

# **Project 1- Design & Implement a Relational Database**

## **Database & Design**

### **Raneem Belbisi**

## **Business Requirement:**

### **Introduction:**

This project aims to create a bike-sharing system that allows residences and tourists to access bicycles for short-trips across the Bay Area. This is important as it is a sustainable transportation solution, promotes healthy living and exercise, provides accessible and affordable transportation as well as provides valuable data on consumer patterns that can be used for urban planning and traffic management within the area.

### **Business Rules:**

- 1) Each bike has a unique ID and belongs to a specific station.
- 2) Stations have unique identifiers and are located at distinct geographical points.
- 3) Reservations have a start and end time.
- 4) Stations must have name and location.
- 5) Stations must have a maximum number of bikes it can hold.
- 6) Stations have multiple bikes that can be rented and their status is reported.
- 7) Weather is recorded.
- 8) Administrators can view usage analytics reports.

### **Nouns:**

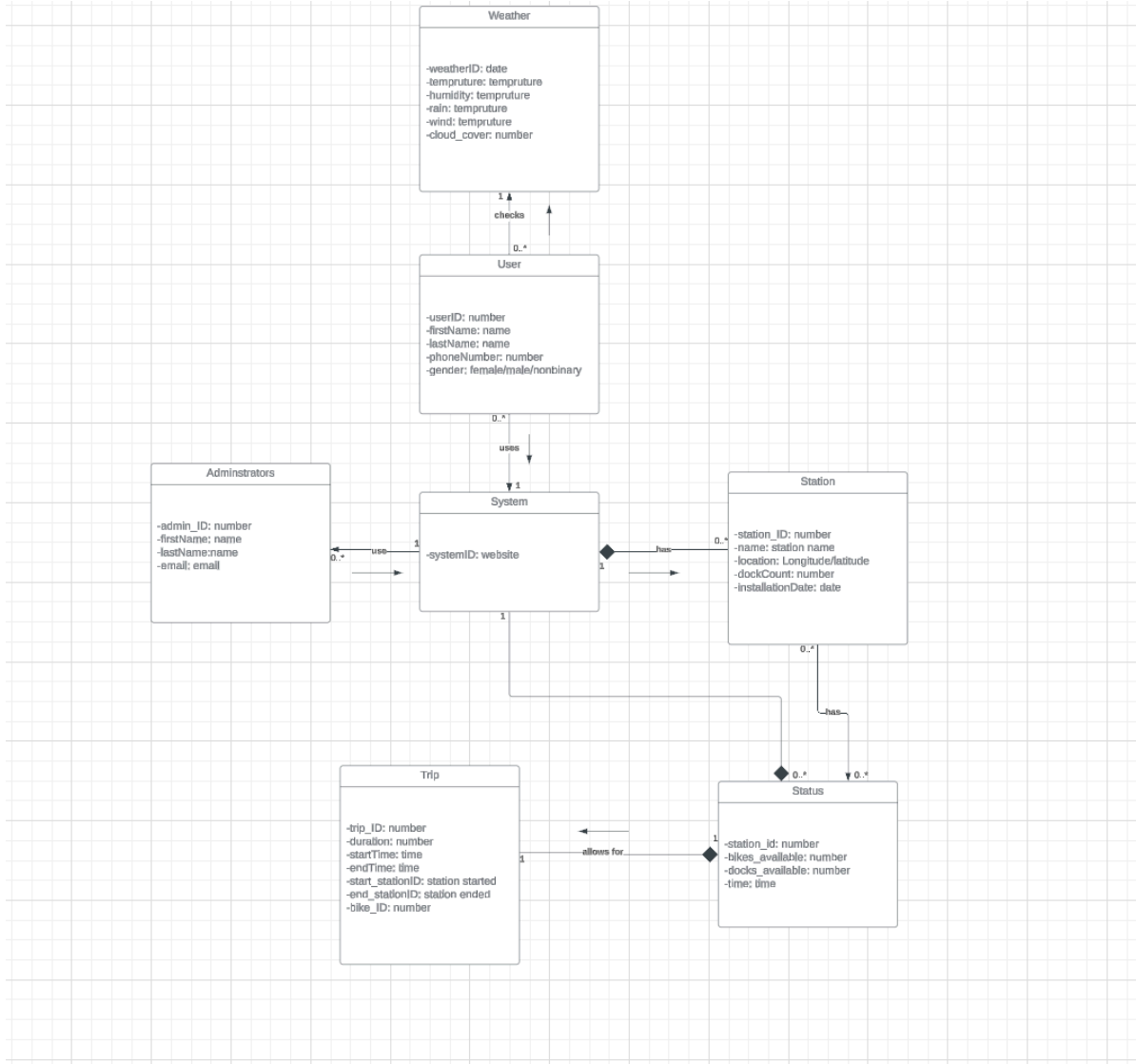
- 1) Users
- 2) Station
- 3) Administrations
- 4) Trips
- 5) Status
- 6) Weather
- 7) System

### **Verbs:**

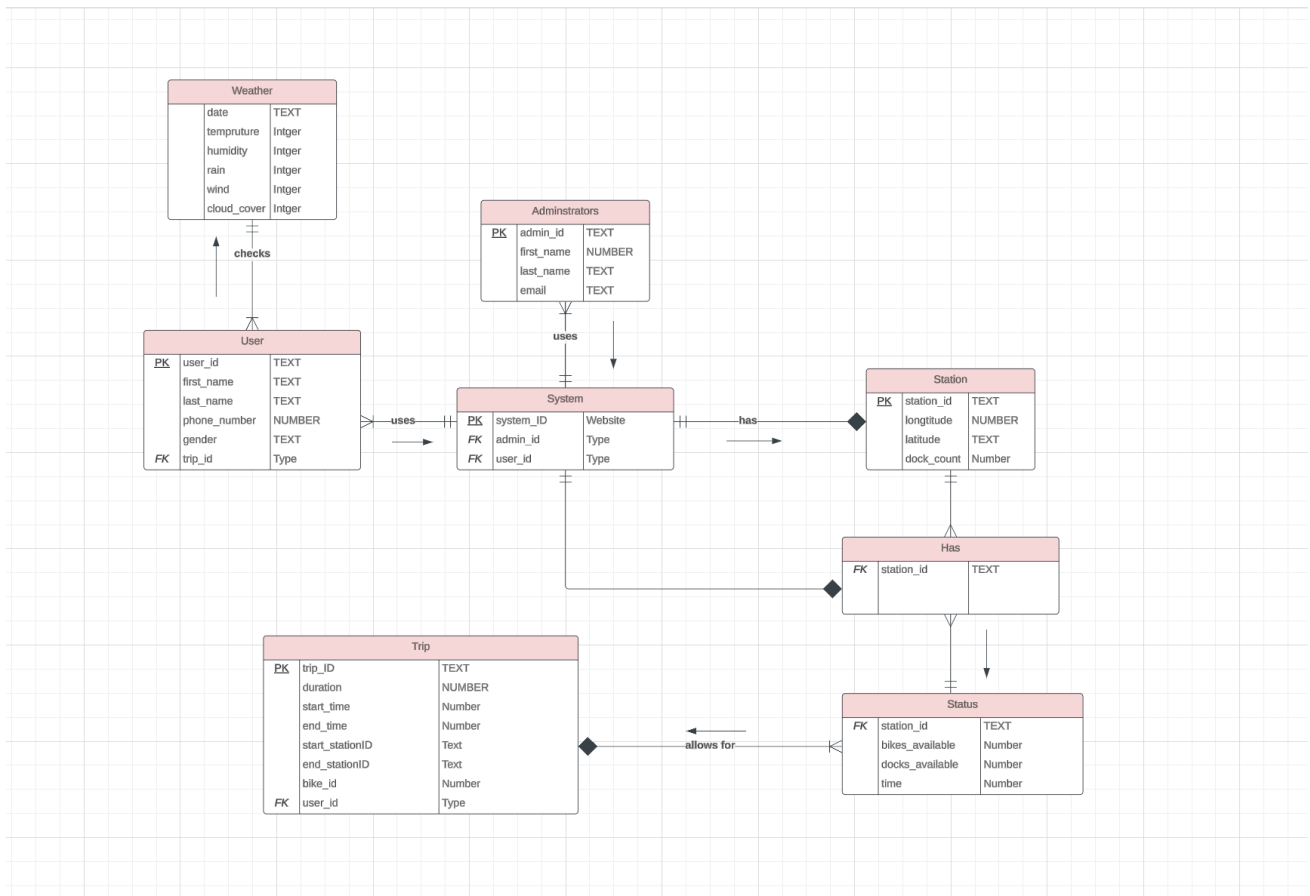
- 1) Register
- 2) Record
- 3) Reserve
- 4) View

For the data collected, I used 4 tables from [SF Bay Area Bikeshare](#) and created two tables 'users' and 'admins' from Mackaroo. I joined them together on db browser, created a zip then uploaded them into observable to complete my project.

## UML Class Diagram:



## ERD Crow's foot:



## Relational Schema:

Weather(date, temperature, humidity, rain, wind, cloud\_cover)

User(user\_id, first\_name, last\_name, phone\_number, gender, trip\_id)

Administrators(admin\_id, first\_name, last\_name, email)

System(system\_id, admin\_id, user\_id)

Station(station\_id, longitude, latitude, dock\_count)

Has(station\_id)

Status(station\_id, bikes\_available, docks\_available, time)

Trip(trip\_id, duration, start\_time, end\_time, start\_stationID, end\_stationID, bike\_id, user\_id)

## Proof that my Relational Schema is a BNF:

- 1) Each table has a superkey or has an attribute that is a superkey.
- 2) A superkey is a set of attributes that can uniquely identify each tuple in the table.

[Observable link](#)

