Social Media Design for QuietThunder

Austin Johns, Jared Wermager, David Kiesel

Grand Canyon University

April 19, 2019

Abstract

QuietThunder is a social media network designed to connect people to express individuality in an environment that brings the world together. The database design for this social media network is designed to be simple and concise. A simple design permits clarity for users and designers alike. The purpose of this database is to securely store information that can be accessed by the user whenever they would like. The goal of our database design is to create a social media experience like no other. We wanted this experience to flow smoothly and connect people easily. Many competitors have created networks that are much too complicated, making the user’s experience less positive. A clear methodology is required to achieve this goal. To create this database, we wanted to design user profiles so that the users can view and message each other. Basic information is stored to give an overview of any individual’s information. In this document, we will review the process of creating this database, including, the design goals, conceptual design, logical design, schemas, and application.

Table of Contents

Introduction……………………………………………………………………………………. 1

Project Design Goal…………………………………………………………………………… 2

Requirements Analysis…………………………………………………………………………. 3

Logiacl Database Design………………………………………………………………………. 4

Application and Security Design………………………………………………………………. 5

Introduction

To begin designing this database, we started with the most crucial part, the user. The user profile includes a username, first name, last name, and age. In their profile, the user has a status, which includes their activity, marital status, permissions, country, and any linked account, such as their Facebook profile. To help connect people, users are able to post. In the system, a user’s post is linked to their profile so that people can view it easily. Each user can have relationships with other users. Users can create page themes and send messages.

Project Design Goal

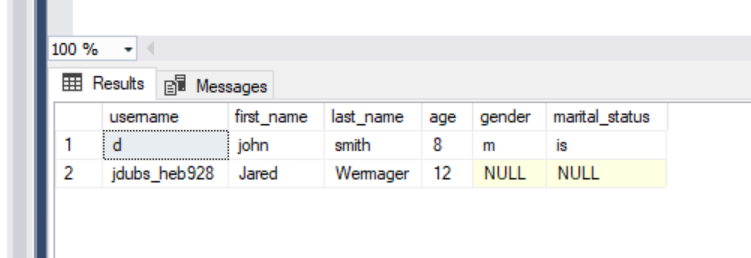
Each explanation from the introduction piece of this presentation will have its own table to house the data from the user. There will be tables called user\_profile, user\_status, relationships, posts, page\_themes, and messaging. Each of these tables acts as a storage center for the data provided by the user. These tables are not just for storage, but for acquiring data by any given user with proper permissions. A foreign key will also be created on username in the table, “user\_status,” and it will be a constraint that prohibits the input of a new user in user status unless that user is already in the user\_profile table. For example, if a designer tried to add data for a user that doesn’t exist yet, the system will provide an error and ask the designer to first create a profile for the user.

Requirement Analysis

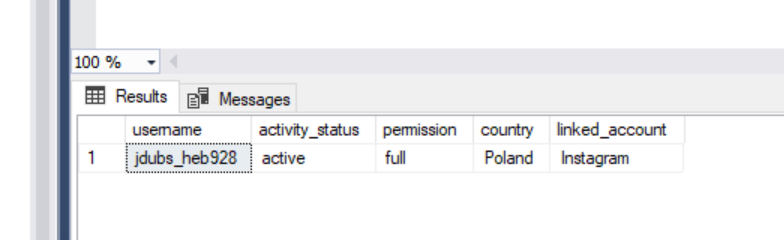
A main requirement of this Final CLC Project is that each individual team member participates evenly and shows beneficial work ethic. This is an important attribute to our project because each person on our team offers a unique perspective. This makes our database easier to use for many types of people that think in different ways. Jared, Austin, and David committed equal amounts of time and work to this project.

Logical Database Design

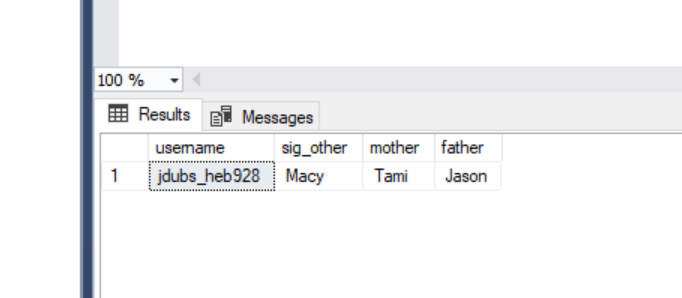
select \* from user\_profile



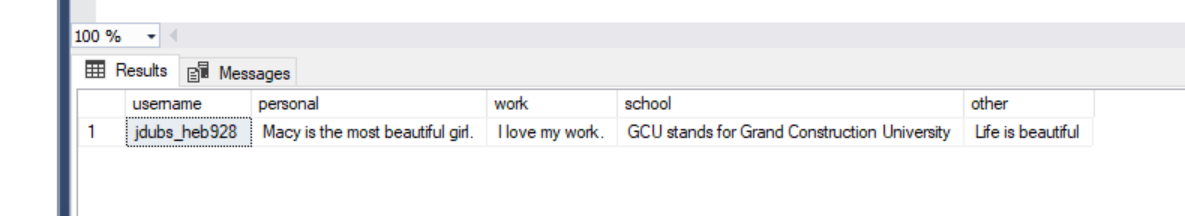
select \* from user\_status



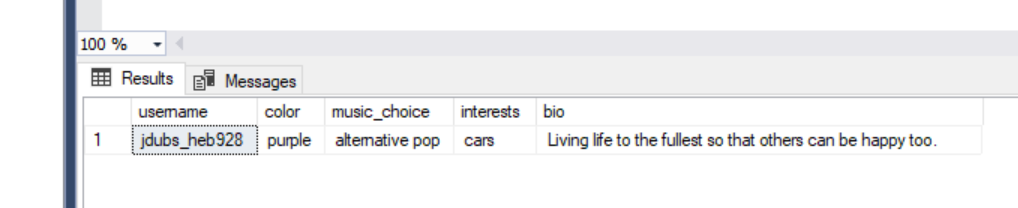
select \* from relationships



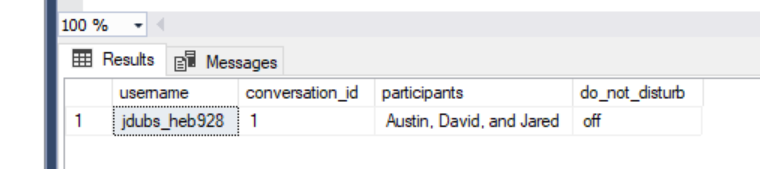
select \* from posts



select \* from page\_themes



select \* from messaging



USE [quiet\_thunder]

GO

DECLARE @return\_value int

EXEC @return\_value = [dbo].[MasterInsUpDel]

@username = N'aajjohns',

@first\_name = N'austin',

@last\_name = N'johns',

@age = 20,

@gender = N'm',

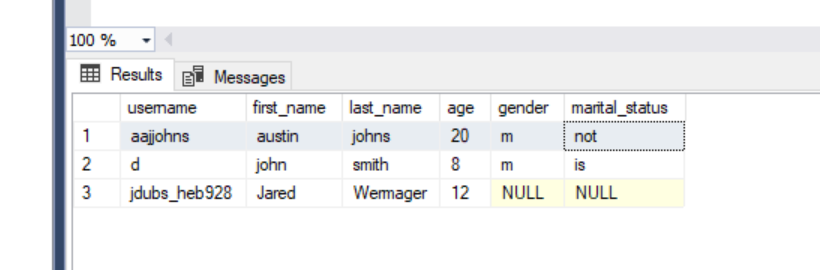
@marital\_status = N'not',

@statementtype = N'insert'

SELECT 'Return Value' = @return\_value

GO

select \* from user\_profile



USE [quiet\_thunder]

GO

DECLARE @return\_value int

EXEC @return\_value = [dbo].[MasterInsUpDel]

@username = N'aajjohns',

@first\_name = N'david',

@last\_name = N'kisiel',

@age = 21,

@gender = N'm',

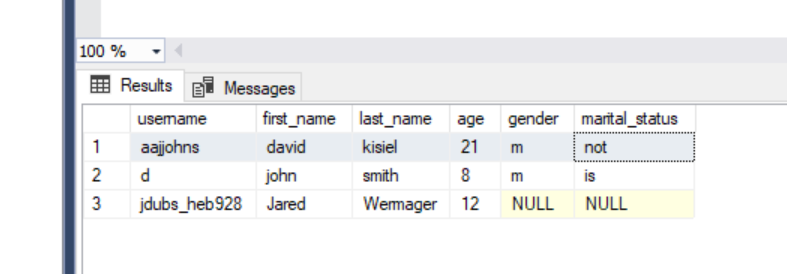
@marital\_status = N'not',

@statementtype = N'update'

SELECT 'Return Value' = @return\_value

GO

select \* from user\_profile



USE [quiet\_thunder]

GO

DECLARE @return\_value int

EXEC @return\_value = [dbo].[MasterInsUpDel]

@username = N'aajjohns',

@first\_name = N'david',

@last\_name = N'kisiel',

@age = 21,

@gender = N'm',

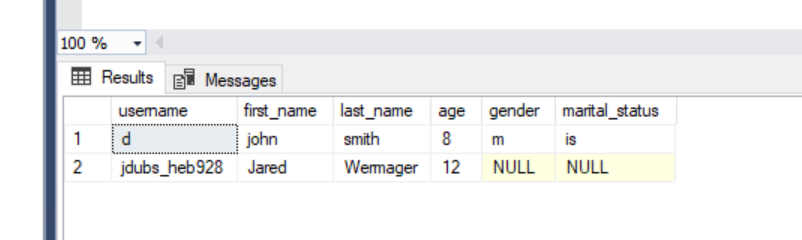
@marital\_status = N'not',

@statementtype = N'delete'

SELECT 'Return Value' = @return\_value

GO

select \* from user\_profile



GO

CREATE FUNCTION AllPosts ()

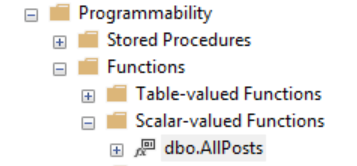
RETURNS INT

AS

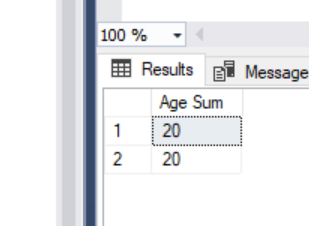
BEGIN

RETURN (SELECT SUM([age]) FROM [user\_profile])

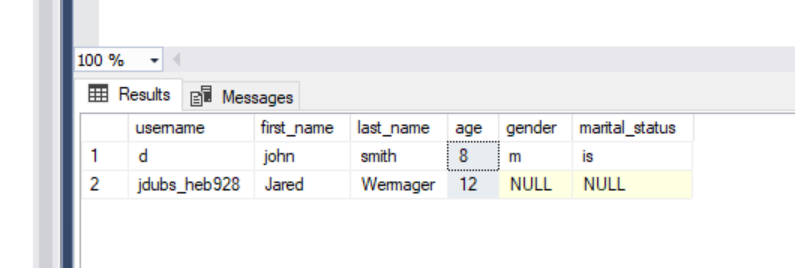
END



select dbo.AllPosts () as [Age Sum] from user\_profile



select \* from user\_profile



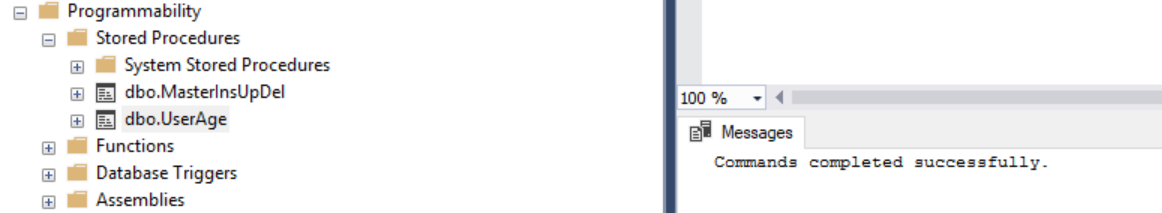
create procedure UserAge

as

begin

select \* from user\_profile where age = 8

end



USE [quiet\_thunder]

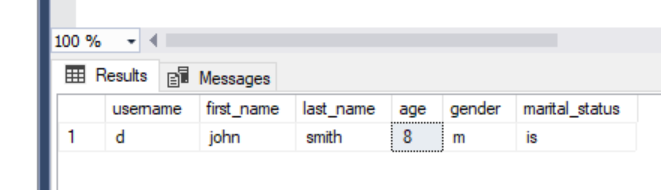
GO

DECLARE @return\_value int

EXEC @return\_value = [dbo].[UserAge]

SELECT 'Return Value' = @return\_value

GO



create procedure UserName

as

begin

select \* from user\_profile where username = 'jdubs\_heb928'

end



USE [quiet\_thunder]

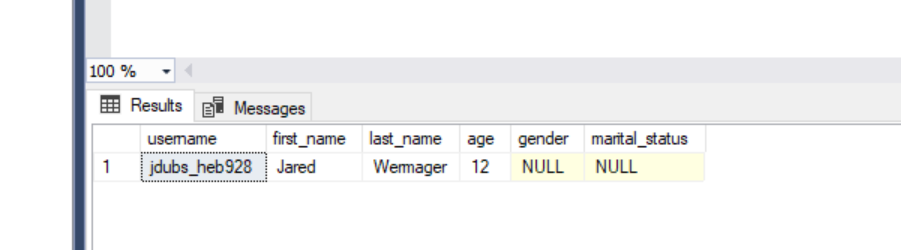
GO

DECLARE @return\_value int

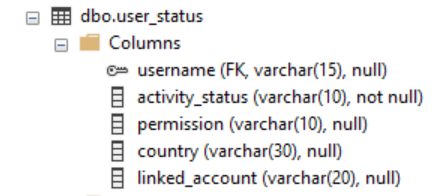
EXEC @return\_value = [dbo].[UserName]

SELECT 'Return Value' = @return\_value

GO



The foreign key on the username in user\_status is a constraint that prohibits the input of a new user in user status unless that user is already in our user\_profile table.



Application and Security Design

For the database’s security, we wanted to implement a couple of safeties for accidental data creation or prevent any kind of foreseen failures. We created a primary key on the username to establish a requirement for unique usernames, as no two users can have the same username. Correct syntax when creating the rules for the database was used to get precise responses from the database consistently.