

Assumptions:

You can make the following assumptions in this project.

1. Ez pass is a RFID device (called transponder) that can be mounted on a vehicle's windshield and when the vehicle passes a toll antenna the antenna can read the radio signal sent by the transponder and retrieves the transponder ID.
2. The system needs to store data about accounts. Each account has account id, name of account holder, email, address, balance (pre-paid amount), and password.
3. Each account will have a list of transponders. Each transponder has a transponder id. Users of the account can use any of the associated transponders.
4. The system stores information about vehicles. You can assume that the database has information for all vehicles in US. Each vehicle has a vehicle ID, license plate number, state, class (1 is car, 2 is truck), address of owner, and an optional account id. Only some vehicles will be associated with an account. Others are not because either their owner does not have an ez pass account or the owner has not registered the vehicle with the account.
5. When a vehicle passes a toll lane, there are two ways that the system can recognize who should pay for it. The first way is when a transponder is mounted and its signal is received by the antenna. The account associated with that transponder will be charged. If this does not happen (e.g., there is no transponder or the transponder malfunctioned), a camera will take a picture of the license plate and the system will use license plate number and state to locate the vehicle. If the vehicle is registered with an account that account will be charged. Otherwise the system sends a video toll bill to the owner of the vehicle. The fee for video toll is higher.
6. The system stores a discount_rate table which only has one row and three columns: the discount rate for EZ pass, the rate for video toll, and the administrative fee. If the toll rate is \$10, and the discount rate for EZ pass is 0.9, the rate for video toll is 1.0 and administrative fee is \$3. Then the real toll for using ez pass is 10×0.9 . The real toll for video toll is $10 \times 1 + 3 = \$13$.
7. The system stores information about toll roads. Each road is a road ID and road name. E.g., I95 or I495.
8. Each road has a number of exits. Each exit has a road ID and integer exit number. E.g., exit 143 of road 1. There is also a description of the exit (e.g., junction with 395). The exit numbers may not be continuous, e.g., the exit numbers could be 143, 156, 161. You can assume that on the same road, every exit number is unique, and these numbers are ordered in a certain direction. E.g., for I95 in Virginia, exit numbers are in increasing order northbound. We use the same exit number for exits on both sides of the road. You can distinguish the direction of the road by the increasing or decreasing order of exit numbers. E.g., if a car enters a toll road at exit 143 and leaves at exit 161. The car must be traveling in the increasing order direction.
9. The system uses a dynamic toll rate scheme to better manage the traffic. There is a toll_rate table that stores toll rate information for every pair of consecutive exits on a toll road for a given direction and at a certain time period. The columns include road id, start exit id (the start exit number), end exit id, direction, toll_start_time, toll_end_time, car toll rate, truck toll rate. For example, suppose a road has three exits: 143, 156, and 161. Suppose a toll is charged from July 30 2018 from 6 am to 8 am on both directions. There should be 4 rows inserted in the toll_rate table for this time period: between exit 143 (as starting exit id) and 156 (as ending exit id),

- between 156 and 161, between 156 and 143 (opposite direction where 156 is start exit id and 143 is end exit id), and between 161 and 156 (opposite direction).
10. The toll rate for different time/direction can be different. E.g., the toll from exit 143 to 156 (north bound) for road 1 is \$2.00 from 6 to 10 on that date but is \$0.5 for exit 156 to 143 (south bound).
 11. The system stores trip information, which represents information about a vehicle travels on a toll road at a specific time. The information includes a trip id, account id, transponder id, vehicle ID, road id, entering exit id, exiting exit id, entrance time, toll, status (how the account is looked up), and deducted (whether this toll has been deducted).
 12. Toll will be computed by adding up tolls for each pair of exits in the range of the entering exit id and exiting exit id at the correct time range and correct direction.
 - a. The direction is not explicitly stored in trip table but you can infer it by checking whether entering exit id is greater than exiting exit id. E.g., if a car enters a toll road at exit 143 and leaves at exit 161. The car must be traveling in the increasing order direction of the road.
 - b. The entrance time should fall between a toll rate's start and end time. For example, if a car travels at 7 am on 2018-7-30 from exit 143 to exit 161, and the toll rate table contains two rows where the toll is \$2.00 from exit 143 to exit 156 and \$1.00 from exit 156 to exit 161 from 6 to 10 am on that date, then the total toll should be $\$2.00 + \$1.00 = \$3.00$.
 - c. The status column in trip table indicates how the vehicle is identified. If status = 1, a valid transponder is used to locate the account, when status = 2, transponder may be missing but the license plate (license and state) can be linked to an account. When status = 3, the license plate can be used to link to a vehicle but not an account (the vehicle owner does not have a ezipass account). When status = 1 or 2, if there is enough balance in the account ezipass rate will apply. If status = 3 or there is not enough balance video rate will apply (a video toll bill will be sent).
 - d. The deducted column in trip table indicates whether the toll of this trip has been deducted from account or a video toll bill has been generated. This column can be used to prevent deducting the same trip twice.
 13. The system needs to store payment to an account, including a payment id, account id, payment date, and amount.
 14. The system stores video_bill information, including a bill id, bill date, associated trip id, and status. Status = 1 means the bill has been generated but not sent, 2 means it is sent. 3 means it is paid.
 15. The system stores a message table which has a message id, an account id, a message time and body of the message.

Features for user management:

1. Create a new account. Input includes name of user, email, address, and password. The procedure should check whether the email already exists in the account table. If so, please print a message saying the account exists. Otherwise create the account with a new account ID.
2. Allow a user to login by providing email and password. Please check whether email exists and password matches. If not, please print a message to indicate the error. Otherwise print a message to indicate user has logged on. The procedure should return a value 1 for success login and 0 for unsuccessful log in.
3. Allow a user to read messages providing account id and a starting date. Print out messages for that account since that date. Please handle the case when the account id is invalid.

Features related to vehicles and transponders

4. Add/delete a vehicle to an account. To add a vehicle, the user needs to provide account id, license, state, address, and class. The procedure will check whether the license and state is already associated with the account. If not the vehicle is added to the account with a new vid. Otherwise print a message the vehicle exists and update the class and address.

To delete a vehicle, just provide account id, license and state. The procedure will delete the vehicle if it is associated with that account and has the same license and state. Please handle the case when the account id is invalid or the license or state does not match.

The procedure also needs to insert a message saying that a vehicle has been added or deleted.

5. Add/delete a transponder to an account. To add a transponder, the input includes account id and transponder id. Please check if the transponder already exists. If so it cannot be added. To delete a transponder, just provide account id and transponder id. Please handle the case when the account id is invalid or the transponder does not match any existing transponder in that account. The procedure also needs to insert a message saying that a transponder has been added or deleted.
6. *** Matching an account given transponder id, license plate number and state. The output includes an account, a vehicle id and a status. It then looks up the account first using transponder id. If the transponder is not null and it matches an existing account, return the account id and status 1 (normal). Please also check whether the license plate# and state matches any of the vehicle associated with the account. If there is no match, insert a message reminding the user that a vehicle not associated with the account has used the transponder (The transponder may get stolen or the user forgot to register the car).

If the transponder is null (e.g., when it is not correctly mounted) or no account has this transponder (in case the user has not registered the transponder yet), the license plate number and state will be used to locate the vehicle id. If this vehicle is registered under an account, return that account id, vehicle id, and status 2 (looked up by license plate).

If the vehicle is not registered to any account, return vehicle id and status 3 (no account found).

7. *** Enter a new trip and compute toll. The input includes transponder id, license plate number, state, road id, entering exit id, exiting exit id, time of entering the road.
 - a. The procedure will first call feature 6 to look up the account id. A new trip will be inserted into the trip table based on result of feature 6. The new trip should have account id and vehicle id, and status returned by feature 6.
 - b. Next the procedure looks up the toll_rate table and compute the appropriate toll. The time to enter the road should be between the toll rate's start and end time. The trip may cover multiple legs of the road so you need to add up toll rate for these legs. The pair of two consecutive exits forms a leg (e.g., exit 143 to 156 is a leg). For example, if a car travels at 7 am on 2018-7-30 from exit 143 to exit 161, and the toll rate table contains two rows where the toll is \$2.00 from exit 143 to exit 156 and \$1.00 from exit 156 to exit 161 from 6 to 10 am on that date, then the total toll should be $\$2.00 + \$1.00 = \$3.00$.
 - c. The final toll though depends on status of the trip as well as class of vehicle. If the vehicle is a car then car_toll_rate applies. Otherwise truck_toll_rate applies. If trip status = 1 or 2, an account is located. So the toll equals computed toll * ez pass discounted rate stored in rate_discount table. If trip status = 3 (no account located), the toll equals computed toll * video toll rate stored in rate_discount table plus administration fee stored in the same table.
E.g., if the computed toll is \$5. If status = 1 or 2 and discounted rate for ez pass is 10% off (or 0.9 of full rate), the real toll is $\$5 * 0.9 = \4.5 . If video toll rate is 100% of original toll and administration fee is \$3. The real toll when status = 3 is $\$5 * 1 + \$3 = \$8$. The feature also updates the toll column in trip table to reflect the computation.
8. *** Deduct toll of a trip from an account or generate a video toll bill. Input is trip id. This feature is called after feature 7 is called.
 - a. First check whether the trip's toll has been deducted, if so do nothing and print a message saying the toll has been deducted.
 - b. If the trip status is 1 or 2 (account found) and the account has enough balance, deduct the toll amount from balance.
 - c. If the balance is not sufficient (will become negative), recompute the toll as the video toll rate (see feature 7), update the toll and status column in the trip table such that the toll will be video toll, and generate a video toll bill with a newly assigned bill id, bill date as current date, trip id and status 1 (bill is generated). Please also insert a message to message table reminding the user to replenish her account.
 - d. If the trip status is 3 (account not found), generate a video toll bill as above.
 - e. Please set deducted column to 1 (i.e., this trip's toll has been deducted) at the end. This will prevent multiple deduction of the same trip.
9. Allow a user to make a payment to an account. Input includes account id and amount. Add the amount to that account and insert a row into the payment table with a new payment id, account id, current date as payment date, and amount. Please also insert a row to message table saying payment received. Please handle the case when the account cannot be found.

10. Update the video_bill's status when the bill is sent or paid. Input is bill id and new status. Update the bill's status if the bill id is valid.

Features to look up trips or bills

11. Given an account id and date range, display all trips and payments in that date range for that account. For each trip shows transponder id, license plate number, road name, entering exit id, exiting exit id, entrance time and toll. For each payment, show payment date and amount. Please check whether account id is valid.
12. Generate a monthly statement for an account. Input includes account id and last statement date. This feature prints out all trips and payments from the last statement date to one month from the last statement date (you can call feature 11), and also computes the total toll, total payment in the period. Please handle the case when account id is invalid.

Statistical features

13. Given a date range, print out road name, total toll for that road in the date range, total number of trips in the date range. Please order the results in descending order of total tolls.
14. ***Given a date range and a road id, print out number of trips made through each exit in each direction. Order the results in descending order of number of trips. Please handle the case of invalid road id.

Hint:

Each road has two directions: the direction with increasing exit id or the direction with decreasing exit id. For a trip based on whether entering eid is greater than or less than exiting eid, you can decide direction of a trip as well.

15. Given a date range and a parameter k, print out the top k accounts with the most toll, the k accounts with the most trips made. Please also print out the top k vehicle id with the highest number of video toll bill, and the top k vehicle IDs with the highest total amount of toll for video toll bill.
16. Given a date range, find pairs of vehicles that travel at approximately similar time and similar route, i.e., the pair of vehicles that have entrance time within 30 minutes from each other and having the same road, same entering and exiting exit id for some trips within the date range. Print out vid of these vehicles and the road id, entering and exiting exit ids, and entrance time of each vehicle.

Hint: you need to do a self join on trip table. To avoid duplicate pairs, you can force first vehicle's ID < second vehicle's ID.