Database Documentation

**Oracle Database**

**Summary:**

The database is run through an Oracle Autonomous database, and connected to the website through a TLS connection. It is used to store patient data to be displayed on the website, and staff data to manage access.

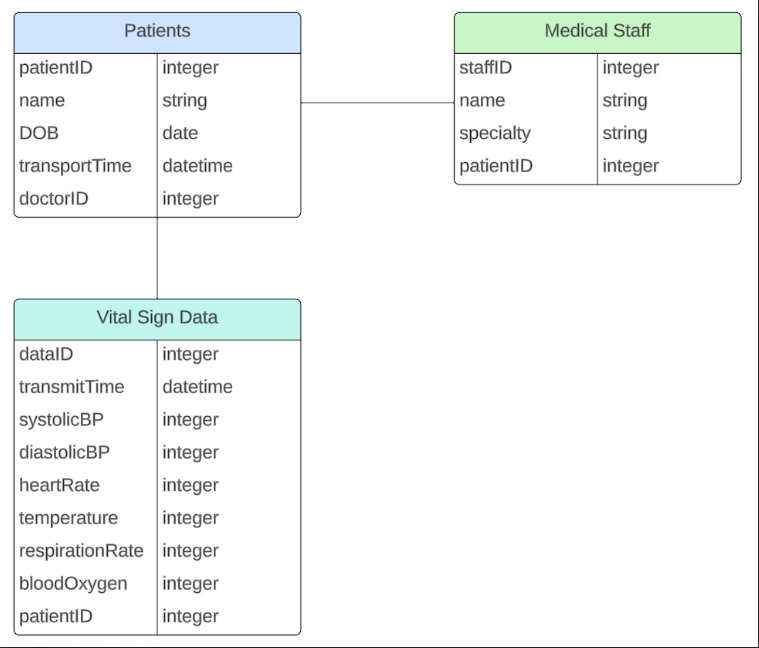
The schema for the database is displayed below:

Table fore VITAL\_SIGN database:

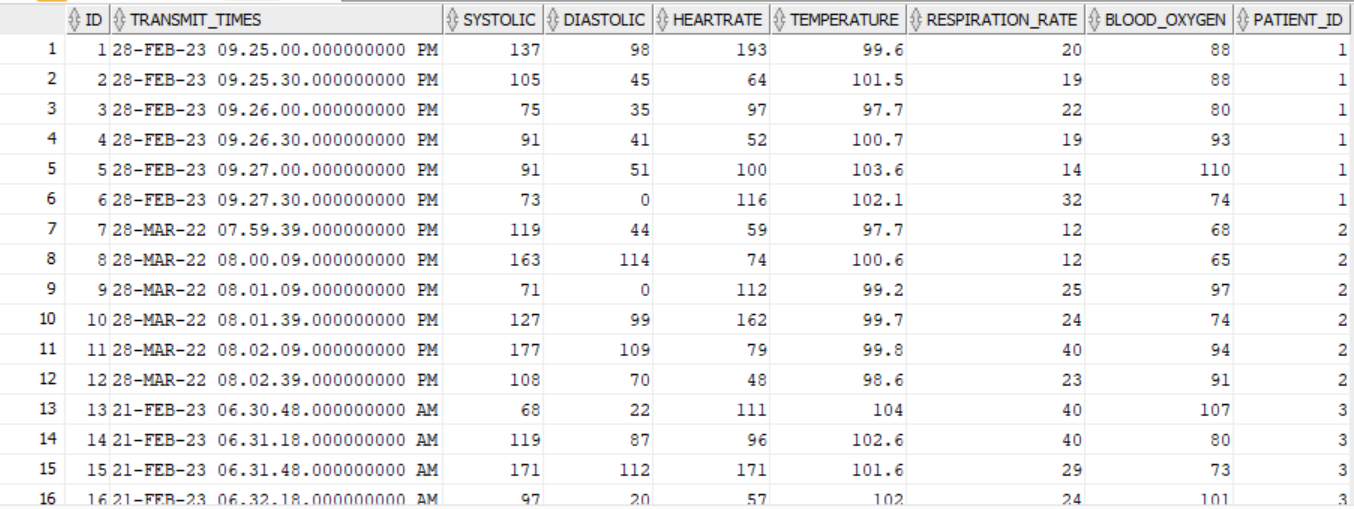
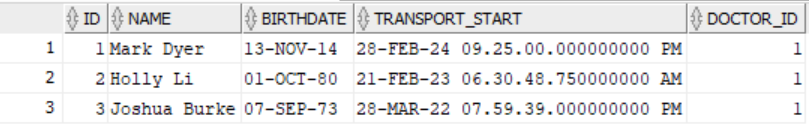


Table for PATIENT database:



**Relevant Files:**

db\_connection.js: This file connects the database to the website.

**Process:**

The Oracle Autonomous database performs database management automatically and does not need outside management from developers. It can be edited through CRUD operations run through the website program, or through direct edits using Oracle SQL developer. Connections between the database and other programs can be easily created through TLS authentication or wallet authentication.

**Team Members:** Kierra Young

**Data Generator**

**Summary:** This Python file generates mock patient, medical staff, and vital sign data and saves it to a CSV file.

Input Parameters:

* outputFile: Specify the name of the CSV file where the data will be saved.
* numPatients: Determine the number of fictional patients for whom data will be generated.
* numStaff: Specify the number of fictional medical staff members to generate data for.

Data Generated:

* Patient Data: Each patient entry includes a unique patient ID, name, date of birth (DOB), transport time, and the ID of the doctor assigned to them.
* Vital Sign Data: Vital signs such as systolic and diastolic blood pressure, heart rate, temperature, respiration rate, and blood oxygen level are simulated for each patient. These readings are generated at various 30-second time intervals following the transport time.
* Medical Staff Data: Each medical staff entry includes a unique staff ID, name, specialty (e.g., EMT, nurse, doctor, radiologist), and a list of patient IDs associated with them.

Execution:

* Install the Faker library before running the Python file (pip install Faker).
* Specify the desired output file name, number of patients, and number of medical staff members.
* Run the script to generate the mock data and save it to the specified CSV file.

Additional Notes:

* The data generated by this script is entirely fictional and should not be used for any real-world medical analysis or decision-making purposes.
* Random values are generated for vital signs within realistic ranges.

**Relevant Files:**

* data\_generator.py: This Python file generates mock patient, medical staff, and vital sign data, which is then saved in a CSV file.
* db\_connection.js: This JavaScript file establishes a connection between the Oracle database and the website.
* patient\_data.csv: This CSV file contains the mock patient, medical staff, and vital sign data generated by the data\_generator.py script.

**Process:**

Users submit the number of patients that they want to create fake data for and the number of fake staff they want available. Based on this information, the program generates a CSV file containing fake data for these patients. The data is based on biological knowledge and contains data for systolic/diastolic blood pressure, oxygen level, temperature, and heart rate.

**Known Bugs and Troubleshooting:**

**Team Members:** Lauren Twist