Del 2:

March, 26, 2023

—

Project DB2

Name: Kebene Amanuel Baissa

Changes made from DB1

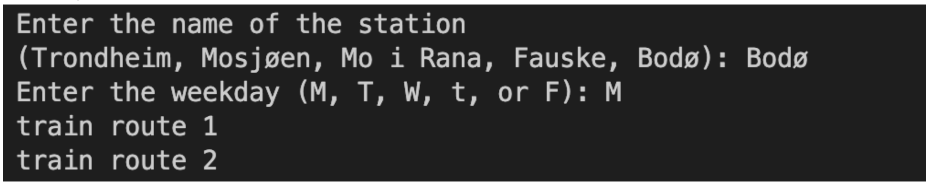
- I have the same table format from **DB1** but have made minor adjustments in **SQL** such as, typing errors.
- Also, I have added "***ON UPDATE CASCADE ON DELETE CASCADE*"** on the tables where this is needed.
- I added an extra column to the **TrainOccurrence** table to have adequate information. Now it also takes in ***startStationId*** and ***endStationId***. This is the easiest solution I found to solve the time problem in timetable two where the time goes form **23:05** to **00:57** and ascending time became a problem.

How I solved the problems

For **A, B**, and **F** I added the values as per-requested in the project write-up in the sql script and it is included in the *SQL* file.

C. For a specified station, the user should be able to get all train routes that stop at the station on a given weekday. Here I assumed that a start station, train stops, and an end station are accepted.

Example result:



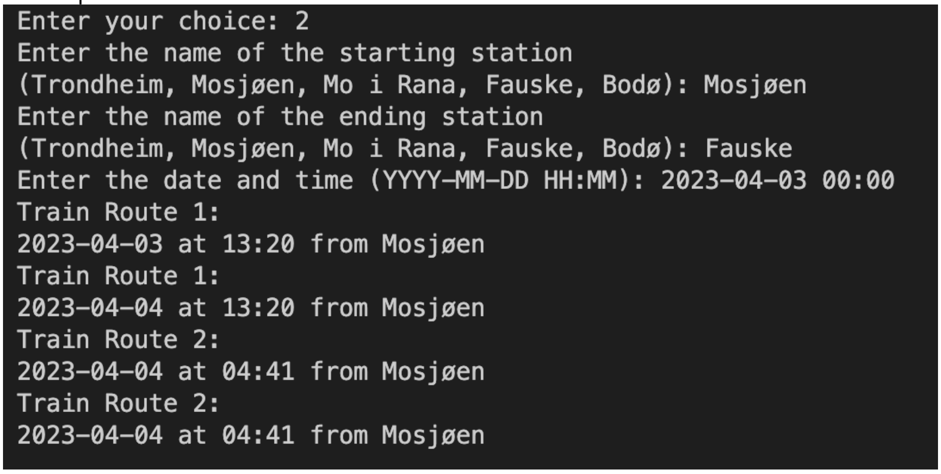
```
Enter the name of the station
(Trondheim, Mosjøen, Mo i Rana, Fauske, Bodø): Bodø
Enter the weekday (M, T, W, t, or F): M
train route 1
train route 2
```

D. The user should be able to search for train routes going between a starting station and an ending station based on the date and time. All routes for the same day and the next should be returned and sorted by time. This functionality should be programmed.

To solve this, I created ***3*** *tables* that contain the timetable for the 3 routes given with train occurrence in python. I also created a column with DateTime ***`(YYYY-MM-DD HH: MM)`*** to make it easier when comparing the time by combining the occurrence date and time.

I also take in DateTime in the form of ***`(YYYY-MM-DD HH: MM)`*** to make it easier. I also add for two dates ***2023-04-03*** and ***2023-04-04*** for all routes. These tables are in a new **`timetablecreate.py`** file to make the **`usercases.py`** clear.

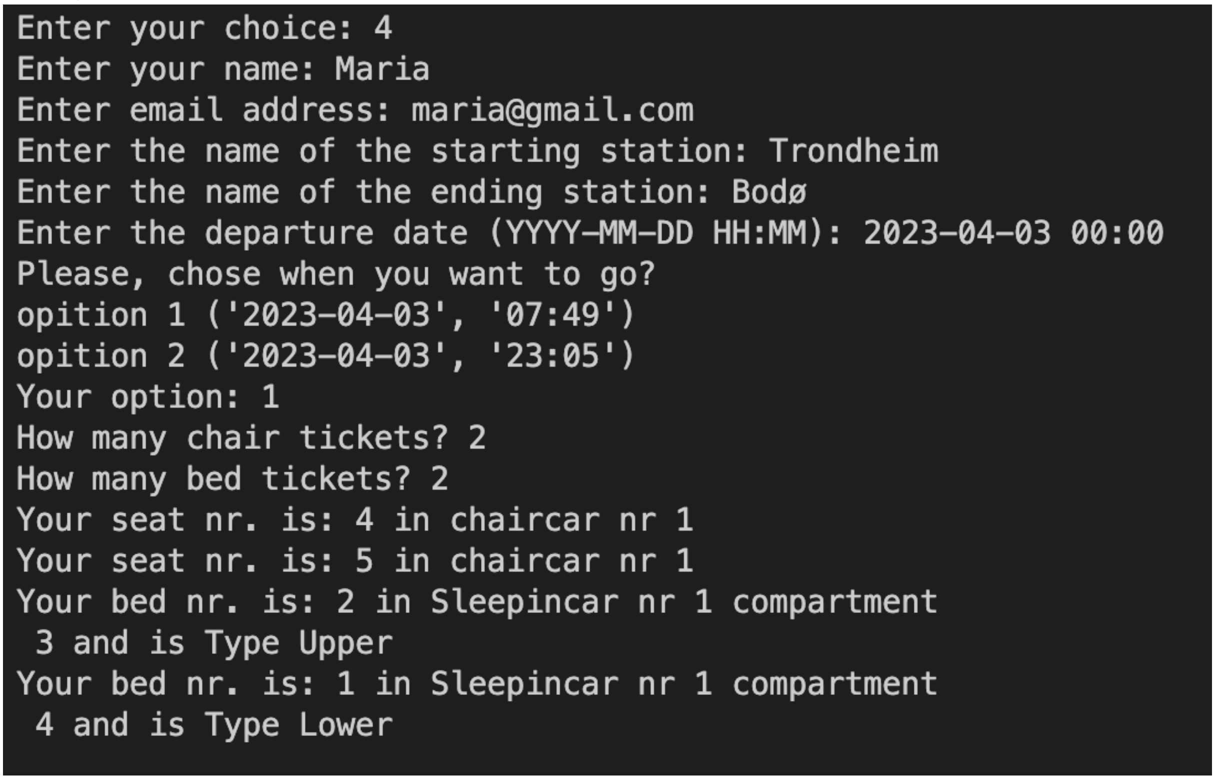
Example result:



```
Enter your choice: 2
Enter the name of the starting station
(Trondheim, Mosjøen, Mo i Rana, Fauske, Bodø): Mosjøen
Enter the name of the ending station
(Trondheim, Mosjøen, Mo i Rana, Fauske, Bodø): Fauske
Enter the date and time (YYYY-MM-DD HH:MM): 2023-04-03 00:00
Train Route 1:
2023-04-03 at 13:20 from Mosjøen
Train Route 1:
2023-04-04 at 13:20 from Mosjøen
Train Route 2:
2023-04-04 at 04:41 from Mosjøen
Train Route 2:
2023-04-04 at 04:41 from Mosjøen
```

G. Registered customers should be able to find available tickets for a desired train route and purchase the tickets they would like. This functionality should be programmed.

Example result:

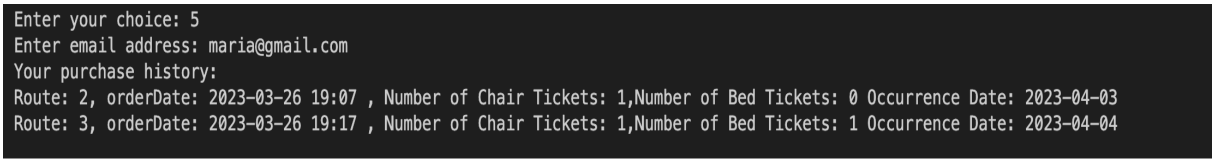


```
Enter your choice: 4
Enter your name: Maria
Enter email address: maria@gmail.com
Enter the name of the starting station: Trondheim
Enter the name of the ending station: Bodø
Enter the departure date (YYYY-MM-DD HH:MM): 2023-04-03 00:00
Please, chose when you want to go?
opition 1 ('2023-04-03', '07:49')
opition 2 ('2023-04-03', '23:05')
Your option: 1
How many chair tickets? 2
How many bed tickets? 2
Your seat nr. is: 4 in chaircar nr 1
Your seat nr. is: 5 in chaircar nr 1
Your bed nr. is: 2 in Sleepincar nr 1 compartment
 3 and is Type Upper
Your bed nr. is: 1 in Sleepincar nr 1 compartment
 4 and is Type Lower
```

h) All information about purchases made for future trips should be available for a user. This functionality should be programmed.

I chose *email* to identify ***custumerID***. Assuming that everyone has unique mail, and every customer must be registered by their mail address.

Example result:



```
Enter your choice: 5
Enter email address: maria@gmail.com
Your purchase history:
Route: 2, orderDate: 2023-03-26 19:07 , Number of Chair Tickets: 1,Number of Bed Tickets: 0 Occurrence Date: 2023-04-03
Route: 3, orderDate: 2023-03-26 19:17 , Number of Chair Tickets: 1,Number of Bed Tickets: 1 Occurrence Date: 2023-04-04
```

The Files

- The original tables with **"insert information"** are in the ***Norwegian_railway_system.sql*** file.

- The ***railwaydatabase.sql*** file is directly exported from ***railwaydatabase.db*** and has the extra helper tables ***Timetable* 1** ,**2** and **3**.

- The ***usercases.py*** is the **python** code connected to ***railwaydatabase.db*** and that is where the user cases are solved.

- The ***timetablecreate.py*** is where the **python** code that makes the **three Timetables** is.

Additional comments

The project is big and I worked alone. Because of the amount of work I didn't have time to make it perfect and the code works very well for the most part. Wrong inputs aren't taken care of in some parts.