Ausgewählte Datenbankkonzepte/-techniken, HTW Berlin

Aufgabe 3: Mobilität

Erledigt von **Olga Petrova, 0562984** **21.11.2019, Berlin**

TIMETABLE SERVICE

Intro

Timetable service stores and provides the data about train trips and schedule
It can create train trips automatically for trains travelling in regular intervals
Timetable service database consists of 3 linked talbes
I decided to use moment.js library to work with dates and times

Der Timetable service speichert und liefert die Daten über Zugfahrten und Fahrpläne.
Es kann Zugfahrten für Züge, die in regelmäßigen Abständen fahren, automatisch erstellen.
Die Datenbank besteht aus 3 verknüpften Tables.
Ich habe mich entschieden, die moment.js Bibliothek zu verwenden, um mit Datum und Uhrzeit zu arbeiten.

“Repeating trips” and “single trips” unterscheidet man indem, dass

- bei “repeating trips” gibt es *interval* parameter (rhythmus, takt, d.h. wie oft train goes) und
- bei “single trips” (einzelnen Zugfahrt bzw. sonderfahrt) interval ist gleich null
  - wenn start_date und end_date ungleich sind, wird ein einzelne Zugfahrt für jeden Tag erstellt.
  - end_time wird ignoriert, es wird immer nur ein Zugfahrt (trip) angelegt (bei start_time)

Tables

train

train_id	lid	direction	start_date	end_date	start_time	end_time	interval
1	20006	ab	2019-11-15	2019-11-20	07:00	08:59	00:30
2	20006	ab	2019-11-15	2019-11-20	9:00	09:59	00:15
3	20006	ba	2019-11-18	2019-11-22	08:00	08:00	null

trip

trip_id	train_id	departure
1	1	2019-11-15 07:00:00
2	1	2019-11-15 07:30:00
3	1	2019-11-15 08:00:00
4	1	2019-11-15 08:30:00
5	2	2019-11-15 09:00:00
6	2	2019-11-15 09:15:00
7	2	2019-11-15 09:30:00
8	2	2019-11-15 09:45:00
9	3	2019-11-18 08:00:00
10	1	2019-11-16 07:00:00
11	1	2019-11-16 07:30:00
12	1	2019-11-16 08:00:00
13	1	2019-11-16 08:30:00
14	2	2019-11-16 09:00:00
15	2	2019-11-16 09:15:00
16	2	2019-11-16 09:30:00
17	2	2019-11-16 09:45:00
18	3	2019-11-19 08:00:00
...		

schedule

trip_id	hid	transit_period
1	10122	2019-11-15 07:00:00
1	10179	2019-11-15 07:04:00
1	10124	2019-11-15 07:12:00
2	10122	2019-11-15 07:30:00
2	10179	2019-11-15 07:34:00
2	10124	2019-11-15 07:42:00
...		

```
create table train (
    train_id integer not null primary key
    generated by default as identity,
    lid integer not null,
    direction varchar(10) not null,
    start_date date not null,
    end_date date not null,
    start_time time not null,
    end_time time not null,
    interval time, --repeating trip has interval, for single trip interval is null
    foreign key (lid) references linie
);

create table trip (
    trip_id integer not null primary key
    generated by default as identity,
    train_id integer not null,
    departure_datetime seconddate not null,
    foreign key (train_id) references train
);

create table schedule (
    trip_id integer not null,
    hid integer not null,
    transit_datetime seconddate not null, --date and time when train is passing through
station
    foreign key (trip_id) references trip,
    foreign key (hid) references haltestelle
);
```

Description

```
/*
 When accessing the resource /add_train, the service takes parameters (line_id,
direction,  start_date,  end_date, start_time,  end_time, interval) and
Adds a record to the train table
Adds one or more records to the trips table
Calls a function, which adds all corresponding rows to the schedule table

"Repeating trips" and "single trips" are distinguished by the fact that
"repeating trips" have interval parameter (rhythm, i.e. how often train goes) and
for "single trips" interval equals NULL, and:
- if start_date and end_date are unequal, a single train trip is created for each day.
- end_time is ignored, only one train trip is created (with start_time), if interval is NULL
*/
```

1) beim Zugriff auf die Ressource /**add_train** z.b. (localhost:3010/add_train?lid..) nimmt der Service die Parameter *(line_id, direction*, *start_date, end_date, start_time, end_time, interval),* und :

- Fügt einen Datensatz zur Tabelle **train** hinzu,
- fügt alle entsprechenden Einträge zur Tabelle **trip** hinzu,
- ruft eine Methode auf, die jeweils alle entsprechenden Einträge zur Tabelle **schedule** hinzufügt

/*
When accessing the resource /remove_train, the service takes parameter train_id and
Removes all corresponding rows from the schedule table
Removes all corresponding rows from the trip table
Removes a corresponding row row from the train table
*/

2) beim Zugriff auf die Ressource **/remove_train** nimmt der Service den Parameter *train_id* und

- löscht den entsprechenden Datensatz aus der Tabelle **train**
- ruft eine Methode auf, die löscht alle mit diesem Datensatz verbundenen Datensätze aus der Tabelle **trips**
- ruft eine Methode auf, jeweils alle entsprechenden Einträge aus der Tabelle **schedule** löscht

/*
When accessing the resource /remove_trip, the service takes parameter trip_id and
Removes all corresponding rows from the schedule table
Removes a corresponding row from the trip table
*/

3) beim Zugriff auf die Ressource **/remove_trip** (einzeln) nimmt der Service *trip_id* und

- löscht den entsprechenden Datensatz aus der Tabelle **trip**
- ruft eine Methode auf, jeweils alle entsprechenden Einträge aus der Tabelle **schedule** löscht

/*
When accessing the resource /get_info, the service returns a JSON
with data distinguished for parameters:
1. **All trains:** (without parameters): returns all data from the "train" table. -
2. **All trips for single train:** (train_id) returns the data from the "trip" table for the train_id.
3. **All stations stops for single trip:** (trip_id) returns the data from the "schedule" table for the trip_id (incl. "bez")
4. **All trips for single line for the period:** (lid, datetime_start, datetime_end) returns the data from the "trip" table for the specified line ("linie") and for all train_id in that period between two given dates.
5. **Schedule for single station for the period:** (hid, datetime_start, datetime_end) returns the data: line id, train_id, trip_id, transit_datetime for this station ("haltestelle") with given hid and time period between two given dates.
*/

4) beim Zugriff auf die Ressource **/get_info** gibt der Service einen JSON mit Daten zurück, der sich bei Parameter unterscheidet:

1. *Ohne Parameter:* gibt die Daten aus der Tabelle **train** zurück
2. *(train_id)* gibt die Daten aus der Tabelle **trip** für die train_id zurück
3. *(trip_id)* gibt die Daten aus der Tabelle **schedule** für die trip_id zurück (inkl. Bez fur Haltestellen)
4. (*lid, datetime_start, datetime_end*) gibt die Daten aus der Tabelle **trip** für die Linie zurück, fur alle train_id in diesem Zeitraum.
5. *(hid, datetime_start, datetime_end)* gibt die Daten: linie id, train_id, trip_id, transit_datetime für diese Haltestelle mit gegebenen hid und Zeitraum

Readme: Installation Manual for timetableservice

1. Create tables using timetableservice.sql
2. Install moment.js: npm install moment

- All trains (#1)
- Trips for single train (#2)
- Stops for single trip (#3)
- All trips for line for the period (#4)
- Schedule for station for the period (#5)

DEMO with exact browser queries for all steps

Initial state: timetable service is stopped, timetable service tables are dropped

- Run sql (in HANA Web IDE) timetableservice.sql to create tables
- Start timetable service

1. Resource ***/add_train***
   - Add trains:

http://localhost:3010/add_train?lid=20001&direction=ab&start_date=2019-12-05&end_date=2019-12-06&start_time=09%3A00&end_time=11%3A00&interval=0%3A20

http://localhost:3010/add_train?lid=20002&direction=ba&start_date=2019-12-06&end_date=2019-12-08&start_time=10%3A00&end_time=13%3A00&interval=1%3A25

http://localhost:3010/add_train?lid=20003&direction=ab&start_date=2019-12-05&end_date=2019-12-08&start_time=12%3A15&end_time=12%3A15&interval=null

http://localhost:3010/add_train?lid=20001&direction=ba&start_date=2019-12-07&end_date=2019-12-07&start_time=08%3A40&end_time=08%3A40&interval=null

2. Resource ***/get_info***
   - All trains (#1): gibt die Daten aus der Tabelle **train** zurück
     Show all trains in #1 output:

http://localhost:3010/get_info

   - Trips for single train (#2): gibt die Daten aus der Tabelle **trip** für die train_id zurück
     Show all trips for train_id=1

http://localhost:3010/get_info?train_id=1

   - Stations (train stops) for trip (#3): gibt die Daten aus der Tabelle **schedule** für die trip_id zurück
     Show all station stops for trip_id=5 anzeigen alle haltestellen für gegebenen trip_id

http://localhost:3010/get_info?trip_id=5

   - All trips for single line for the period (#4):
     Show all trips for line lid=20001 U1 for the period between 2019-12-05 00:00 and 2019-12-06 23:59

http://localhost:3010/get_info?lid=20001&datetime_start=2019-12-05%2000%3A00&datetime_end=2019-12-06%2023%3A59

- Schedule for station for the period (#5) für diese Haltestelle in diesem Zeitraum. Show new train stops for station hid=10288 U Schlesisches Tor for the period of two days (between 2019-12-06 00:00 and 2019-12-07 23:59)

http://localhost:3010/get_info?hid=10288&datetime_start=2019-12-06%2000%3A00&datetime_end=2019-12-07%2023%3A59

3. Resource **/remove_train** // den Zug löschen

Erst in in HANA WEB IDEE schauen, dass train_id 4 noch da ist, und entsprechende trips (28) - das ist einzelne Fahrt, und Datensätze in schedule tabelle für trip 28.

http://localhost:3010/remove_train?train_id=4

- Show that train_id=4 is removed from #1 output
  Zeigt an, dass train_id=4 aus der output entfernt wird.

http://localhost:3010/get_info

Or **show tables in HANA WEB IDEE**: Einträge in train (train_id 4), trip (28), schedule

- Show changes in #2 output (trips are deleted for this train)

http://localhost:3010/get_info?train_id=4

- Show changes in #4 output for line 20001- trips with train_id = 4 should be removed

http://localhost:3010/get_info?lid=20001&datetime_start=2019-12-05%2000%3A00&datetime_end=2019-12-07%2023%3A59

- Show changes in #5 output for station 10288 - schedule records with train_id = 4 should be removed

http://localhost:3010/get_info?hid=10288&datetime_start=2019-12-06%2000%3A00&datetime_end=2019-12-07%2023%3A59

4. Resource **/remove_trip**

http://localhost:3010/remove_trip?trip_id=3