# Database Basics (MSSQL) Exam – 28 June 2020

Submit your solutions in the SoftUni Judge system at [Judge](https://judge.softuni.org/Contests/2483/Databases-MSSQL-Server-Exam-28-Jun-2020)

# Colonial Journey

2000 years from now, the known space is colonized by the human race. However, the four Citadel Council races are planning to populate new home worlds in the SoftUnia Galaxy as part of a strategy called the SoftUnia Initiative. 20000 citizens are sent aboard space transportation vessels. The Council has asked you to create a Colonization Management system so they can keep track of the colonists' journeys trough the stars.

# Section 1. DDL (30 pts)

You have given an Entity / Relationship Diagram of the CJMS Database:



The **ColonialJourney** Database holds information about colonists, their travel cards, information about the journeys, types of space vessels and destination planets. Your task is to create a database called **ColonialJourney**. Then you will have to create several **tables**.

* **Planets** – contains information about **planets**;
* **Spaceports** – contains information about **space ports**;
* **Spaceships –** contains information about **space ships**;
* **Colonists –** contains information about **colonists**;
* **Journeys –** contains information about **journeys**;
* **TravelCards –** contains information about **travel cards**.

**NOTE: Please keep in mind that in case you have to work with a date, you have to use the exact same data type, described in the models tables. For example, data type Date means that you have to use Date, DateTime means that you have to use DateTime. If you don't use the correct type, the Judge system won't accept your submission as correct.**

**NOTE: Keep in mind that Judge doesn't accept "ALTER" statement and square brackets naming (when the names are not keywords).**

You have been tasked to create the tables in the database by the following models:

### Planets

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| Id | **Integer** from **1** to **2,147,483,647** | Unique table **identificator**, **Identity** |
| Name | **String** up to **30** symbols, non-unicode | **NULL** is **not** allowed |

### Spaceports

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| Id | **Integer** from **1** to **2,147,483,647** | Unique table **identificator**, **Identity** |
| Name | **String** up to **50** symbols, non-unicode | **NULL** is **not** allowed |
| PlanetId | **Integer,** from **1** to **2,147,483,647.** | **NULL** is **not** allowed, Relationship with table **Planets** |

### Spaceships

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| Id | **Integer** from **1** to **2,147,483,647** | Unique table **identificator**, **Identity** |
| Name | **String** up to **50** symbols, non-unicode | **NULL** is **not** allowed |
| Manufacturer | **String** up to **30** symbols, non-unicode | **NULL** is **not** allowed |
| LightSpeedRate | **Integer** from **1** to **2,147,483,647** | Has a **default value** of **0** |

### Colonists

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| Id | **Integer** from **1** to **2,147,483,647** | Unique table **identificator**, **Identity** |
| FirstName | **String** up to **20** symbols, non-unicode | **NULL** is **not** allowed |
| LastName | **String** up to **20** symbols, non-unicode | **NULL** is **not** allowed |
| Ucn | **String** up to **10** symbols, non-unicode | **NULL** is **not** allowed **UNIQUE** values. |
| BirthDate | **Date** | **NULL** is **not** allowed |

### Journeys

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| Id | **Integer** from **1** to **2,147,483,647** | Unique table **identificator**, **Identity** |
| JourneyStart | **DateTime** | **NULL** is **not** allowed |
| JourneyEnd | **DateTime** | **NULL** is **not** allowed |
| Purpose | **String** up to **11** symbols, non-unicode | Should **only** contain one of the following purposes: "**Medical"**, "**Technical"**, "**Educational**", "**Military"**.  **NULL** is **allowed** |
| DestinationSpaceportId | **Integer** from **1** to **2,147,483,647** | **NULL** is **not** allowed, Relationship with table **Spaceports** |
| SpaceshipId | **Integer** from **1** to **2,147,483,647** | **NULL** is **not** allowed, Relationship with table **Spaceships** |

### TravelCards

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| Id | **Integer** from **1** to **2,147,483,647** | Unique table **identificator**, **Identity** |
| CardNumber | **String** exactly to **10** symbols, non-unicode | **NULL** is **not** allowed **UNIQUE** values. |
| JobDuringJourney | **String** up to **8** symbols, non-unicode | Should **only** contain one of the following jobs: “**Pilot**”, “**Engineer**”, “**Trooper**”, “**Cleaner**”, “**Cook**”.  **NULL** is **allowed** |
| ColonistId | **Integer** from **1** to **2,147,483,647** | **NULL** is **not** allowed, Relationship with table Colonists |
| JourneyId | **Integer** from **1** to **2,147,483,647** | **NULL** is **not** allowed, Relationship with table Journeys |

## Database Design

Submit all of your **CREATE** **statements** to the **Judge** system.

# Section 2. DML

**Before you start, you must import "**DataSet-ColonialJourney.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:

## Insert

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

**Planets**

|  |
| --- |
| **Name** |
| Mars |
| Earth |
| Jupiter |
| Saturn |

**Spaceships**

|  |  |  |
| --- | --- | --- |
| **Name** | **Manufacturer** | **LightSpeedRate** |
| Golf | VW | 3 |
| WakaWaka | Wakanda | 4 |
| Falcon9 | SpaceX | 1 |
| Bed | Vidolov | 6 |

## Update

Update all spaceships light speed rate with **1** where the **Id** is between **8** and **12.**

## Delete

Delete first three inserted **Journeys** (be careful with the relationships).

# Section 3. Querying

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

## Select all military journeys

Extract from the database, all **Military** journeys in the format "dd-MM-yyyy". Sort the results **ascending** by **journey start.**

### Required Columns

* **Id**
* **JourneyStart**
* **JourneyEnd**

### Example

|  |  |  |
| --- | --- | --- |
| **Id** | **JourneyStart** | **JourneyEnd** |
| 7 | 04/01/2019 | 09/12/2049 |
| 3 | 21/02/2019 | 03/01/2049 |
| ... | ... | ... |

## Select all pilots

Extract from the database all colonists, which have a **pilot job.** Sort the result by **id, ascending.**

### Required Columns

* **Id**
* **FullName**

### Example

|  |  |
| --- | --- |
| **Id** | **full\_name** |
| 6 | Clark Cowan |
| 18 | Wald Bim |
| ... | ... |

## Count colonists

Count all colonists that are on **technical journey.**

### Required Columns

* **Count**

### Example

|  |
| --- |
| **count** |
| 16 |

## Select spaceships with pilots younger than 30 years

Extract from the database those **spaceships**, **which have pilots**, **younger** than **30 years old**. In other words, **30 years from 01/01/2019**. Sort the results **alphabetically** by spaceship **name**.

### Required Columns

* **Name**
* **Manufacturer**

### Example

|  |  |
| --- | --- |
| **Name** | **Manufacturer** |
| Anarchy | Fivebridge |
| ... | ... |

## Select all planets and their journey count

Extract from the database all **planets' names** and their **journeys count**. Order the results by journeys **count**, **descending** and by **planet name ascending**.

### Required Columns

* **PlanetName**
* **JourneysCount**

### Example

|  |  |
| --- | --- |
| **PlanetName** | **JourneysCount** |
| Otroyphus | 4 |
| Eipra | 2 |
| ... | ... |

## Select Second Oldest Important Colonist

Find all colonists and their job during journey with rank **2**. Keep in mind that all the selected colonists with rank 2 must be the oldest ones. You can use ranking over their job during their journey.

### Required Columns

* **JobDuringJourney**
* **FullName**
* **JobRank**

### Example

|  |  |  |
| --- | --- | --- |
| **JobDuringJourney** | **FullName** | **JobRank** |
| Cleaner | Hale O'Doireidh | 2 |
| Cook | Laurie Askin | 2 |
| … | … | … |

# Section 4. Programmability

## Get Colonists Count

Create a **user defined function** with the name **dbo.udf\_GetColonistsCount(PlanetName VARCHAR (30))** that receives **planet name** and returns the count of all colonists sent to that planet.

### Example

|  |
| --- |
| **Query** |
| SELECT dbo.udf\_GetColonistsCount('Otroyphus') |
| Output |
| 35 |

## Change Journey Purpose

Create a **user defined stored procedure**, named **usp\_ChangeJourneyPurpose(@JourneyId, @NewPurpose)**, that receives a **journey id** and **purpose**, and attempts to **change the purpose of that journey**. An purpose will only be changed if all of these conditions **pass**:

* If the **journey id** doesn't exist, then it **cannot be changed.** **Raise an error** with the message "The journey does not exist!"
* If the **journey** has already that purpose, **raise an error** with the message "You cannot change the purpose!"

If all the above conditions pass, **change the purpose of that journey**.

### Example

|  |  |
| --- | --- |
| **Query** | **Output** |
| EXEC usp\_ChangeJourneyPurpose 4, 'Technical' |  |
| EXEC usp\_ChangeJourneyPurpose 2, 'Educational' | You cannot change the purpose! |
| EXEC usp\_ChangeJourneyPurpose 196, 'Technical' | The journey does not exist! |