

Table Creation

```
CREATE TABLE Traveller(  
    id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    national_code CHAR(10) NOT NULL UNIQUE,  
    first_name varchar(50) NOT NULL,  
    last_name varchar(50) NOT NULL,  
    phone_number CHAR(11) NOT NULL,  
    date_of_birth DATE NOT NULL,  
    gender BOOL NOT NULL,  
    PRIMARY KEY(id)  
)
```

```
CREATE TABLE Station(  
    id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    name varchar(100) NOT NULL,  
    establishment_date DATE NOT NULL,  
    city varchar(50) NOT NULL UNIQUE,  
    PRIMARY KEY(id)  
)
```

```
CREATE TABLE Station_phone_number(  
    phone_number CHAR(11) NOT NULL,  
    station_id MEDIUMINT UNSIGNED NOT NULL,  
    FOREIGN KEY(station_id) REFERENCES Station(id) ON DELETE CASCADE,  
    PRIMARY KEY(phone_number),  
    INDEX(station_id)  
)
```

```
CREATE TABLE Train_track(  
    id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    from_station_id MEDIUMINT UNSIGNED NOT NULL,  
    to_station_id MEDIUMINT UNSIGNED NOT NULL,  
    distance INT UNSIGNED NOT NULL,  
    FOREIGN KEY(to_station_id) REFERENCES Station(id) ON DELETE CASCADE,  
    FOREIGN KEY(from_station_id) REFERENCES Station(id) ON DELETE CASCADE,  
    PRIMARY KEY(id),  
    UNIQUE(to_station_id,from_station_id)  
)
```

```
CREATE TABLE Company(  
    id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    name varchar(50) NOT NULL,  
    foundation_date DATE NOT NULL,  
    city varchar(59) NOT NULL,  
    PRIMARY KEY(id),  
    INDEX(name)  
)
```

```
CREATE TABLE Company_phone_number(  
    phone_number CHAR(11) NOT NULL,  
    company_id MEDIUMINT UNSIGNED NOT NULL,  
    FOREIGN KEY(company_id) REFERENCES Company(id) ON DELETE CASCADE,  
    PRIMARY KEY(phone_number),  
    INDEX(company_id)  
)
```

```
CREATE TABLE Train(  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    model CHAR(5) NOT NULL,  
    number_of_carriages MEDIUMINT UNSIGNED NOT NULL,  
    capacity MEDIUMINT UNSIGNED NOT NULL,  
    owner_company_id MEDIUMINT UNSIGNED NOT NULL,  
    FOREIGN KEY(owner_company_id) REFERENCES Company(id) ON DELETE CASCADE,  
    PRIMARY KEY(id)  
)
```

```
CREATE TABLE Train_seat(  
    id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    carriage_number TINYINT UNSIGNED NOT NULL,  
    seat_number MEDIUMINT UNSIGNED NOT NULL,  
    class CHAR(1) DEFAULT 'D',  
    train_id INT UNSIGNED NOT NULL,  
    FOREIGN KEY(train_id) REFERENCES Train(id) ON DELETE CASCADE,  
    PRIMARY KEY(id)  
)
```

```
CREATE TABLE Trip(  
    id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    rate TINYINT UNSIGNED NOT NULL,  
    departure_time TIMESTAMP NOT NULL,  
    arrival_time TIMESTAMP NOT NULL,  
    from_station_id MEDIUMINT UNSIGNED NOT NULL,  
    to_station_id MEDIUMINT UNSIGNED NOT NULL,  
    train_id INT UNSIGNED,  
    company_id MEDIUMINT UNSIGNED,  
    FOREIGN KEY(from_station_id) REFERENCES Station(id) ON DELETE CASCADE,  
    FOREIGN KEY(to_station_id) REFERENCES Station(id) ON DELETE CASCADE,  
    FOREIGN KEY(train_id) REFERENCES Train(id) ON DELETE SET NULL,  
    FOREIGN KEY(company_id) REFERENCES Company(id) ON DELETE SET NULL,  
    PRIMARY KEY(id)  
)
```

```
CREATE TABLE Trip_stops(  
    trip_stop_id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    departure_time TIMESTAMP NOT NULL,  
    arrival_time TIMESTAMP NOT NULL,  
    train_track_id BIGINT UNSIGNED NOT NULL,  
    trip_id BIGINT UNSIGNED NOT NULL,  
    FOREIGN KEY(trip_id) REFERENCES Trip(id) ON DELETE CASCADE,  
    FOREIGN KEY(train_track_id) REFERENCES Train_track(id) ON DELETE CASCADE,  
    PRIMARY KEY(trip_stop_id)  
)
```

```
CREATE TABLE Ticket(  
    id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    price INT UNSIGNED NOT NULL,  
    purchase_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
    trip_id BIGINT UNSIGNED,  
    train_seat_id BIGINT UNSIGNED,  
    traveller_id BIGINT UNSIGNED DEFAULT NULL,  
    FOREIGN KEY(trip_id) REFERENCES Trip(id) ON DELETE SET NULL,  
    FOREIGN KEY(train_seat_id) REFERENCES Train_seat(id) ON DELETE SET NULL,  
    FOREIGN KEY(traveller_id) REFERENCES Traveller(id) ON DELETE SET NULL,  
    PRIMARY KEY(id)  
)
```


Trigger

```
CREATE TRIGGER Trip_check BEFORE INSERT ON Trip
FOR EACH ROW
BEGIN
    IF (EXISTS(
        SELECT * FROM Trip AS t
        WHERE ((t.from_station_id=NEW.from_station_id AND t.to_station_id=NEW.to_station_id)
        AND (NEW.departure_time<=t.arrival_time AND NEW.arrival_time>=t.departure_time))))
    THEN
        SIGNAL SQLSTATE '23000'
        SET MESSAGE_TEXT = 'insert cancelled', MYSQL_ERRNO=1052;
    END IF;
END;
```

```
CREATE TRIGGER Trip_stop_check BEFORE INSERT ON Trip_stops
FOR EACH ROW
BEGIN
    IF (EXISTS(
        SELECT * FROM Trip_stops AS ts
        WHERE ((ts.train_track_id=NEW.train_track_id )
        AND (NEW.departure_time<=ts.arrival_time AND NEW.arrival_time>=ts.departure_time))))
    THEN
        SIGNAL SQLSTATE '23000'
        SET MESSAGE_TEXT = 'insert cancelled', MYSQL_ERRNO=1052;
    END IF;
END;
```

Cardinality

Traveller	Trip	Station	Company	Train	Train_seat	Train_track	Trip_stops	Station_phone	ticket
10000	336	200	30	226	70960	270	3272	600	106980

Query

```
SELECT Trip.id,Trip.departure_time AS Trip_departure_time,Trip.arrival_time AS Trip_arrival_time,  
Trip.to_station_id AS Trip_destination_id,dest_station.city AS destination_city,  
Trip_stops.departure_time AS Stop_departure_time,Trip_stops.arrival_time AS Stop_arrival_time,  
Train_track.to_station_id AS Stop_station,  
stop_station.city AS stop_city  
FROM Trip  
INNER JOIN Station AS dest_station ON dest_station.id=Trip.to_station_id  
INNER JOIN Trip_stops ON Trip.id=Trip_stops.trip_id  
INNER JOIN Train_track ON Train_track.id=Trip_stops.train_track_id  
INNER JOIN Station AS stop_station ON Train_track.to_station_id=stop_station.id  
ORDER BY Trip_stops.arrival_time
```

id ▲	Trip_departure_time ▲	Trip_arrival_time ▲	Trip_destination_id ▲	destination_city ▲	Stop_departure_time ▲	Stop_arrival_time ▲	Stop_station ▲	stop_city ▲
110922	2010-01-06 10:24:17	2010-01-07 05:25:17	175	Zaporizhzhia	2010-01-06 10:24:17	2010-01-06 23:17:41	10	Burg
110922	2010-01-06 10:24:17	2010-01-07 05:25:17	175	Zaporizhzhia	2010-01-06 23:17:41	2010-01-07 02:14:03	114	Waterbury
110922	2010-01-06 10:24:17	2010-01-07 05:25:17	175	Zaporizhzhia	2010-01-07 02:14:03	2010-01-07 03:33:11	52	Rajanpur
110922	2010-01-06 10:24:17	2010-01-07 05:25:17	175	Zaporizhzhia	2010-01-07 03:33:11	2010-01-07 03:49:07	137	San Leucio del Sannio
110922	2010-01-06 10:24:17	2010-01-07 05:25:17	175	Zaporizhzhia	2010-01-07 03:49:07	2010-01-07 04:28:22	89	Pamplona
110922	2010-01-06 10:24:17	2010-01-07 05:25:17	175	Zaporizhzhia	2010-01-07 04:28:22	2010-01-07 04:38:00	42	Turriff
110922	2010-01-06 10:24:17	2010-01-07 05:25:17	175	Zaporizhzhia	2010-01-07 04:38:00	2010-01-07 05:25:17	175	Zaporizhzhia

```
SELECT Station.id AS Station_id,Station.city AS city,COUNT(*) AS Stop_number
FROM Trip_stops
INNER JOIN Train_track ON Trip_stops.train_track_id=Train_track.id
INNER JOIN Station ON Station.id=Train_track.to_station_id
GROUP BY Station.id,Station.city ORDER BY Stop_number DESC LIMIT 5
```

Station_id ▲	city ▲	Stop_number ▲
137	San Leucio del Sannio	136
89	Pamplona	132
101	Oryol	118
42	Turriff	115
11	Astrakhan	107

```

SELECT Trip.id AS Trip_id,s1.city AS beginning_city,
s2.city AS destination_city,
SUM(Train_track.distance) AS Total_distance
FROM Trip
INNER JOIN Trip_stops ON Trip.id=Trip_stops.trip_id
INNER JOIN Train_track ON Train_track.id=Trip_stops.train_track_id
INNER JOIN Station AS s2 ON s2.id=Trip.to_station_id
INNER JOIN Station AS s1 ON s1.id=Trip.from_station_id
GROUP BY Trip.id ORDER BY Total_distance DESC LIMIT 5

```

Trip_id ▲	beginning_city ▲	destination_city ▲	Total_distance ▲
111192	Moulins	Elbląg	8088
110781	Güssing	Berlin	7020
110581	Zielona Góra	Shanxi	6706
111176	Chungju	Rajanpur	6230
110500	Luik	Van	6228

```
SELECT Company.id AS Company_id, Company.name AS Company_name , COUNT(*)*100/(
SELECT COUNT(*) FROM Trip
INNER JOIN Company ON Company.id=Trip.company_id
INNER JOIN Ticket ON Ticket.trip_id=Trip.id
WHERE Company.id=Company_id)
AS percentage_of_empty_seats
FROM Trip
INNER JOIN Company ON Trip.company_id=Company.id
INNER JOIN Train ON Train.id=Trip.train_id
INNER JOIN Ticket ON Ticket.trip_id=Trip.id
WHERE Ticket.traveller_id IS NULL
GROUP BY Company.id, Company.name ORDER BY percentage_of_empty_seats DESC LIMIT 5
```


Company_id ▲	Company_name ▲	percentage_of_empty_seats ▲
106	Resource Electronic	2.1443
118	Virtual Star Industries	1.6246
111	Solutions Omega	1.5825
94	Solutions Systems Vision	1.4919
97	Contract East	1.4507

```
SELECT Company.id AS Company_id, Company.name AS Company_name
FROM Company
INNER JOIN Trip ON Trip.company_id=Company.id
WHERE Trip.to_station_id IN
(SELECT Station.id FROM Station
 INNER JOIN Train_track ON Station.id=Train_track.to_station_id
 WHERE Station.city LIKE 'B%g'
 GROUP BY Station.id HAVING COUNT(*)=2
)
GROUP BY Company.id, Company.name HAVING COUNT(*)=
(
 SELECT COUNT(*) FROM (SELECT Station.id FROM Station
 INNER JOIN Train_track ON Station.id=Train_track.to_station_id
 WHERE Station.city LIKE 'B%g'
 GROUP BY Station.id HAVING COUNT(*)=2) AS tmp
)
```

Company_id ▲	Company_name ▲
111	Solutions Omega
118	Virtual Star Industries

Insert

```
import math
import json
import mysql.connector as mysql
import random
import names
import string
import datetime
import threading
from barnum import gen_data
from datetime import date, timedelta
```

```
CONFIG = {
    'user' : 'eyvy',
    'password' : '6174',
    'host' : 'localhost',
    'database' : 'Train'
}
```

```
def multi_insert_traveller(query,data):
```

```
    cnx=mysql.connect(**CONFIG)
    cursor=cnx.cursor()
    cursor.executemany(query,data)
    cnx.commit()
    cnx.close()
```

```
        cursor.execute(query_trip.format(
            rate=rate,
            departure_time=departure_time,
            arrival_time=arrival_time,
            from_station_id=from_station_id,
            to_station_id=to_station_id,
            train_id=train_id,
            company_id=company_id
        ))
    trip_id=cursor.lastrowid
```