Database Basics MS SQL Exam – 24 Jun 2018

Exam problems for the "Database Basics" course @ SoftUni.

Submit your solutions in the SoftUni Judge system at https://judge.softuni.bg/

Trip Service

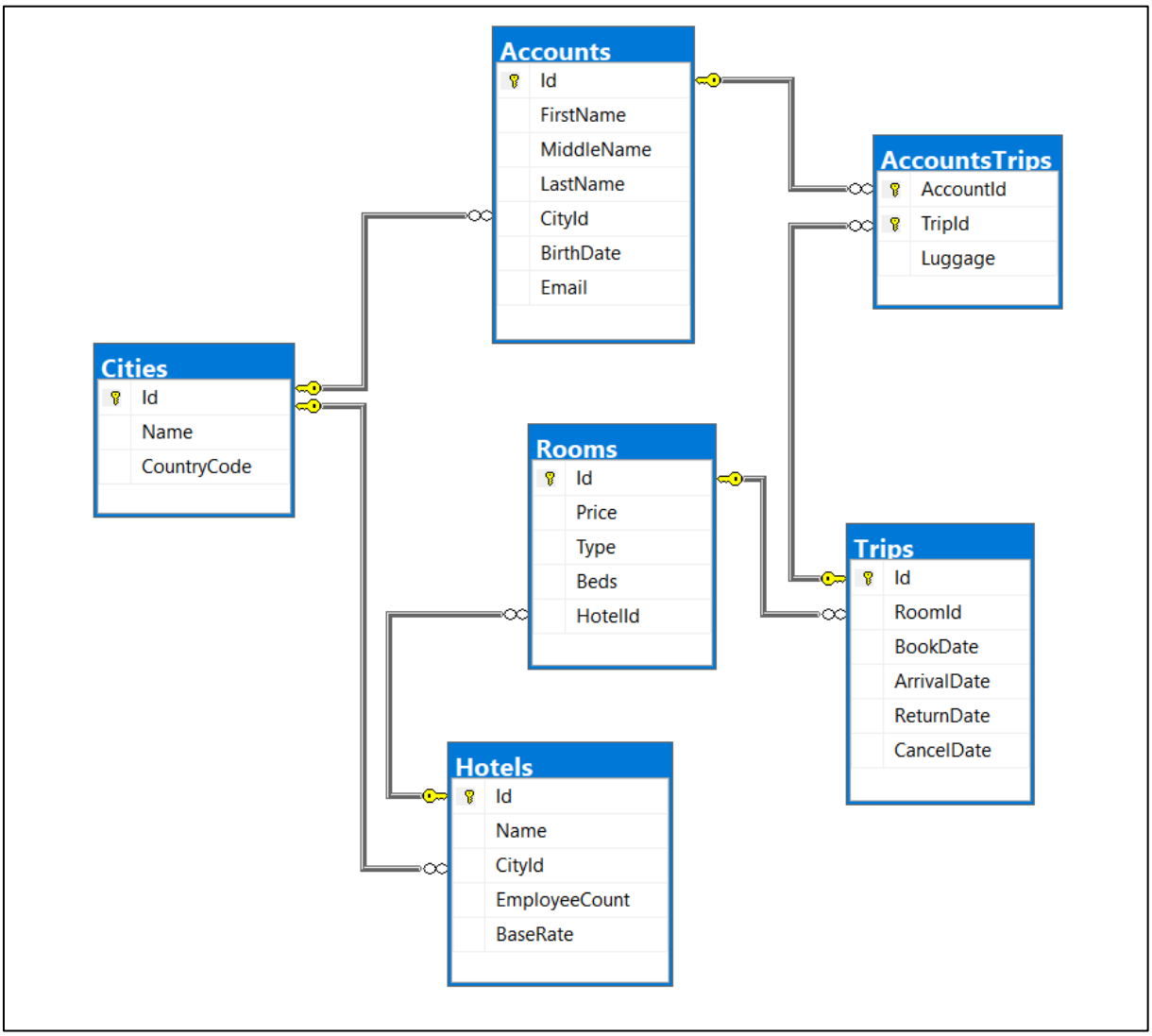
You've been an intern at **Krivodol Trip Service LLC** ever since you finished high school. The Krivodol Trip Service doesn't really pay much, but it's the only trip company within a 50km radius in northwestern Bulgaria.

You've recently been appointed as **Chief Database Engineer**. The Chief Database Engineer's job is to keep track of every single city, account, trip, hotel and hotel room – all on a giant ledger (paper, not blockchain), which has been passed down for the last 3 generations.

Word around the office is that the company is going to hire a few programmers to try and automate the entire process. As you know, the holidays are coming up, and of course, you want to go to the overpopulated, overpriced and overbuilt seaside just to post a couple of photos of your rakia-hardened beach body on your Instagram. So, you decided to **design a relational database** in **SQL Server** and let the new code monkeys take care of everything else.

Section 1. DDL (30 pts)

You are given an E/R Diagram of the Trip Service:



Create a database called **`TripService`**. You need to create **6 tables**:

- **`Cities`** – contains information about cities and their countries.
- **`Hotels`** – contains information about the hotels in the system.
- **`Rooms`** – contains information about the rooms each hotel has.
- **`Trips`** – contains information about each trip.
- **`Accounts`** – contains information about the trip service users.
- **`AccountsTrips`** – contains information about all **accounts** and their **trips**.

Cities

Column Name	Data Type	Constraints
Id	**Integer** from **0** to **2,147,483,647**	Unique table **identificator**, **Identity**
Name	**String** up to 20 symbols, Unicode	**NULL** is **not** allowed
CountryCode	**String** with **exactly 2** symbols	**NULL** is **not** allowed

Hotels

Column Name	Data Type	Constraints
Id	**Integer** from **0** to **2,147,483,647**	Unique table **identificator**, **Identity**
Name	**String** up to 30 symbols, Unicode	**NULL** is **not** allowed
CityId	**Integer** from **0** to **2,147,483,647**	**NULL** is **not** allowed, Relationship with table Cities
EmployeeCount	**Integer** from **0** to **2,147,483,647**	**NULL** is **not** allowed
BaseRate	**Decimal** number with **two-digit** precision	

Rooms

Column Name	Data Type	Constraints
Id	**Integer** from **0** to **2,147,483,647**	Unique table **identificator**, **Identity**
Price	**Decimal** number with **two-digit** precision	**NULL** is **not** allowed
Type	**String** up to 20 symbols, Unicode	**NULL** is **not** allowed
Beds	**Integer** from **0** to **2,147,483,647**	**NULL** is **not** allowed
HotelId	**Integer** from **0** to **2,147,483,647**	**NULL** is **not** allowed, Relationship with table Hotels

Trips

Column Name	Data Type	Constraints
Id	**Integer** from **0** to **2,147,483,647**	Unique table **identificator**, **Identity**
RoomId	**Integer** from **0** to **2,147,483,647**	**NULL** is **not** allowed, Relationship with table Rooms
BookDate	**Date**	**NULL** is **not** allowed, **must be before ArrivalDate**
ArrivalDate	**Date**	**NULL** is **not** allowed, **must be before ReturnDate**
ReturnDate	**Date**	**NULL** is **not** allowed
CancelDate	**Date**	

Column Name	Data Type	Constraints
Id	**Integer** from **0** to **2,147,483,647**	Unique table **identificator**, **Identity**
FirstName	**String** up to 50 symbols, Unicode	**NULL** is **not** allowed
MiddleName	**String** up to 20 symbols, Unicode	
LastName	**String** up to 50 symbols, Unicode	**NULL** is **not** allowed
CityId	**Integer** from **0** to **2,147,483,647**	**NULL** is **not** allowed, Relationship with table Cities
BirthDate	**Date**	**NULL** is **not** allowed
Email	**String** up to 100 symbols	**NULL** is **not** allowed, **Unique**

Accounts

Column Name	Data Type	Constraints
AccountId	**Integer** from **0** to **2,147,483,647**	**NULL** is **not** allowed, Relationship with table Accounts
TripId	**Integer** from **0** to **2,147,483,647**	**NULL** is **not** allowed, Relationship with table Trips
Luggage	**Integer** from **0** to **2,147,483,647**	**NULL** is **not** allowed, must be **at least** 0

AccountsTrips

1. Database design

Submit all of yours **create statements** to Judge (only creation of tables).

Section 2. DML (10 pts)

Before you start, you must import "`DataSet-TripService.sql`". If you have created the structure correctly, the data should be successfully inserted without any errors.

In this section, you have to do some data manipulations:

2. Insert

Insert some sample data into the database. Write a query to add the following records into the corresponding tables. **All Ids should be auto-generated**.

Accounts

FirstName	MiddleName	LastName	CityId	BirthDate	Email
John	Smith	Smith	34	1975-07-21	j_smith@gmail.com
Gosho	NULL	Petrov	11	1978-05-16	g_petrov@gmail.com
Ivan	Petrovich	Pavlov	59	1849-09-26	i_pavlov@softuni.bg
Friedrich	Wilhelm	Nietzsche	2	1844-10-15	f_nietzsche@softuni.bg

Trips

RoomId	BookDate	ArrivalDate	ReturnDate	CancelDate
101	2015-04-12	2015-04-14	2015-04-20	2015-02-02
102	2015-07-07	2015-07-15	2015-07-22	2015-04-29
103	2013-07-17	2013-07-23	2013-07-24	NULL
104	2012-03-17	2012-03-31	2012-04-01	2012-01-10
109	2017-08-07	2017-08-28	2017-08-29	NULL

3. Update

Make all rooms' prices **14% more expensive** where the **hotel ID** is either **5**, **7** or **9**.

4. Delete

Delete **all** of Account ID **47**'s **account's trips** from the mapping table.

Section 3. Querying (40 pts)

You need to start with a fresh dataset, so recreate your DB and import the sample data again (DataSet-TripService.sql).

5. Bulgarian Cities

Select all **cities** in **Bulgaria**. Order them by **city name**.

Examples

Id	Name

15	Blagoevgrad
4	Burgas
8	Dobrich
18	Gabrovo
…	…

6. People Born After 1991

Select all **full names** and **birth years** from accounts, who are born after **1991**.

Order them by **birth year (descending)**, then by first name **(ascending)**. Keep in mind that middle names can be **NULL** 😊

Examples

Full Name	BirthYear
Claudia Keely Lotze	1994
Jourdan Marketa Fawcitt	1994
Nealson Waiter Villalta	1994
Palm Van	1994
Xever Leoine Santi	1994
Cornell Alidia Grieg	1993
…	…

7. EEE-Mails

Select **accounts** whose **emails start** with the **letter** "**e**". Select their **first and last name**, their **birthdate** in the format "**`MM-dd-yyyy`**", and their **city name**.

Order them by **city name (descending)**

Examples

FirstName	LastName	BirthDate	Hometown	Email
Evvie	Covolini	01-11-1979	Wolverhampton	e_covolini@softuni.bg
Eward	Prigg	12-10-1982	Shumen	e_prigg@gmail.com
Eadith	Gull	03-03-1983	Haskovo	e_gull@outlook.com
Edgardo	Slessar	12-29-1983	Glasgow	e_slessar@outlook.com

8. City Statistics

Select all **cities** with the **count of hotels** in them. Order them by the **hotel count (descending)**, then by **city name**. Include cities, which have **no hotels** in them as well.

Examples

City	Hotels

Belfast	11
Cardiff	11
Chelyabinsk	11
Phoenix	11
San Francisco	11
Seattle	11
Veliko Tarnovo	11
Houston	10
…	…

9. Expensive First-Class Rooms

Find all First-Class rooms and select the **Id**, **Price**, **Hotel name** and **City name**.

Order them by **Price (descending)**, then by **Room ID**.

Examples

Id	Price	Hotel	City
51	90.90	Recreation Hotel & Spa	Ruse
863	90.90	Exalted Resort & Spa	Volgograd
427	90.20	Stargaze Hotel & Spa	New York
727	90.20	Enterprise Hotel	Yekaterinburg
777	90.00	Nimbus Hotel & Spa	Chelyabinsk
729	89.90	History Resort	Yekaterinburg
512	89.70	Nimbus Hotel & Spa	San Diego
901	89.70	Diorama Resort	Tyumen
116	89.50	Holiday Resort	Blagoevgrad
…	…	…	…

10. Longest and Shortest Trips

Find the **longest** and **shortest trip** for each **account**, in **days**. Filter the results to **accounts** with **no middle name** and **trips, which aren't cancelled** (**CancelDate** is **null**).

Order the results by **Longest Trip days** (**descending),** then by **Account ID.**

Examples

AccountId	FullName	LongestTrip	ShortestTrip
40	Winna Maisey	7	1
47	Evvie Covolini	7	2
56	Tillie Windress	7	1
57	Eadith Gull	7	1
…	…	…	…

11. Metropolis

Find the **top 5** cities, which have the most registered accounts in them. Order them by the **count of accounts (descending)**.

Examples

Id	City	Country	Accounts
76	Tyumen	RU	5
12	Haskovo	BG	4
33	Belfast	UK	4
…	…	…	…

12. Romantic Getaways

Find all accounts, which have had **one or more** trips to **a hotel in their hometown**.

Order them by the **trips count (descending)**, then by **Account ID**.

Examples

Id	Email	City	Trips
50	t_joules@mail.com	New York	2
19	m_stango@yahoo.com	Burgas	1
48	n_revitt@softuni.bg	Bradford	1
…	…	…	…

13. Lucrative Destinations

Find the **top 10** cities' **Total Revenue Sum** (**Hotel Base Rate + Room Price**) and **trip count**.

Count only trips, which were **booked in 2016**.

Order them by **Total Revenue (descending)**, then by **Trip count (descending)**

Examples

Id	Name	Total Revenue	Trips
56	Seattle	795.80	6
47	Dallas	649.40	6
17	Vratsa	536.60	4
28	Cardiff	504.30	4
65	Chelyabinsk	386.30	3
…	…	…	…

14. Trip Revenues

Find all trips' **revenue (hotel base rate + room price)**. If a trip is **canceled**, its **revenue** is **always 0**. Extract the **trip's ID**, the **hotel's name**, the **room type** and the **revenue**.

Order the results by **Room type**, then by the **Trip ID**.

Examples

Id	HotelName	RoomType	Revenue
9	Cloud Resort	Economy	51.10
14	Lethargy Hotel & Spa	Economy	39.90
43	Courtyard Hotel	Economy	82.20
49	Ranch Hotel	Economy	0.00
…	…	…	…

15. Top Travelers

Find the **top traveler** for **each country**. The top traveler is the **account**, which has the **most trips** to that **country**.

Order the results by the **count of trips (descending)**, then by **Account ID**.

Examples

AccountId	Email	CountryCode	Trips
80	a_flucks@gmail.com	RU	6
14	t_ludwikiewicz@outlook.com	UK	5
32	a_roskell@softuni.bg	US	5
…	…	…	…

16. Luggage Fees

Apart from its base rate and room price, each hotel also has a hidden “luggage fee”. It’s in the terms and conditions, but nobody reads those…

The luggage fee only comes into action if a trip has **more than 5 items of luggage** and it’s equal to the **number of luggage items, multiplied by 5**.

Take into account only trips, which have **more than 0** luggage.

Order the results by the **count of luggage (descending)**

Examples

TripId	Luggage	Fee
632	7	$35
617	6	$30
833	6	$30
264	6	$30
273	6	$30
306	6	$30
323	6	$30
330	6	$30
428	6	$30
457	6	$30
405	5	$0
…	…	…

17. GDPR Violation

Retrieve the following information about each trip:

- Trip ID
- Account Full Name
- From – Account hometown
- To – Hotel city
- Duration – the **duration** between the **arrival date** and **return date** in **days**. If a trip is cancelled, the value is "**`Canceled`**"

Order the results by **full name**, then by **Trip ID**.

Examples

Id	Full Name	From	To	Duration
273	Adah Douglass Lathaye	Stara Zagora	Cardiff	Canceled
491	Adah Douglass Lathaye	Stara Zagora	Houston	4 days
776	Adah Douglass Lathaye	Stara Zagora	Chelyabinsk	3 days
133	Allissa Rickey Gigg	Austin	Veliko Tarnovo	6 days
…	…	…	…	…

Section 4. Programmability (14 pts)

18. Available Room

Create a **user defined function**, named **`udf_GetAvailableRoom(@HotelId, @Date, @People)`**, that receives a **hotel ID**, a desired **date**, and the count of **people** that are going to be signing up.

The total price of the room can be calculated by using this formula:

- **`(HotelBaseRate + RoomPrice) * PeopleCount`**

The function should find a suitable room in the provided hotel, based on these conditions:

- The room must **not be already occupied**. A room is occupied if the **date** the customers want to book is **between** the **arrival** and **return dates** of a trip to that room and the trip is **not canceled**.
- The room must be **in** the provided **hotel**.
- The room must have enough **beds** for all the **people**.

If any rooms in the desired hotel **satisfy** the customers' conditions, find the **highest priced room (by total price)** of all of them and provide them with that room.

The function must return a **message** in the **format**:

- "**`Room {Room Id}: {Room Type} ({Beds} beds) - ${Total Price}`**"

If no room could be found, the function should return "**`No rooms available`**".

Example:

Query
`SELECT dbo.udf_GetAvailableRoom(112, '2011-12-17', 2)`

Output
`Room 211: First Class (5 beds) - $202.80`

Query
`SELECT dbo.udf_GetAvailableRoom(94, '2015-07-26', 3)`
Output
`No rooms available`

19. Switch Room

Create a **user defined stored procedure**, named **`usp_SwitchRoom(@TripId, @TargetRoomId)`**, that receives a **trip** and a **target room**, and attempts to **move** the **trip** to **the target room**. A room will only be switched if all of these conditions are true:

- If the **target room** ID is in **a different hotel**, than the **trip** is in, **raise an exception** with the message "**`Target room is in another hotel!`**".
- If the **target room** doesn't have **enough beds** for all the **trip's accounts**, **raise an exception** with the message "**`Not enough beds in target room!`**".

If all the above conditions pass, **change** the **trip's room ID** to the **target room ID**.

Example usage:

Query	Output
`EXEC usp_SwitchRoom 10, 11` `SELECT RoomId FROM Trips WHERE Id = 10`	`11`
`EXEC usp_SwitchRoom 10, 7`	`Target room is in another hotel!`
`EXEC usp_SwitchRoom 10, 8`	`Not enough beds in target room!`

Section 5. Bonus (6 pts)

20. Cancel Trip

Create a **trigger**, which fires **when a trip is deleted**. Instead of deleting a trip, **set its cancel date to the current date** and **IGNORE** trips, which have **already been canceled**.

Example usage:

Query
`DELETE FROM Trips` `WHERE Id IN (48, 49, 50)`
Response

`(2 rows affected)`
 	
`(3 rows affected)`