What I did:

I used cache. Each time we want to get records for last 48 hours, program does not run query on database. It first checks a cache variable to see whether a new record inserted into the database after previous database query. If not, it just checks records on cache, deletes if any of them older than 48 hours and returns. If inserted, then it just makes new query to database. By this way, I was able to reduce the amount of time that query is made.

Suggestions:

Instead of storing all navigation records of vehicles, we can update record when new data arrives. And, if we only hold most recent navigation record per vehicle in database, then query to get data would be very simple, no need for checking each record of vehicle if it is most recent one.
Also, we can make use of caching in an improved way. It can be possible that during the lifetime of program, only one query would be made to database. Only, the first query. After that first request, we could store resulted data in cache. And whenever a new navigation record arrives we can check data in cache update if necessary. Later, we can add record to database. And when a new get request arrives, we can just check records in cache to delete older records(out of 48 hour range). Then we return all records in cache. By this way only one initial query will be made to database all other operations will be on cache memory.