JS Programming Planning Doc

Task

Build a terminal-based Node.js app that parses TransLink data and provides a summary of all route stops.

Flow of the Application User Interface

1. Prompt User Input
2. Display Data w/ console.table()
    a. The short name for the route
    (Table column name: Route Short Name);
    b. The long name for the route
    (Table column name: Route Long Name);
    c. The service ID for the trip
    (Table column name: Service ID);
    d. The head sign for the trip (otherwise known as destination sign)
    (Table column name: Heading Sign);
    e. The scheduled arrival time of the vehicle at the starting stop
    (Table column name: Scheduled Arrival Time);
    f. The live arrival time of the vehicle at the starting stop
    (Table column name: Live Arrival Time);
    g. The live geographic position of vehicle
    (Table column name: Live Position).
    h. Estimated time to destination stop
    (Table column name: Estimated Travel Time).
    The calculation of travel time does not need to take into account any live data.
3. Re-Prompt for User Input

File Structure

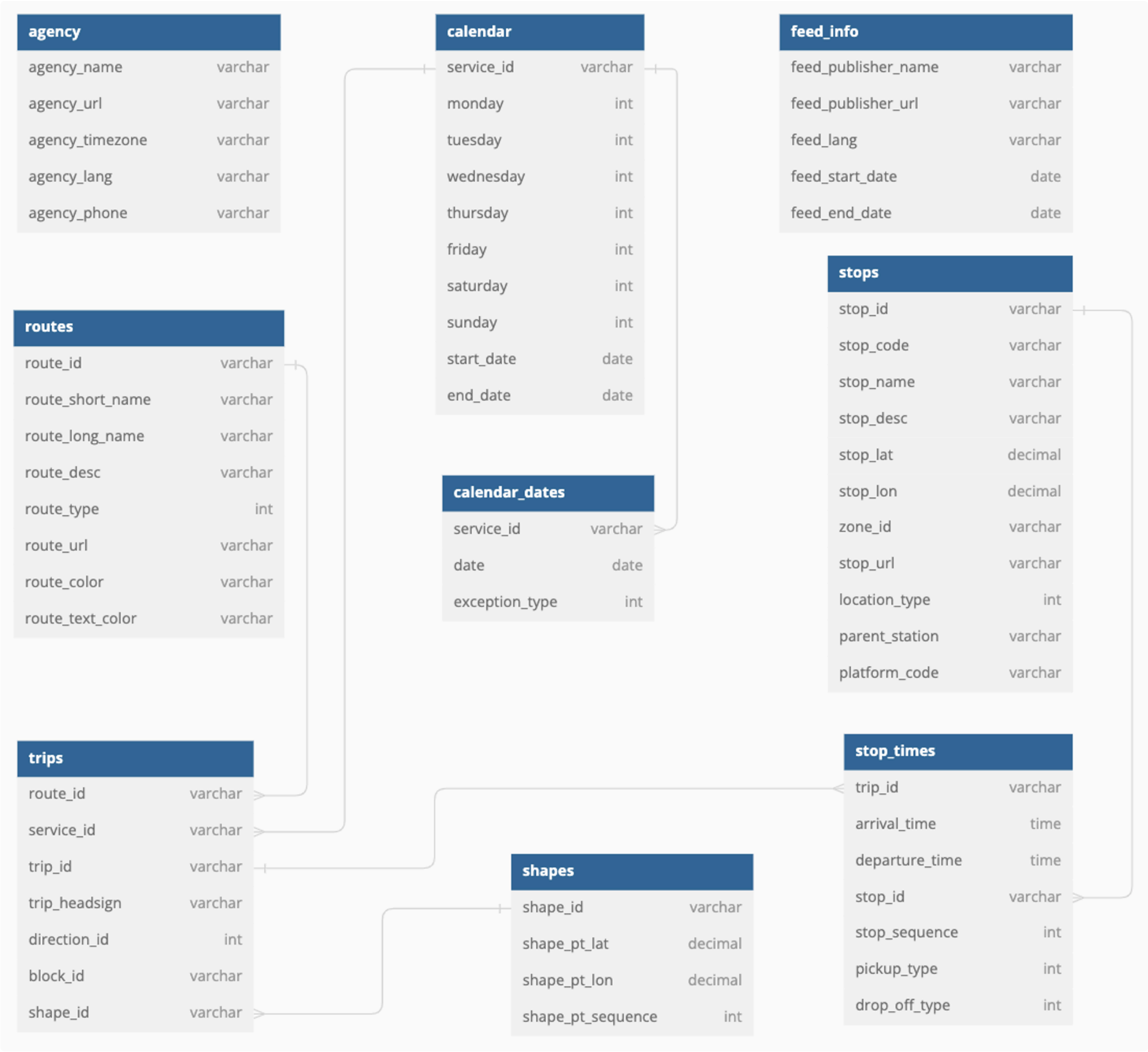
- TXT Static Data
- JSON Live Data Caching
- JS Main

- JS Helper Functions

Steps to Achieve Task

1. Build a UI Loop
2. Implement Join Functions
3. Implement Live Data Caching

Visualisation



agency	
agency_name	varchar
agency_url	varchar
agency_timezone	varchar
agency_lang	varchar
agency_phone	varchar

calendar	
service_id	varchar
monday	int
tuesday	int
wednesday	int
thursday	int
friday	int
saturday	int
sunday	int
start_date	date
end_date	date

feed_info	
feed_publisher_name	varchar
feed_publisher_url	varchar
feed_lang	varchar
feed_start_date	date
feed_end_date	date

stops	
stop_id	varchar
stop_code	varchar
stop_name	varchar
stop_desc	varchar
stop_lat	decimal
stop_lon	decimal
zone_id	varchar
stop_url	varchar
location_type	int
parent_station	varchar
platform_code	varchar

routes	
route_id	varchar
route_short_name	varchar
route_long_name	varchar
route_desc	varchar
route_type	int
route_url	varchar
route_color	varchar
route_text_color	varchar

calendar_dates	
service_id	varchar
date	date
exception_type	int

trips	
route_id	varchar
service_id	varchar
trip_id	varchar
trip_headsign	varchar
direction_id	int
block_id	varchar
shape_id	varchar

stop_times	
trip_id	varchar
arrival_time	time
departure_time	time
stop_id	varchar
stop_sequence	int
pickup_type	int
drop_off_type	int

shapes	
shape_id	varchar
shape_pt_lat	decimal
shape_pt_lon	decimal
shape_pt_sequence	int

▼ DBML

```
Table agency { agency_name varchar agency_url varchar
agency_timezone varchar agency_lang varchar agency_phone varchar
} Table calendar_dates { service_id varchar [ref: >
calendar.service_id] date date exception_type int } Table
calendar { service_id varchar [pk] monday int tuesday int
wednesday int thursday int friday int saturday int sunday int
start_date date end_date date } Table feed_info {
feed_publisher_name varchar feed_publisher_url varchar feed_lang
varchar feed_start_date date feed_end_date date } Table routes {
route_id varchar [pk] route_short_name varchar route_long_name
varchar route_desc varchar route_type int route_url varchar
route_color varchar route_text_color varchar } Table shapes {
shape_id varchar [pk] shape_pt_lat float shape_pt_lon float
shape_pt_sequence int } Table stop_times { trip_id varchar [ref:
> trips.trip_id] arrival_time time departure_time time stop_id
varchar [ref: > stops.stop_id] stop_sequence int pickup_type int
drop_off_type int } Table stops { stop_id varchar [pk] stop_code
varchar stop_name varchar stop_desc varchar stop_lat float
stop_lon float zone_id varchar stop_url varchar location_type
int parent_station varchar [ref: > stops.stop_id] platform_code
varchar } Table trips { route_id varchar [ref: >
routes.route_id] service_id varchar [ref: > calendar.service_id]
trip_id varchar [pk] trip_headsign varchar direction_id int
block_id varchar shape_id varchar [ref: > shapes.shape_id] }
```

Important Joins

Stop details (given route-short-name)

Find route given route.route_short_name.

Join routes.route_id on trips.route_id (need only 1 = find())→

Join trips.trip_id on stop_times.trip_id →

Join stop_times.stop_id on stops.stop_id (need all stops = filter())

Get all route details (Given route route-information

- The short name for the route (Table column name: Route Short Name);
- The long name for the route (Table column name: Route Long Name);
- The service ID for the trip (Table column name: Service ID);
- The head sign for the trip (otherwise known as destination sign) (Table column name: Heading Sign);
- The scheduled arrival time of the vehicle at the starting stop (Table column name: Scheduled Arrival Time);
- The live arrival time of the vehicle at the starting stop (Table column name: Live Arrival Time);
- The live geographic position of vehicle (Table column name: Live Position).
- Estimated time to destination stop (Table column name: Estimated Travel Time). The calculation of travel time does not need to take into account any live data.

References