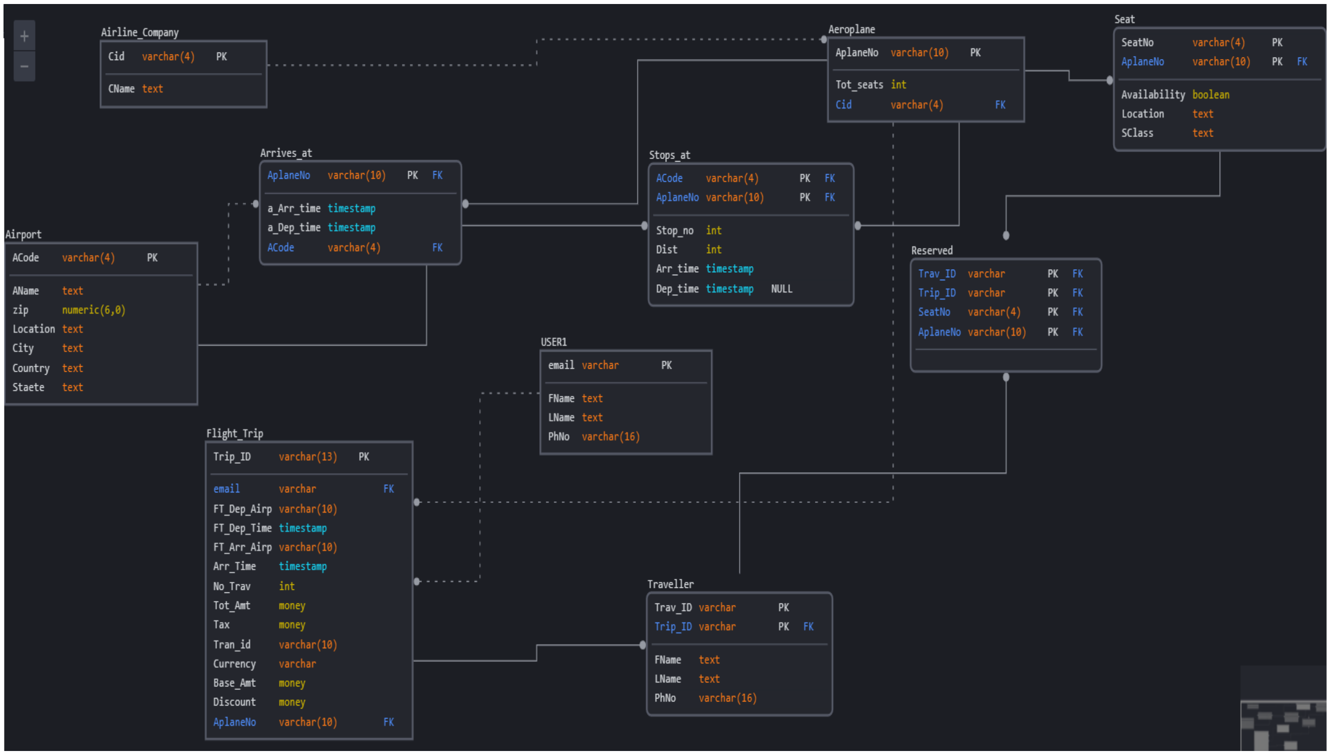
DATABASE MANAGEMENT SYSTEM

AIRLINES RESERVATION SYSTEM

TEAM ID: 05

RITHIKA PAI	RITHIKA SHANKAR	RAMYA N PRABHU
PES1UG19CS388	PES1UG19CS387	PES1UG19CS380

ER to Schema



```
Airline_Company
Cid     varchar(4)     PK
CName   text

Aeroplane
AplaneNo   varchar(10)   PK
Tot_seats  int
Cid        varchar(4)         FK

Seat
SeatNo        varchar(4)    PK
AplaneNo      varchar(10)   PK   FK
Availability  boolean
Location      text
SClass        text

Arrives_at
AplaneNo    varchar(10)   PK   FK
a_Arr_time  timestamp
a_Dep_time  timestamp
ACode       varchar(4)         FK

Stops_at
ACode     varchar(4)     PK   FK
AplaneNo  varchar(10)    PK   FK
Stop_no   int
Dist      int
Arr_time  timestamp
Dep_time  timestamp   NULL

Airport
ACode     varchar(4)    PK
AName     text
zip       numeric(6,0)
Location  text
City      text
Country   text
Staete    text

Reserved
Trav_ID   varchar       PK   FK
Trip_ID   varchar       PK   FK
SeatNo    varchar(4)    PK   FK
AplaneNo  varchar(10)   PK   FK

USER1
email  varchar      PK
FName  text
LName  text
PhNo   varchar(16)

Flight_Trip
Trip_ID      varchar(13)   PK
email        varchar            FK
FT_Dep_Airp  varchar(10)
FT_Dep_Time  timestamp
FT_Arr_Airp  varchar(10)
Arr_Time     timestamp
No_Trav      int
Tot_Amt      money
Tax          money
Tran_id      varchar(10)
Currency     varchar
Base_Amt     money
Discount     money
AplaneNo     varchar(10)        FK

Traveller
Trav_ID  varchar       PK
Trip_ID  varchar       PK   FK
FName    text
LName    text
PhNo     varchar(16)
```

Choice of DBMS

Why RDBMS?

- Relational database normalization means that each fact in your data is stored only once, so you shouldn't get data anomalies.
- The data we wish to capture and store is structured and is best stored with enforcement of declarative constraints, such as NOT NULL, UNIQUE KEY, or FOREIGN KEY
- SQL is the dominant query language for relational databases. SQL is an industry standard, interoperable between platforms and application programming languages, well-documented, and stable.
- Relational schema means you always know what columns (i.e. attributes) exist for a given row so it's easier to write queries, find the required record or field.
- Data types help to ensure data in a given column is well-formed. CHECK constraints can further validate data (if your RDBMS supports CHECK constraints).
- Authentication and access privileges are handled better in SQL. Security is still pretty primitive in many NoSQL products (this varies; some are more capable than others).
- SQL is the dominant query language for relational databases. SQL is an industry standard, interoperable between platforms and application programming languages, well-documented, and stable.

- Many RDBMS exist that have solved the ACID requirement with a good balance between durability and performance. The performance of NoSQL databases tends to plummet if you configure them to have durability matching that of an RDBMS.
- You have different types of tabular data that you want to query and manage using dynamic relationships between the data (also known as relational joins).
- Relational databases also have constant exposure to numerous workloads, and many more hours running in production. Edge cases are more often handled well in mature projects, and you have numerous patterns to follow for backups, for change management and overall operational rigor.

- Ensuring strict transaction semantics, This allows you to robustly build financial application with strict data consistency guarantees
- Schema discovery. Just by looking at the tables and columns you can infer a lot of information
- Incredible expressibility of the SQL language to generate analytical reports. There isn't really a better API for analytical queries
- Mature and well developed query optimization techniques. It applies even more so to analytical workloads: transforming subqueries into joins, index selection, bushy joins, etc
- A Relational Database system is the most simple model, as it does not require any complex structuring or querying processes.
- In the relational database system, there can be multiple tables related to one another with the use of a primary key and foreign key concepts. This makes the data non-repetitive. There is no chance for duplication of data. Hence the accuracy of data in the relational database is more than any other database system.
- In the Relational Database System, there is no pattern or pathway for accessing the data, as to another type of databases can be accessed only by navigating through a tree or a hierarchical model. Anyone who accesses the data can query any table in the relational database.
- Data integrity is a crucial characteristic of the Relational Database system. Sturdy Data entries and legitimacy validations ensure that all the Data in the database confines within suitable arrangements and the data necessary for creating the relationships are present.
- As the relational database system holds records in separate tables based on their categories, it is straightforward to insert, delete or update records that are subjected to the latest requirements.
- As the data is divided amongst the tables of the relational database system, it is possible to make a few tables to be tagged as confidential and others not.

Why PostgreSQL?

- PostgreSQL is popular with developers due to numerous features such as: native partitioning, parallel query, support of foreign data wrappers, powerful JSON features, streaming and logical replication and the availability of many open source tools for HA, backups and monitoring.
- It has good community support and like Python is cross-platform . It supports commonly used Python ORMs such as SQLAlchemy, Django ORM,etc
- Postgres regularly updates itself with new features.
- Multiple technical options for operating PostgreSQL at scale
- PostgreSQL is fully ACID compliant and therefore ideal for OLTP (Online Transaction Processing) workloads.

Why PostgreSQL over MySQL?

- Postgres is an object-relational database, while MySQL is a purely relational database. This means that Postgres includes features like table inheritance and function overloading, which can be important to certain applications. Postgres also adheres more closely to SQL standards.
- Postgres handles concurrency better than MySQL for multiple reasons:
    - Postgres implements Multiversion Concurrency Control (MVCC) without read locks Postgres supports parallel query plans that can use multiple CPUs/cores Postgres can create indexes in a non-blocking way (through the CREATE INDEX CONCURRENTLY syntax), and it can create partial indexes (for example, if you have a model with soft deletes, you can create an index that ignores records marked as deleted) Postgres is known for protecting data

integrity at the transaction level. This makes it less vulnerable to data corruption.

- Runs on all major OS platforms that you may have
- PostgreSQL is largely SQL compliant.
- It also offers table inheritance, rules systems, custom data types, and database events

Contributions

Rithika Pai

- Insert File

2 hours

Rithika Shankar

- Insert file
- Report

4 hours

Ramya Prabhu

- Create File
- Report

4 hours