Descriptions of each entity and assumptions regarding relationships:

In the Agency table, we collect data on the agency ID (**which we assume to be unique, hence the primary key**), name, website, timezone, and language of the agencies that manage the transportation system in Sao Paulo. Agency only has a management-based relationship with the Routes table, which describes each route in terms of its route ID (**which we assume to be unique, hence the primary key**), (long and short) name, route colors, text colors, and route type. We assume **Agency has a One-Many relationship with Routes** because an agency can manage multiple routes, but a route must be managed by an agency for it to be considered a transportation route. Note that we also assume an agency must manage a route, or else it would not be included in the table. Hence, we chose the minimum number of routes to be one.

In the Fare_Attributes table, we store information regarding the different fares in the system. This includes information regarding the fare ID (**which we assume to be a unique identifier, so it is the primary key**), price, accepted currency, payment method, transfers, and the duration of each transfer. In terms of relationships, Fare_Attributes is only connected to the Routes table in terms of the fare rules, or how fares are distributed among routes. Note that we hold the ID of the original stop, the destination stop, and the related zones in the Fare_Rules relationship table. **We assume this is a Many-Many relationship**. In one case, the fare attributes have a set of fare rules that are used in multiple routes. In the other case, fare systems can use a combination of fares at a time. These combinations can be used in a route, allowing it to utilize multiple fare attributes. Lastly, we assume the minimum amount of fare attributes and routes in the relationship is one because a route must have some sort of fare, and fare attributes are assumed to be tied to at least one route or it wouldn't exist in the system.

The Trip table contains information about what trips are available on a particular route. It holds information regarding the route ID (**which references the route ID in the Routes table**), the service schedule ID (**which references the service ID in the Calendars table**), the unique ID of the trip (**which we assume to be a primary key since it's a unique identifier that doesn't reference another table**), information regarding the trip destination, a direction ID indicator, and the ID of the route shape. While the Trips table references information from other tables, it does not use them to uniquely identify each trip. **It leaves the unique identification to the trip ID, letting us assume this is not a weak-entity table.** However, it will have multiple relationships, defined below.

The first is a relationship with the Riders table. In Riders, we hold the user's login attributes for using the transportation website. This includes their unique user ID (**which we assume to be a primary key since it's a unique identifier that doesn't reference another table**), email, and password. **The Riders table has a Many-Many relationship with the Trips in terms of traveling**, with the accompanying Travel relationship table holding information regarding the date of the trip. We can assume this because each trip can be traveled by multiple riders, and a rider can travel on many trips. Note that the minimal value for Trips can be zero because a rider could have made an account but not created or taken any trips yet. The number of riders could also have a minimum value of zero because a recently created trip could have a period where no rider has traveled in it.

Trips has a Many-One relationship with Calendars in terms of operation. The Calendars table holds the unique service ID (**which we assume to be a primary key since it's a unique identifier that doesn't reference another table**), the days of operation, and the start and end dates of service. **The relationship between Trips and Calendars is Many-One.** This is because service dates can be operated by many trips, but a trip must be operated by a service. Note that we assume a schedule must be operated by at least one trip, or else it would not be a service in the first place. Note that **by our assumption, Trips does reference the service ID, but does not use it as a unique form of identification (hence, it is not a weak-entity relationship).**

Additionally, **Trips has a Many-One relationship with the Routes table**. This is because a route can contain multiple trips, but a trip can only be contained in one route by definition. A route must contain at least one trip, or else there is no way for the person to go from one destination to another on that route. Hence, we can assume a route must have at least one trip. Note that **by our assumption, Trips does reference the route ID, but does not use it as a unique form of identification (hence, it is not a weak-entity relationship).**

The Stop_Times has a weak-entity relationship with the Trips table. Stop Times holds data on the trip ID (**which references the trip ID in the Trips table, making it a foreign key**), the arrival time, departure time, and the order in which the stops are used for each trip. **Both the trip ID and the stop sequences act as unique identifiers, hence creating a composite key.** **Stop_Times has a Many-One relationship with Trips in terms of association.** This is because a trip is associated with many stop times, but a stop time must be associated with a specific trip. Note that we require at least one stop time, or else there would not be a trip, to begin with.

Additionally, **Stop_Times has a Many-One relationship with the Stop table**. The Stops data table describes each stop in Sao Paulo in terms of a unique stop ID (**which we assume to be a primary key since it's a unique identifier that doesn't reference another table**), name, and the coordinates of the stop in terms of longitude and latitude. **Note that by our assumption, Stop_Times does reference the stop_id attribute in the Stops table. However, it doesn't use it for identification, so this is not a weak-entity relationship.** This relationship is Many-One because every stop can have multiple stop times, but a stop time can only be associated with a particular stop. Note that there must be at least one stop time, or else that location wouldn't be a stop, to begin with.

Next, the Shapes table has a weak-entity relationship with Trips in terms of usage. Shapes stores information that holds the shape ID (**which is referenced by the shape_id in the Trips table, creating the weak-entity relationship**), the longitude, latitude, distance traveled, and point sequences of various paths. **Both the trip ID and the point sequence values in the Shapes table act as a unique identifier, hence creating a composite key. The usage relationship between Shape and Trips is Many-One** because every point that creates a unique shape must be used by a trip, but a trip will use multiple points that help create a shape. Note we require at least one point in the shapes table, or else the trip wouldn't exist.

Finally, there is a Frequency table that has a weak-entity relationship with Trips in terms of operation. The Frequency relationship table contains information regarding the start times, end times, and the time between the departures (which is stored in the attribute headway_secs). It also references the trip ID in the Trips table. **<u>Both the trip and the start times for the frequency act as a unique identifier for each frequency, hence creating a composite key</u> (using the trip_id as a source of identification results in the weak-entity relationship). Overall, Frequencies has a Many-One relationship with Trips.** This is because a trip can have many frequencies of operation, but each frequency can only be operated in one trip. Note that the minimum value for the many component is one because if the trip did not have a frequency, it would not be a trip in the first place.

Work Cited:

Google. (n.d.). *Fares - Transit Partners help*. Google. https://support.google.com/transitpartners/answer/6377425?hl=en&ref_topic=6377359&sjid=5708126637267213214-NA

Google. (n.d.-b). *Trips - Transit Partners help*. Google. https://support.google.com/transitpartners/answer/6377388?hl=en&ref_topic=6377359&sjid=5708126637267213214-NA