International Airport ER Diagram — Report

Introduction

This mini-report explains what I actually did to design the ER diagram of the "International Airport" system in DrawSQL. The goal was to capture the real-world objects we care about (airports, airlines, flights, passengers and their bookings) and show how they relate to each other in a clean, readable model that can be implemented in a database through website drawSQL.

Tasks Completed

- Planned the model: listed the core entities and clarified what each one stores.
- Built the diagram in DrawSQL: created tables for airports, airlines, flights, passengers, bookings, and supporting entities (booking_changes, boarding_passes, baggage, baggage_checks, security_checks).
- Defined keys and constraints: marked Primary Keys, set Foreign Keys, and added unique business identifiers (airline_code, passport_number).
- Modeled relationships: one airport can serve many flights (departure/arrival), one passenger can have many bookings, and bookings link out to their artifacts (boarding passes, baggage and checks).
- Added indexes on foreign keys so joins are fast and referential checks are efficient; made airline_code and passport_number unique.
- Exported the finished diagram to PDF/PNG and prepared a small legend so the icons are clear during the presentation.

Conclusion

The resulting schema is compact, normalized and covers the typical questions we need to ask: which flights depart from a given airport, what a passenger has booked, and how baggage and security checks are recorded. The model is easy to extend—e.g., by adding flight legs or extra reference tables—without breaking existing data.