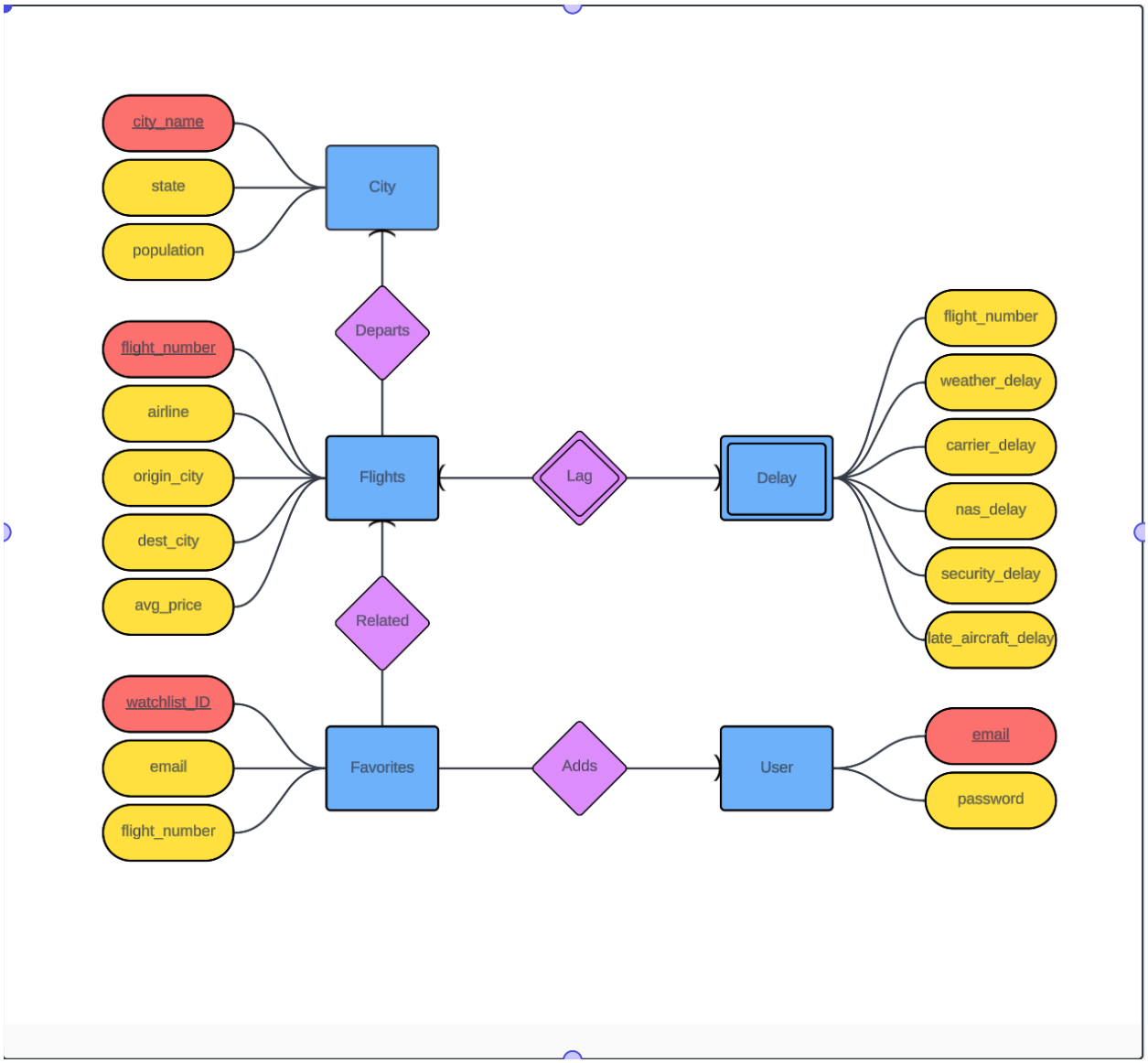
ER Diagram


city_name
state
population
City
Departs
flight_number
airline
origin_city
dest_city
avg_price
Flights
Lag
Delay
flight_number
weather_delay
carrier_delay
nas_delay
security_delay
late_aircraft_delay
Related
watchlist_ID
email
flight_number
Favorites
Adds
User
email
password

Normalization

- **Minimal Basis:** flight_number->Airline/total_delays, city_name -> origin_city, origin_city/dest_city->avg_price, origin_city/dest_city/airline->flight_number, watchlist_ID->email
- **3NF Decomposition:** (flight_number, airline, total_delays), (city_name, origin_city), (avg_price, origin_city, dest_city), (flight_number, origin_city, dest_city, airline), (watchlist_ID, email), (city_name, watchlist_ID, origin_city/dest_city, origin_city/dest_city/airline)
- We chose to use 3NF over BCNF normalization because the relations between tables were simple enough that only the minimal functional dependencies were required to be kept. We also had a weak entity relation between Flights and Delay so we wanted to preserve that dependency, which is something that BCNF would have been too aggressive in decomposing.

Entity Assumptions

- Flights:
    - Contains details about individual flights.
    - Flight_number is a unique identifier (primary key)
- City:
    - Specifies which city a flight is departing from and landing in
- Delay:
    - Contains the individual sources of delay that aggregate to the total delay time. We chose to create a separate Delay entity to establish better separation between a flight and the unique reasons for its delay. This is a weak entity because each Delay item is uniquely identified by flight_number.
- Favorites:
    - A customized watchlist containing flights that users want to keep an eye on.
- User:
    - Contains user login information

Relationship Assumptions

(* = any number of)

- Flights and City
    - Each flight can only depart from exactly 1 city
    - A city can be the departure point of 0 to * flights
- Flights and Delay

    - Each flight can only have exactly 1 block of delay data
    - Each unique delay block corresponds to exactly 1 flight
- Flights and Favorites
    - Each flight can be on 0 to * favorite lists
    - Each entry on favorites must correspond to exactly 1 flight
- Favorites and User
    - Each user can favorite 0 to * flights
    - Each entry of favorites must correspond to exactly 1 user

Relational Schema

```
Flights(
  flight_number:INT [PK],
  airline:VARCHAR(30),
  origin_city:VARCHAR(50) [FK to City.city_name],
  dest_city:VARCHAR(50) [FK to City.city_name],
  avg_price:Decimal
)

City(city_name:VARCHAR(50) [PK], state:VARCHAR(20), population:INT)

Delay(
  flight_number:INT [FK to Flights.flight_number],
  weather_delay:INT,
  carrier_delay:INT,
  nas_delay:INT,
  security_delay:INT,
  late_aircraft_delay:INT
)

Favorites(
  watchlist_ID:INT [PK],
  email:VARCHAR(50) [FK to User.email],
  flight:INT [FK to Flights.flight_number]
)

User(email:VARCHAR(50) [PK], password:VARCHAR(50))
```