	−2.644	−1.764
precipitation	−0.0342	0.082	−0.419	0.675	−0.194	0.126
temperature_2m	0.0180	0.004	4.085	0.000	0.009	0.027
wind_gusts_10m	0.0073	0.003	2.418	0.016	0.001	0.013
cloud_cover	−0.0005	0.001	−0.569	0.570	−0.002	0.001
Season_Spring	−0.1678	0.092	−1.819	0.069	−0.349	0.013
Season_Summer	−0.1780	0.089	−2.007	0.045	−0.352	−0.004
Season_Winter	0.0467	0.111	0.422	0.673	−0.170	0.264
TimeOfDay_Afternoon	−0.1996	0.197	−1.015	0.310	−0.585	0.186
TimeOfDay_Evening	−0.1845	0.203	−0.910	0.363	−0.582	0.213
TimeOfDay_Morning	−0.2744	0.204	−1.344	0.179	−0.674	0.126
Region_Central	0.2089	0.095	2.198	0.028	0.023	0.395
Region_South	0.3840	0.082	4.699	0.000	0.224	0.544
y = 3						
const	−1.3105	0.169	−7.770	0.000	−1.641	−0.980
precipitation	−0.0425	0.055	−0.780	0.436	−0.149	0.064
temperature_2m	0.0022	0.003	0.832	0.406	−0.003	0.007
wind_gusts_10m	0.0046	0.002	2.431	0.015	0.001	0.008
cloud_cover	−0.0011	0.001	−2.272	0.023	−0.002	−0.000
Season_Spring	0.2551	0.057	4.500	0.000	0.144	0.366
Season_Summer	0.0262	0.058	0.455	0.649	−0.087	0.139
Season_Winter	0.1327	0.070	1.884	0.059	−0.005	0.271
TimeOfDay_Afternoon	0.4689	0.155	3.019	0.003	0.164	0.773
TimeOfDay_Evening	0.2375	0.160	1.487	0.137	−0.075	0.550
TimeOfDay_Morning	0.4981	0.158	3.147	0.002	0.188	0.808
Region_Central	0.1290	0.056	2.295	0.022	0.019	0.239
Region_South	0.1291	0.050	2.568	0.010	0.031	0.228
y = 4						
const	−0.7793	0.138	−5.647	0.000	−1.050	−0.509
precipitation	−0.0352	0.048	−0.727	0.467	−0.130	0.060
temperature_2m	0.0142	0.003	5.561	0.000	0.009	0.019
wind_gusts_10m	0.0037	0.002	2.088	0.037	0.000	0.007
cloud_cover	−0.00009	0.000	−0.195	0.846	−0.001	0.001
Season_Spring	−0.0136	0.054	−0.250	0.802	−0.120	0.093

TABLE IX –

Variabile	**coef**	**std err**	**z**	**P > $\|z\|$**	**[0.025,**	**0.975]**
Season_Summer	−0.0819	0.053	−1.548	0.122	−0.186	0.022
Season_Winter	0.2034	0.065	3.127	0.002	0.076	0.331
TimeOfDay_Afternoon	−0.0883	0.123	−0.717	0.474	−0.330	0.153
TimeOfDay_Evening	−0.1193	0.127	−0.939	0.348	−0.368	0.130
TimeOfDay_Morning	−0.0250	0.127	−0.198	0.843	−0.273	0.223
Region_Central	0.2039	0.053	3.835	0.000	0.100	0.308
Region_South	0.2018	0.047	4.264	0.000	0.109	0.295

Target class mapping: 0: Collision / Obstacle; 1: Environmental Conditions; 2: Technical Failure; 3: Human Error; 4: Loss of Situational Awareness.

U.S. Airline Traffic Data model			
Attribute	Type	Constraints	Relevant
Year	int32	between 2003-2023	no
Month	int32	between 1-12	no
Dom_Pax	int32	>=0	no
Int_Pax	int32	>=0	no
Pax	int32	>=0	yes
Dom_Flt	int32	>=0	no
Int_Flt	int32	>=0	no
Flt	int32	>=0	yes
Dom_LF	int32	>=0	no
Int_LF	int32	>=0	no
LF	int32	>=0	yes

TABLE III: U.S. Airline Traffic Data model

Open-Meteo API model			
Attribute	Type	Constraints	Relevant
AccidentID	object	none	no
time	datetime64[ns]	datetime	no
temperature_2m	float64	none	yes
relative_humidity_2m	int32	none	yes
dew_point_2m	float64	none	yes
pressure_msl	float64	none	yes
surface_pressure	float64	none	yes
precipitation	float64	none	yes
rain	float64	none	yes
snowfall	float64	none	yes
cloud_cover	int64	none	yes
cloud_cover_low	int32	none	yes
cloud_cover_mid	int32	none	yes
cloud_cover_high	int32	none	yes
wind_speed_10m	float64	none	yes
wind_speed_100m	float64	none	yes
wind_direction_10m	int32	none	yes
wind_direction_100m	int32	none	yes
wind_gusts_10m	float64	none	yes
weather_code	int32	none	yes
snow_depth	float64	none	yes

TABLE IV: Open-Meteo API model

Aircraft Database model			
Attribute	Type	Constraints	Relevant
aircraft	object	none	yes
nbBuilt	int64	>0	no
startDate	int64	datetime	yes
endDate	Int64	datetime	yes

TABLE V: Aircraft Database model

Origin Dataset	**From**	**Target**	**Type Corresp.**	**Description**
NTSB	NtsbNumber	AccidentNumber	1-1	
NTSB	EventDate	DateTime	1-1	
NTSB	City	City	1-1	
NTSB	State	State	1-1	
NTSB	Longitude	Longitude	1-1	
NTSB	Latitude	Latitude	1-1	
NTSB	AirportName	AirportName	1-1	
NTSB	Operator	Operator	1-1	
NTSB	Aircraft Damage	Aircraft Damage	1-1	
NTSB	FatalInjuryCount; SeriousInjuryCount; MinorInjuryCount	TotalInjuryCount	N-1	Sum all the value
NTSB	HighestInjury	HighestInjury	1-1	
NTSB	Model, Make	Aircraft	N-1	Blocking with Aircraft Data
Aircraft Data	StartDate	ProductionStartDate	1-1	
Aircraft Data	EndDate	ProductionEndDate	1-1	
Weather API	Temperature_2m	Temperature	1-1	
Weather API	Precipitation	Precipitation	1-1	
Weather API	Wind_Speed_10m	WindSpeed	1-1	
Weather API	Weather code	Weather code	1-1	
Weather API	other weather info	other weather info	1-1	
Airline Traffic	Pax	PassengersPerMonth	1-1	
Airline Traffic	Flt	FlightsPerMonth	1-1	
Airline Traffic	LF	LoadFactorPerMonth	1-1	

TABLE VI: Integrated Model

Origin Dataset	**From**	**Target**	**Type Corresp.**	**Description**
NTSB	Vehicles.DamageLevel	Vehicles.DamageLevel	1-1	
NTSB	Vehicles.ExplosionType	Vehicles.ExplosionType	1-1	
NTSB	Vehicles.FireType	Vehicles.FireType	1-1	
NTSB	Vehicles.SerialNumber	Vehicles.SerialNumber	1-1	
NTSB	Vehicles.Make	Vehicles.Make	1-1	
NTSB	Vehicles.Model	Vehicles.Model	1-1	
NTSB	Vehicles.NumberOfEngines	Vehicles.NumberOfEngines	1-1	
NTSB	Vehicles.RegistrationNumber	Vehicles.RegistrationNumber	1-1	
NTSB	Vehicles.FlightOperationType	Vehicles.FlightOperationType	1-1	
NTSB	Vehicles.OperatorName	Vehicles.OperatorName	1-1	
NTSB	Oid	Oid	1-1	
NTSB	MKey	MKey	1-1	
NTSB	HighestInjury	HighestInjury	1-1	
NTSB	NtsbNumber	NtsbNumber	1-1	
NTSB	ProbableCause	ProbableCause	1-1	
NTSB	City	City	1-1	
NTSB	Country	Country	1-1	
NTSB	EventDate	EventDate	1-1	
NTSB	State	State	1-1	
NTSB	Agency	Agency	1-1	
NTSB	AirportId	AirportId	1-1	
NTSB	AirportName	AirportName	1-1	
NTSB	Latitude	Latitude	1-1	
NTSB	Longitude	Longitude	1-1	
NTSB	FatalInjuryCount; SeriousInjuryCount; MinorInjuryCount	TotalInjuryCount	N-1	Sum all the value
Weather API	weather_time	weather_time	1-1	
Weather API	temperature_2m	temperature_2m	1-1	
Weather API	relative_humidity_2m	relative_humidity_2m	1-1	
Weather API	dew_point_2m	dew_point_2m	1-1	
Weather API	pressure_msl	pressure_msl	1-1	
Weather API	surface_pressure	surface_pressure	1-1	
Weather API	precipitation	precipitation	1-1	
Weather API	other weather info	other weather info	1-1	
Airline Traffic	Pax	TotalPassengers	1-1	
Airline Traffic	Flt	TotalFlights	1-1	
Airline Traffic	LF	LoadFactor	1-1	
Aircraft Data	engine_type	engine_type	1-1	
Calculated from NTSB and Weather API	EventDate - weather_time	time_diff	1-1	

TABLE VII: Final Integrated Model

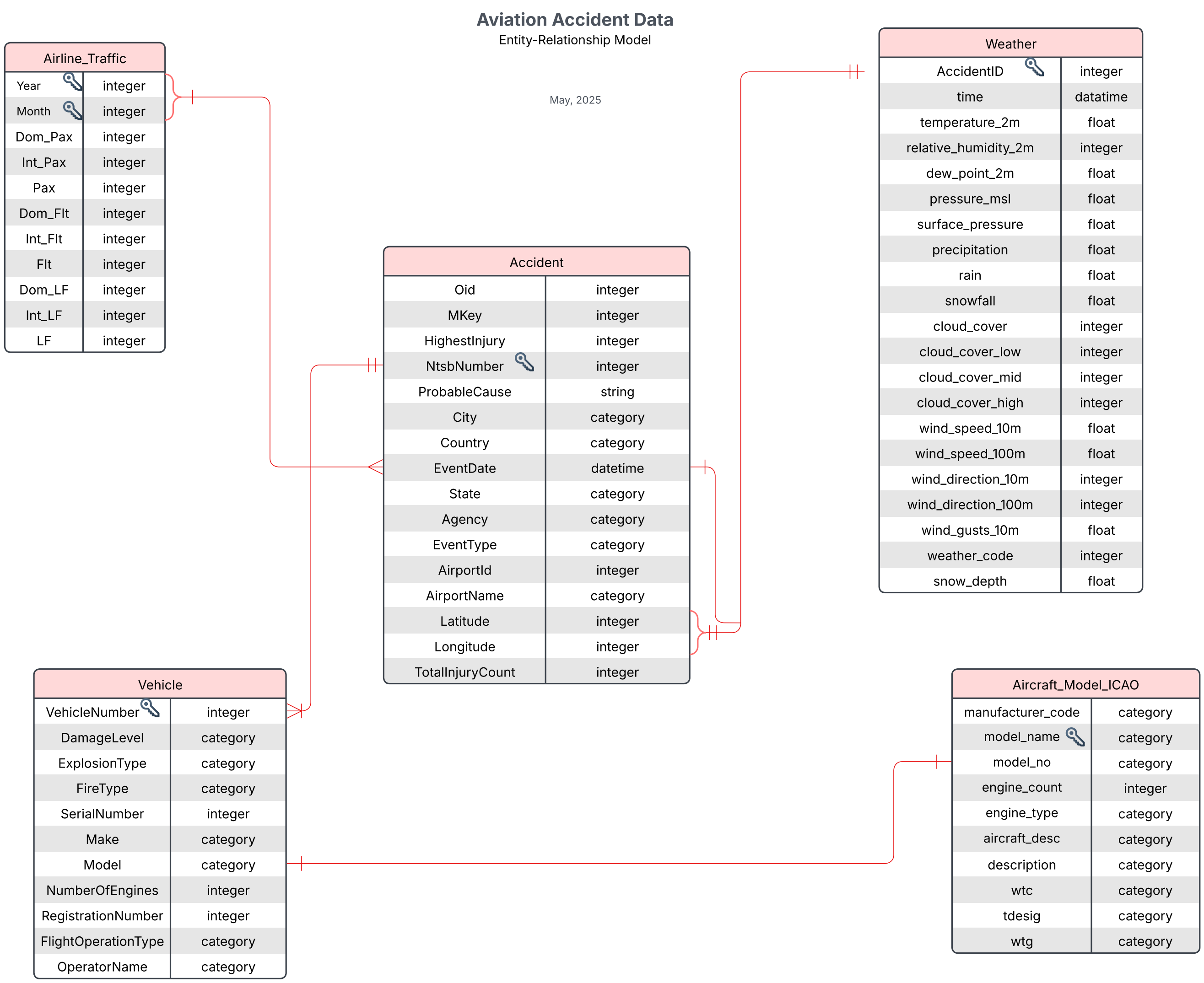

Fig. 6: Entity-Relationship Model