

```
USE [MLogger]
GO
/***** Object: User [mlogger]      Script Date: 12/21/2008 19:18:08 *****/
CREATE USER [mlogger] FOR LOGIN [mlogger] WITH DEFAULT_SCHEMA=[dbo]
GO
/***** Object: FullTextCatalog [Full-Text Index]      Script Date: 12/21/2008 19:18:08 *****/
/
CREATE FULLTEXT CATALOG [Full-Text Index]
WITH ACCENT_SENSITIVITY = ON
AS DEFAULT
AUTHORIZATION [dbo]
GO
/***** Object: Table [dbo].[tblUsers]      Script Date: 12/21/2008 19:18:12 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tblUsers] (
    [userID] [int] NOT NULL,
    [userName] [varchar](50) NOT NULL,
    [password] [varchar](50) NOT NULL,
    CONSTRAINT [PK_Users] PRIMARY KEY CLUSTERED
(
    [userID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
    = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[tblUsers] ([userID], [userName], [password]) VALUES (0, N'ironix', N'trustno1')
INSERT [dbo].[tblUsers] ([userID], [userName], [password]) VALUES (1, N'foo', N'bar')
/***** Object: StoredProcedure [dbo].[spValidateUser]      Script Date: 12/21/2008 19:18:20 *
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Stored procedure for user validation
-- Returns: 1 if valid
--          0 if invalid
CREATE PROCEDURE [dbo].[spValidateUser](@userName varchar(50), @password varchar(50)) AS
    IF EXISTS (SELECT * FROM tblUsers WHERE userName = @userName AND [password] = @password)
        RETURN 1
    ELSE
        RETURN 0
GO
/***** Object: StoredProcedure [dbo].[spDeleteUser]      Script Date: 12/21/2008 19:18:20 ***
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Stored procedure for user deletion
-- Returns: 1 if existed and deleted
--          0 if didn't exist
CREATE PROCEDURE [dbo].[spDeleteUser](@userID INT) AS
    IF EXISTS (SELECT * FROM tblUsers WHERE userID = @userID)
    BEGIN
        DELETE FROM tblUsers WHERE userID = @userID
        RETURN 1
    END
    ELSE
        RETURN 0
GO
/***** Object: StoredProcedure [dbo].[spCreateUser]      Script Date: 12/21/2008 19:18:20 ***
*****/
```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Stored procedure for creating a new user
-- Returns: 1 if created
--          0 if user name existed and not created
CREATE PROCEDURE [dbo].[spCreateUser] (@userName varchar(50), @password varchar(50)) AS
    DECLARE @count INT
    DECLARE @maxUserID INT

    SELECT @count = COUNT(*) FROM tblUsers

    IF @count > 0
        SELECT @maxUserID = MAX(userID + 1) FROM tblUsers
    ELSE
        SET @maxUserID = 0

    IF NOT EXISTS (SELECT * FROM tblUsers WHERE userName = @userName)
    BEGIN
        INSERT INTO tblUsers(userID, userName, [password]) VALUES(@maxUserID, @userName, @
        password)
        RETURN 1
    END
    ELSE
        RETURN 0
GO
/***** Object: Table [dbo].[tblSessions]      Script Date: 12/21/2008 19:18:20 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[tblSessions] (
    [userID] [int] NOT NULL,
    [sessionID] [int] NOT NULL,
    [expiration] [datetime] NOT NULL,
    CONSTRAINT [PK_Session] PRIMARY KEY CLUSTERED
(
    [userID] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
    = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
INSERT [dbo].[tblSessions] ([userID], [sessionID], [expiration]) VALUES (0, 1234567890, CAST
    (0x00009E51003A6D2B AS DateTime))
/***** Object: Table [dbo].[tblMessages]      Script Date: 12/21/2008 19:18:20 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tblMessages] (
    [messageID] [int] NOT NULL,
    [userID] [int] NOT NULL,
    [postDate] [datetime] NOT NULL,
    [title] [varchar](max) NULL,
    [message] [varchar](max) NULL,
    CONSTRAINT [PK_tblMessages] PRIMARY KEY CLUSTERED
(
    [messageID] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
    = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[tblMessages] ([messageID], [userID], [postDate], [title], [message]) VALUES (0,
    0, CAST(0x00009B79012F9322 AS DateTime), N'Memo Title', N'Memo Description')

```

```

/***** Object: Table [dbo].[tblMedia]      Script Date: 12/21/2008 19:18:20 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tblMedia] (
    [mediaID] [int] NOT NULL,
    [messageID] [int] NOT NULL,
    [mediaType] [varchar](50) NOT NULL,
    [mediaLocation] [varchar](255) NOT NULL,
    CONSTRAINT [PK_tblMedia] PRIMARY KEY CLUSTERED
(
    [mediaID] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[tblMedia] ([mediaID], [messageID], [mediaType], [mediaLocation]) VALUES (0, 0, N'gif-image', N'C:\Inetpub\MessageLogger\message-media\ironix\0\2008_25_21-06_25_16_976')
INSERT [dbo].[tblMedia] ([mediaID], [messageID], [mediaType], [mediaLocation]) VALUES (1, 0, N'gif-image', N'C:\Inetpub\MessageLogger\message-media\ironix\0\2008_25_21-06_25_17_007')
/***** Object: StoredProcedure [dbo].[spNewSessionID]      Script Date: 12/21/2008 19:18:20 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Creates a new session ID and stores it in the Session table
-- Returns: Session ID
CREATE PROCEDURE [dbo].[spNewSessionID] (@userName varchar(50)) AS
    DECLARE @sessionID INT
    DECLARE @userID INT

    -- Grab the user's ID
    SELECT @userID = userID FROM tblUsers WHERE userName = @userName

    -- Set the new session ID
    SELECT @sessionID = ABS(CAST(CAST(NEWID() AS VARBINARY) AS INT));

    -- Clear all previous sessions for the user
    DELETE FROM tblSessions WHERE userID = @userID

    -- Insert the new userID/session ID pair into the table with a 2-hour expiration
    INSERT INTO tblSessions(userID, sessionID, expiration) VALUES(@userID, @sessionID, DATEADD(hh, 2, GETDATE()))

    -- Returns the session ID
    RETURN @sessionID
GO
/***** Object: StoredProcedure [dbo].[spNewSession]      Script Date: 12/21/2008 19:18:20 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Creates a new session ID and stores it in the Session table
-- Returns: Session ID
CREATE PROCEDURE [dbo].[spNewSession] (@userID int) AS
    DECLARE @session INT

    -- Set the new session ID
    SELECT @session = ABS(CAST(CAST(NEWID() AS VARBINARY) AS INT));

    -- Clear all previous sessions for the user
    DELETE FROM tblSessions WHERE userID = @userID
```

```
-- Insert the new userID/session ID pair into the table with a 2-hour expiration
GO
/***** Object:  StoredProcedure [dbo].[spGetUserFromSession]      Script Date: 12/21/2008 19:18:20 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Stored procedure for posting new message
-- Returns: userName
CREATE PROCEDURE [dbo].[spGetUserFromSession](@sessionID int) AS
    DECLARE @userID      INT
    DECLARE @userName    VARCHAR(50)

    -- Retrieve the UserID associated with the current session
    SELECT @userID = userID FROM tblSessions WHERE sessionID = @sessionID

    -- Retrieve the User Name associated with the UserID
    SELECT @userName = userName FROM tblUsers WHERE userID = @userID

    SELECT @userName AS 'UserName'
GO
/***** Object:  StoredProcedure [dbo].[spClearExpiredSessions]      Script Date: 12/21/2008 19:18:20 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Clear expired sessions from the Session table
CREATE PROCEDURE [dbo].[spClearExpiredSessions] AS
    DELETE FROM tblSessions WHERE expiration < GETDATE()
GO
/***** Object:  StoredProcedure [dbo].[spValidateSession]      Script Date: 12/21/2008 19:18:20 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Stored procedure to check if the session ID is valid
-- Returns: 1 if valid
--          0 if invalid
CREATE PROCEDURE [dbo].[spValidateSession](@sessionID int) AS
    IF EXISTS (SELECT * FROM tblSessions WHERE sessionID = @sessionID AND expiration >= GETDATE())
        RETURN 1
    ELSE
        RETURN 0
GO
/***** Object:  StoredProcedure [dbo].[spPostNewMessage]      Script Date: 12/21/2008 19:18:20 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Stored procedure for posting new message
-- Returns: messageID
CREATE PROCEDURE [dbo].[spPostNewMessage](@sessionID int, @title varchar(MAX), @message varchar(MAX)) AS
    DECLARE @count      INT
    DECLARE @maxMessageID INT
    DECLARE @userID      INT

    SELECT @count = COUNT(*) FROM tblMessages

    -- Set the new message ID
    IF @count > 0
        SELECT @maxMessageID = MAX(messageID + 1) FROM tblMessages
    ELSE
```

```
SET @maxMessageID = 0

-- Retrieve the UserID associated with the current session
SELECT @userID = userID FROM tblSessions WHERE sessionID = @sessionID

INSERT INTO tblMessages(messageID, userID, postDate, title, [message]) VALUES(@
maxMessageID, @userID, GETDATE(), @title, @message)

RETURN @maxMessageID
GO
/***** Object:  StoredProcedure [dbo].[spPostMsgAttachment]    Script Date: 12/21/2008 19:18
:20 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Stored procedure for posting new message
CREATE PROCEDURE [dbo].[spPostMsgAttachment](@sessionID int, @messageID int, @mediaType
varchar(50), @mediaLocation varchar(255)) AS
DECLARE @count      INT
DECLARE @maxMediaID INT
DECLARE @userID      INT

SELECT @count = COUNT(*) FROM tblMedia

-- Set the new message ID
IF @count > 0
    SELECT @maxMediaID = MAX(mediaID + 1) FROM tblMedia
ELSE
    SET @maxMediaID = 0

INSERT INTO tblMedia(mediaID, messageID, mediaType, mediaLocation) VALUES(@maxMediaID, @
messageID, @mediaType, @mediaLocation)
GO
/***** Object:  ForeignKey [FK_tblSessions_tblUsers]    Script Date: 12/21/2008 19:18:20 ***
*****/
ALTER TABLE [dbo].[tblSessions] WITH CHECK ADD CONSTRAINT [FK_tblSessions_tblUsers] FOREIGN
KEY([userID])
REFERENCES [dbo].[tblUsers] ([userID])
GO
ALTER TABLE [dbo].[tblSessions] CHECK CONSTRAINT [FK_tblSessions_tblUsers]
GO
/***** Object:  ForeignKey [FK_tblMessages_tblUsers]    Script Date: 12/21/2008 19:18:20 ***
*****/
ALTER TABLE [dbo].[tblMessages] WITH CHECK ADD CONSTRAINT [FK_tblMessages_tblUsers] FOREIGN
KEY([userID])
REFERENCES [dbo].[tblUsers] ([userID])
GO
ALTER TABLE [dbo].[tblMessages] CHECK CONSTRAINT [FK_tblMessages_tblUsers]
GO
/***** Object:  ForeignKey [FK_tblMedia_tblMessages]    Script Date: 12/21/2008 19:18:20 ***
*****/
ALTER TABLE [dbo].[tblMedia] WITH CHECK ADD CONSTRAINT [FK_tblMedia_tblMessages] FOREIGN
KEY([messageID])
REFERENCES [dbo].[tblMessages] ([messageID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[tblMedia] CHECK CONSTRAINT [FK_tblMedia_tblMessages]
GO
```