driver: A driver is assigned to a main bus, but also a substitute bus if another driver is unable to work.

- d_driverid: The ID of the driver.
- d_busid: The ID of the bus the driver is assigned to. This is used to JOIN with bus.
- d_drivername: The name of the driver.
- d_type: Describes the driver's role, either "PRIMARY" or "SECONDARY". PRIMARY drivers work from Mon-Sat. SECONDARY drivers only work on Sun.
- d_subid: The driver ID this driver is substituting for. A driver cannot sub for another driver of the same type. 0 indicates this driver is not substituting.
- d_subbusid: The bus ID that belongs to the substituted driver. 0 indicates this driver is not substituting.
- d_status: Determines if the driver is able to work. A value other than "ABLE" indicates the driver cannot work, such as "SICK".
- d_hours: The number of hours the driver has worked this week. Resets at the end of Sunday every week.

bus: One bus is assigned to a route on a given day.

- b_busid: The ID of the bus. Can be used to JOIN with driver and route.
- b_busname: The name of the bus.
- b_condition: Determines if the bus can be driven. Example values are "FUNCTIONAL", "DAMAGED", and "MAINTENANCE".
- b_gas: The current fuel in the bus, in gallons. The bus must have enough gas to complete all paths in the route. The standardized capacity of all buses is 60.

stop: A stop in a route.

- s_stopid: The ID of the stop.
- s_locationx: The x coordinate of the stop. 0 is the left.
- s_locationy: The y coordinate of the stop. 0 is the bottom.
- s_stopname: The name of the stop.

stopdetails: Has the ordered stop information for each route. **Currently unused.**

- sd_routetype: The type of route.
- sd_stopid: The ID of the stop.
- sd_stopno: The first, second, etc. stop in the route. 0 represents the starting point.

path: A path from one stop to another. This contains the distance information for a route.

- p_pathid: The ID of the path.
- p_startid: The ID of the starting stop.

- p_endid: The ID of the ending stop.
- p_pathlength: The length of the path in miles.
- p_gasusage: The amount of gas required to complete this path. A bus should check that it has sufficient gas. All buses have 7.5 MPG.
    - To calculate the gas usage, use the formula: p_pathlength / 7.5.
    - For every 0.1 miles: 0.1 / 7.5 = 0.0133.

pathdetails: Has the ordered path information for each route.

- pd_routetype: The type of route.
- pd_pathid: The ID of the path.
- pd_pathno: The first, second, etc. path in the route. 1 represents the first path.

route: A full route comprised of multiple stops and paths.

- r_routeid: The ID of the route. Used to JOIN with routedetails.
- r_busid: The ID of the bus assigned to this route, involved in JOIN bus.
- r_routename: The publicly known name of the route.
- r_routetype: The type of route. 1 is full route, 2 is half route A, 3 is half route B.
    - Refer to “Additional Information” section below for details.
- r_starttime: The expected start time of the route.
- r_endtime: The expected end time of the route.
- r_day: The day the route runs on. Possible values are “MON”, “TUE”, “WED”, “THU”, “FRI”, “SAT”, “SUN”.

roadconditions: Determines if a road can be driven on, which will affect the availability of the related path. **Currently unused.**

- rc_pathid: The ID of the path this road belongs to.
- rc_clearance: Determines if the road can be driven on. A non “CLEAR” value indicates the road is unavailable.
- rc_comments: A description of the road’s conditions.

Additional Information:

- The Database Station is operational from 6:00 AM to 12:00 AM. No drivers are active from midnight until 6 AM.
    - In the database, 0:00 represents midnight and 12:00 represents noon.
- The full route takes 15 minutes per cycle, but it visit each stop at least once.
    - The full route visits the following stops in order: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 8, 5, 18, 19, 20, 1. It is 3.2 miles long.
- The half routes take 7.5 minutes per cycle. While they cover less stops, they visit each stop more frequently and are more suitable for shorter distance travel.

    - Half route A visits the following stops in order: 6, 7, 8, 5, 18, 19, 20, 1, 2, 3, 4, 5, 6. It is 1.4 miles long.
    - Half route B visits the following stops in order: 15, 16, 17, 8, 9, 10, 11, 12, 13, 14, 15. It is 1.8 miles long.
- There are 3 drivers in rotation for each route group that work for 2 hour shifts. The last 15 minutes in a shift are dedicated to maintenance and refueling.
    - Rotation A's shifts are: 6:00-8:00 AM, 12:00-2:00 PM, 6:00-8:00 PM.
    - Rotation B's shifts are: 8:00-10:00 AM, 2:00-4:00 PM, 8:00-10:00 PM.
    - Rotation C's shifts are: 10:00-12:00 PM, 4:00-6:00 PM, 10:00-12:00 AM.
- Drivers alternate between route groups during their shift. As there are 3 route groups and 3 drivers per shift, each driver will operate in every route group in a single day. However, their shift groups remain the same for every day.
- A separate group of drivers is active only on Sundays, or if drivers are unable to work.

Full Route Drivers:

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
6-8	1	7	4	1	7	4	10
8-10	2	8	5	2	8	5	11
10-12	3	9	6	3	9	6	12
12-14	4	1	7	4	1	7	10
14-16	5	2	8	5	2	8	11
16-18	6	3	9	6	3	9	12
18-20	7	4	1	7	4	1	10
20-22	8	5	2	8	5	2	11
22-24	9	6	3	9	6	3	12

Half Route A Drivers:

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
6-8	7	4	1	7	4	1	13
8-10	8	5	2	8	5	2	14
10-12	9	6	3	9	6	3	15

12-14	1	7	4	1	7	4	13
14-16	2	8	5	2	8	5	14
16-18	3	9	6	3	9	6	15
18-20	4	1	7	4	1	7	13
20-22	5	2	8	5	2	8	14
22-24	6	3	9	6	3	9	15

Half Route B Drivers:

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
6-8	4	1	7	4	1	7	16
8-10	5	2	8	5	2	8	17
10-12	6	3	9	6	3	9	18
12-14	7	4	1	7	4	1	16
14-16	8	5	2	8	5	2	17
16-18	9	6	3	9	6	3	18
18-20	1	7	4	1	7	4	16
20-22	2	8	5	2	8	5	17
22-24	3	9	6	3	9	6	18

Function Ideas: Different versions of the app can be launched, each with a different set of functions available to the user depending on their type.

- Public users can only access findRoute and viewSchedule.
- Drivers can access functions that can log their working hours and status.
- Admins can access all functions, including ones that can update drivers.
- Some functions are support functions which are only used for calculations.

calculateDistance(int, int, int, string): Calculates the distance from one stop to another.

- Users: Functions.
- Inputs: Starting stop, ending stop, route type, day.
- Outputs: Distance from start to end.

- Pseudocode: Finds the path with the starting stop ID and the path with the ending stop ID. Joins with pathdetails and sums all the path lengths that have a pathno in between. This is based on route type.
    - If the paths “overflow” (e.g. pathno 20 to 5), calculate from 20 to the max (22) and then from 1 to 5.

calculateTime(int, int, int): Calculates the time it takes to travel from one stop to another.

- Users: Functions.
- Inputs: Starting stop, ending stop, route type.
- Outputs: Time in minutes it takes to travel a specified distance.
- Pseudocode: Calls calculateDistance and then adds strftime based on the distance / mpm. Rounds to the nearest minute.
    - For full route, it travels 3.2 / 15 = 0.2133 miles per minute.

processTime(string): Converts a time string to hours and minutes.

- Users: Functions.
- Inputs: Time.
- Outputs: Hours and minutes.

roundTime(string): Calls processTime, but rounds the time to be within a route.

- Users: Functions.
- Inputs: Time.
- Outputs: Time.

findRoute(int, int, string, string): Finds a route from one stop to another.

- Users: Any.
- Inputs: Starting stop, ending stop, day, current time.
- Outputs: Route name, time bus should arrive to starting point, time bus should arrive at destination.
    - Your route ideal route is (route name), which should arrive at your position by (time). You should then arrive at your destination by (time).
- Pseudocode: Finds all schedules that include both stops and whose time is within the current time. Calls calculateTime twice: once for the starting time to the

viewSchedule(string): Shows all running routes and their times.

- Users: Any.
- Inputs: The day.
- Outputs: Shows all running routes and their times.
    - The bus schedule for (day) is:
    - (list of routes)

logRoute(int, int, string): Driver manually logs when they complete a route. Their hours and remaining gas will be updated accordingly.

- Users: Drivers.
- Inputs: Driver ID, bus ID, route type.
- Outputs: The amount of hours added and their remaining gas. This is based on the route type.
    - (driver name), (hours) hours have been added to your work week. Your bus, (bus name), has (gas) gallons of fuel remaining.

updateRefuel(int): Driver logs when they refuel a bus. Sets the bus's gas to 60.

- Users: Drivers.
- Inputs: Bus ID.
- Outputs: Confirmation message that the bus has been refueled.
    - (bus name) has been refueled.

updateDriver(int, string): Updates a driver's status.

- Users: Drivers, admins.
- Inputs: Driver ID, status update.
- Outputs: Driver name and their new status.
    - Successfully updated (driver name)'s status to (status).

updateBus(int, string): Updates a bus's condition.

- Users: Drivers, admins.
- Inputs: Bus ID, status update.
- Outputs: Bus name and its new condition.
    - Successfully updated (bus name)'s condition to (condition).

mySchedule(int): Shows the days and hours that the driver should be working on.

- Users: Drivers
- Inputs: Driver ID.
- Outputs: Shows the days and hours that the driver should be working on. If they are substituting, shows those days as well.
    - Your schedule is:
    - (hours and days)

showAlerts(): Shows any drivers or buses that are out of service.

- Users: Drivers, admins
- Inputs: None.
- Outputs: Lists the names of any drivers or buses that are out of service.

    - (driver name) is unable to work. Reason: (reason).
    - (bus name) is out of service. Reason: (reason).
    - There are no alerts at this time.

requestDriver(int, int): Sends a request to take on another driver's schedule. A driver cannot take on the schedule of another driver of the same type.

- Users: Drivers
- Inputs: The user's driver ID, the driver ID of the person they are filling in for.
- Outputs: Message that either automatically denies the request or confirms that a request was sent, which will notify admins.
    - (driver name), your request to fill in for (other driver name) has been sent successfully. You will be notified shortly if your request has been approved.
    - Sorry, (driver name). You are unable to fill in for (other driver name) since you are both (driver type) drivers.
    - Sorry, (driver name). You are unable to fill in for (other driver name) since you are already filling in for (another driver name). Please send a request to update your substitute status if this is a mistake.

showRequests(): Shows requests sent by drivers, such as from requestDriver.

- Users: Admins
- Inputs: None.
- Outputs: Shows any requests.
    - (driver name) has requested to fill in for (other driver name).
    - There are no requests at this time.

reassignDriver(int, int): Updates d_subid and d_subbusid of a driver.

- Users: Admins
- Inputs: Driver ID of the substitute, driver ID of the one to add to the substitute.
    - If the second input is 0, set the first driver's d_subid and d_subbusid back to 0.
- Outputs: Confirmation message.
    - Successfully updated (driver name)'s schedule.
    - Invalid update request.

reassignBus(int, int): Updates r_busid and d_busid and/or d_subbusid to another bus ID, used if a bus is no longer operational.

- Users: Admins
- Inputs: Bus ID, bus ID to replace with.
- Outputs: Confirmation message.

    - Successfully updated all routes and drivers associated with (old bus ID) to (replacement bus name).
    - Invalid update request.

restoreBus(): Recreates the route table with default values. Updates drivers to have their default buses.

- Users: Admins
- Inputs: None.
- Outputs: Confirmation message.
    - Successfully reset routes and drivers to their default assigned buses.