

Database Fundamentals MSSQL Retake – The Nerd Herd

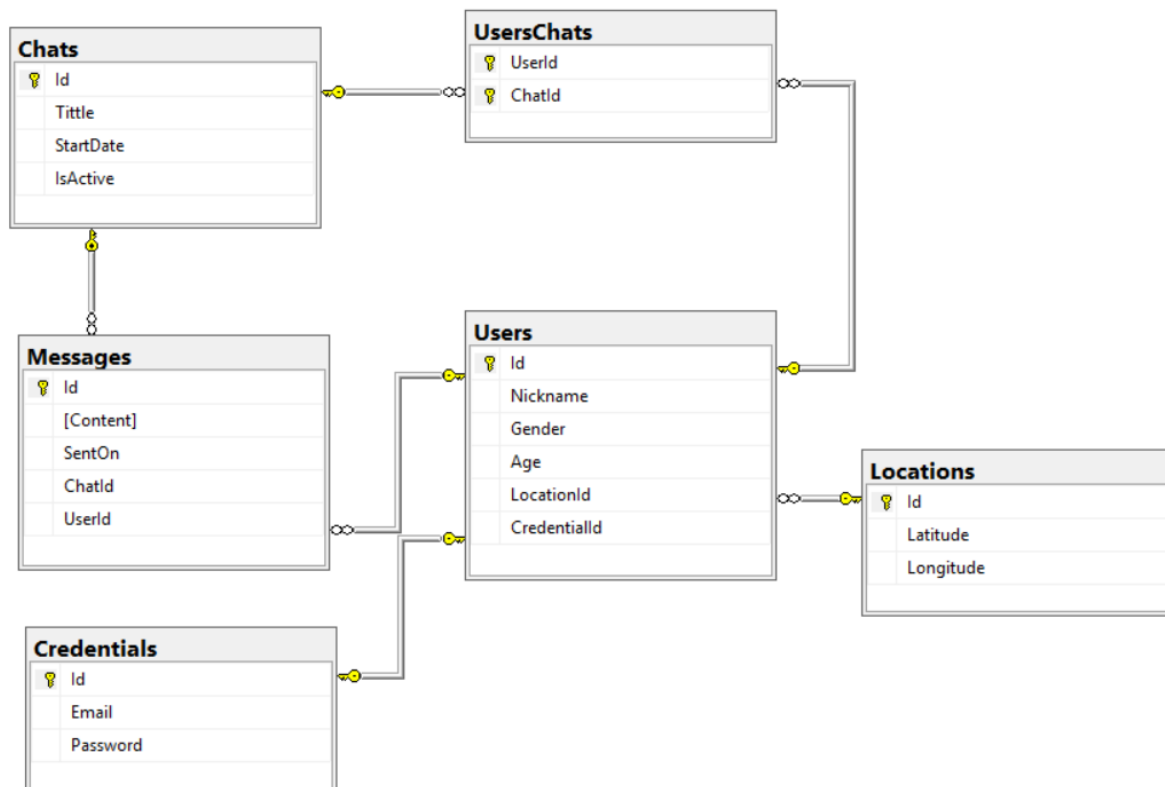
That is the name of the newest super-hot awesome chat application! You are part of the developing team and as a such you are supposed to create the database underneath. Then you have to do some data manipulations. Finally, you have to do some work on the programmability part. Here are some more details:

Section 1. DDL

25 pts

For this section put your queries in judge and use SQL Server run queries and check DB.

You have been given the E/R Diagram of the chat:



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

- Users – contains information about the users who can chat. Each user has location and **exactly one** credential.
- Credentials – contains information about the login details.
- Location – contains geographical information about the location of each person
- Chats – contains information about the chat. Chats can have many users in them.
- UsersChats – mapping table between users and chats
- Messages – contains information about the sent messages.

Users

Column Name	Data Type	Constraints
Id	Integer from 0 to 2,147,483,647	Unique table identifier, Identity
Nickname	String up to 25 symbols	
Gender	Character with 1 symbol	
Age	Integer from 0 to 2,147,483,647	
Location_id	Integer from 0 to 2,147,483,647	Relationship with locations
Credential_id	Integer from 0 to 2,147,483,647	Relationship with table credentials, Unique values

Locations

Column Name	Data Type	Constraints
Id	Integer from 0 to 2,147,483,647	Unique table identifier
Latitude	Floating point number	
Longitude	Floating point number	

Credentials

Column Name	Data Type	Constraints
Id	Integer from 0 to 2,147,483,647	Unique table identifier
Email	String up to 30 symbols	
Password	String up to 20 symbols	

Chats

Column Name	Data Type	Constraints
Id	Integer from 0 to 2,147,483,647	Unique table identifier
Title	String up to 32 symbols	
StartDate	Date without time	
IsActive	Bit	

Messages

Column Name	Data Type	Constraints
Id	Integer from 0 to 2,147,483,647	Unique table identifier
Content	String up to 200 symbols	

SentOn	Date without time	
ChatId	Integer from 0 to 2,147,483,647	Relationship with table chats
UserId	Integer from 0 to 2,147,483,647	Relationship with table users

UsersChats

Column Name	Data Type	Constraints
UserId	Integer from 0 to 2,147,483,647	Unique table identifier, Relationship with table users
ChatId	Integer from 0 to 2,147,483,647	Unique table identifier, Relationship with table chats

1. Database design

Send all yours create statements in judge.

Section 2. DML

15 pts

For this section put your queries in judge and use SQL Server run skeleton, run queries and check DB.

Before you start you have to import [Data.sql](#). If you have created the structure correctly the data should be successfully inserted.

In this section you have to do couple of data manipulations:

2. Insert

Do you remember ASL? It stands for Age, Sex(gender), Location. You have to insert couple of message based on table **Users**.

- Content – it should be concatenation of age, gender, latitude and longitude split by **dash**.
Use a concatenating function.
- SentOn – should be the current date
- ChatId –
 - If the user is **female multiply the age by 2 and then take the square root**
 - if it is a **male divide the age by 18 and take it the power of 3**
 - **when you find the id round it up**
- UserId – take the UserId from the table
- You should insert data for users with id between 10 and 20 inclusively

Example:

Content	SentOn	ChatId	UserId
56-F-10.1857-123.734	2016-12-15	11	12

3. Update

The back-end developers have slightly failed and let some chats to have messages with a date earlier than the creation date of the chat. You have to fix that. For all chats which have messages before the chat StartDate **update the chat StartDate to be equal to the earliest message in that chat.**

Example:

ChatId	StartDate	MessageId	SentOn
4	2013-10-28	53	2011-06-22

ChatId	StartDate	MessageId	SentOn
4	2011-06-22	53	2011-06-22

4. Delete

Delete all locations which doesn't have user located there.

Section 3. Querying

40 pts

For this section put your queries in judge and use SQL Server prepare DB and run queries.

5. Age Range

Select all users that are aged between 22 and 37 inclusively.

Required columns:

- Nickname
- Gender
- Age

Example:

Nickname	Gender	Age
sbell0	F	23

6. Messages

Select all messages that are sent after 12.05.2014 and contain the word "just". Sort the results by the message id in descending order.

Required columns:

- Content
- SentOn

Example:

Content	SentOn
---------	--------

odio cras mi pede malesuada in imperdiet et commodo vulputate justo in	2014-07-30
--	------------

7. Chats

Select all chats that are active and their title length is less than 5 **or** 3rd and 4th letters are equal to "tl". Sort the results by title in descending order.

Required columns:

- Title
- IsActive

Example:

Title	IsActive
Viva	0

8. Chat Messages

Select all chats with messages sent before 26.03.2012 and chat title with last letter equal to "x". Sort by chat Id and message Id in ascending order.

Required columns:

- Id(Chats)
- Tittle
- Id(Messages)

Example:

Id	Tittle	Id
22	Quo Lux	48

9. Message Count

Select all chats and the amount of messages they have. Some messages may not have a chat. Filter messages with id less than 90. Select only the first 5 results sorted by TotalMessages in descending order and chat id in ascending order.

Required columns:

- Id(Chats)
- TotalMessages

Example:

Id	TotalMessages
37	4

10. Credentials

Select all users with emails ending with "co.uk". Sort by email in ascending order.

Required columns:

- Nickname
- Email
- Password

Example:

Nickname	Email	Password
vburkek	mperkinst@amazon.co.uk	ICFO0hSeRt

11. Locations

Select all users who don't have a location.

Required columns:

- Id(Users)
- Nickname
- Age

Example:

Id	Nickname	Age
1	ahunta	63

12. Left Users

Select all messages sent from users who have left the chat (they are not in the chat anymore). Filter data only for chat with id 17. Sort by message Id in descending order.

Required columns:

- Id(Messages)
- ChatId
- UserId

Example:

Id	ChatId	UserId
65	17	24

13. Users in Bulgaria

Select all users that are located in Bulgaria. Consider the latitude is in range [41.14;44.13] and longitude in range [22.21; 28.36]. Sort the results by title in ascending order.

Required columns:

- Nickname
- Title
- Latitude
- Longitude

Example:

Nickname	Title	Latitude	Longitude
slittle1	Lotlux	42.09028	25.03239

14. Last Chat

Select the first message (if there is any) of the last chat.

Required columns:

- Title
- Content

Example:

Title	Content
Bigtax	NULL

Section 4. Programmability

20 pts

For this section put your queries in judge and use SQL Server run skeleton, run queries and check DB.

15. Radians

Create a user defined function that transforms degrees to radians. The formula should multiply the degrees by Pi and then split by 180. The return type must be float. Call the function **udf_GetRadians**.

Parameters:

- Degrees

Example:

```
SELECT dbo.udf_GetRadians(22.12) AS Radians
```

Radians
0.386066830541146

16. Change Password

Create a user defined procedure that receives an email and changes the password with the newly provided one. If the email doesn't exist throw an exception with Severity = 16, State = 1 and message "The email doesn't exist!". Call the procedure **udp_ChangePassword**.

Parameters:

- Email
- NewPassword

Example:

Email	Password
abarnes0@sogou.com	LOL77s

```
exec udp_ChangePassword 'abarnes0@sogou.com', 'new_pass'
```

Email	Password
abarnes0@sogou.com	new_pass

17. Send Message

Create a user defined procedure sends a message with a current date. The procedure should receive UserId, ChatId and the Content of the message. If there is no chat with that user throw an exception with Severity = 16, State = 1 and message "There is no chat with that user!". Call the procedure **udp_SendMessage**.

Parameters:

- UserId
- ChatId
- Content

Example:

```
EXEC dbo.udp_SendMessage 19, 17, 'Awesome'
```

Content	SentOn	ChatId	UserId
Awesome	2016-12-15	17	19

18. Log Messages

Create a trigger that logs any deleted message from table messages. **Submit only your create trigger statement.** The log table should be called **MessageLogs** and should have exactly the same structure as table **Messages**. The name of the trigger is not important.

Example:

```
DELETE FROM [Messages]  
WHERE [Messages].Id = 1
```

MessageLogs

Id	Content	SentOn	ChatId	UserId
1	in faucibus orci luctus et ultrices posuere cubilia curae mauris	2016-11-18	49	20

Section 5. Bonus

10 pts

For this section put your queries in judge and use SQL Server run skeleton, run queries and check DB.

19. Delete users

Create a trigger that will help you to delete a user. **Submit the create trigger statement only.**

Example:

```
DELETE FROM Users  
WHERE Users.Id = 1
```

(3 row(s) affected)