Gamification Sevice

RankUpdater – A time triggered function, it is triggered at 12 am daily

* Creates Rank data records for all active teams, records are created each new week sunday - saturday
* Retrieves the jobOrders for a company the previous day
* It saves all jobOrder scores (using tags and weighted calculations) to their correct active teams.

ShardCreator – A queue triggered function, it is triggered when an item is enqueued. It can parallel execute multiple items simultaneously, this is used to minimize the bottle neck. To over come errors of items shard names matching from the nature of parallel executions. company row being inserted with matching shard name will throw an error due to a unique constraint applied to it. Items that cause an error to be thrown 5 times will be sent to a poison container. The poison container will have the same name as the queue with poison appended to the end e.g: queue name: create-shard. Poisoned container: create- shard-poison. The error can be seen on azure.

* The shardCreator will get the last shard name created and increment it, e.g shard1 => shard2
* It will then add the company to the company table reserving the shard name and setting the created date for other items to increment
* It will then create the database shard using the shard name
* And then create tables in that shard
* It will then send the admin account to the web api for a user row to be created for this shard. (it is recommended to remove this call from the from queue trigger and the functionality to be removed from the web api. Instead a queue should be created that is tasked with inserting the admin accounts when shards are created the items will be enqueued via the shardCreator and also any other area(programs) the client finds necessary. This method will be a more secure approach then the current process. It will also Allow multiple request to be made if an admin account creation fails, in comparison with the current mechanism only being able to create the admin account during shard creation. It will overcome both short comings of the current process and still preserve the core functionality with added benefits)
* Any changes made to the company table or the shard database schema must be reflected here. If the schema is not maintained it is probable that version issues will occur. The shard database schema must be constantly updated and maintained any time that a change is made to the database.

Gamification WebApi

Database

ShardManager

Begin

CREATE TABLE [dbo].[Company](

[Id] [uniqueidentifier] NOT NULL,

[Name] [varchar](100) NOT NULL,

[ShardName] [varchar](100) NOT NULL UNIQUE,

[CreatedDate] [datetime] NOT NULL DEFAULT (getdate()),

primary key([Id]));

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AspNetUsers](

[Id] [nvarchar](450) NOT NULL,

[UserName] [nvarchar](256) NULL,

[NormalizedUserName] [nvarchar](256) NULL,

[Email] [nvarchar](256) NULL,

[NormalizedEmail] [nvarchar](256) NULL,

[EmailConfirmed] [bit] NOT NULL,

[PasswordHash] [nvarchar](max) NULL,

[SecurityStamp] [nvarchar](max) NULL,

[ConcurrencyStamp] [nvarchar](max) NULL,

[PhoneNumber] [nvarchar](max) NULL,

[PhoneNumberConfirmed] [bit] NOT NULL,

[TwoFactorEnabled] [bit] NOT NULL,

[LockoutEnd] [datetimeoffset](7) NULL,

[LockoutEnabled] [bit] NOT NULL,

[AccessFailedCount] [int] NOT NULL,

[CompanyId] [nvarchar](450) NULL,

CONSTRAINT [PK\_AspNetUsers] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AspNetUserTokens](

[UserId] [nvarchar](450) NOT NULL,

[LoginProvider] [nvarchar](450) NOT NULL,

[Name] [nvarchar](450) NOT NULL,

[Value] [nvarchar](max) NULL,

CONSTRAINT [PK\_AspNetUserTokens] PRIMARY KEY CLUSTERED

(

[UserId] ASC,

[LoginProvider] ASC,

[Name] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

ALTER TABLE [dbo].[AspNetUserTokens] WITH CHECK ADD CONSTRAINT [FK\_AspNetUserTokens\_AspNetUsers\_UserId] FOREIGN KEY([UserId])

REFERENCES [dbo].[AspNetUsers] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserTokens] CHECK CONSTRAINT [FK\_AspNetUserTokens\_AspNetUsers\_UserId]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AspNetUserRoles](

[UserId] [nvarchar](450) NOT NULL,

[RoleId] [nvarchar](450) NOT NULL,

CONSTRAINT [PK\_AspNetUserRoles] PRIMARY KEY CLUSTERED

(

[UserId] ASC,

[RoleId] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[AspNetUserRoles] WITH CHECK ADD CONSTRAINT [FK\_AspNetUserRoles\_AspNetRoles\_RoleId] FOREIGN KEY([RoleId])

REFERENCES [dbo].[AspNetRoles] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserRoles] CHECK CONSTRAINT [FK\_AspNetUserRoles\_AspNetRoles\_RoleId]

GO

ALTER TABLE [dbo].[AspNetUserRoles] WITH CHECK ADD CONSTRAINT [FK\_AspNetUserRoles\_AspNetUsers\_UserId] FOREIGN KEY([UserId])

REFERENCES [dbo].[AspNetUsers] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserRoles] CHECK CONSTRAINT [FK\_AspNetUserRoles\_AspNetUsers\_UserId]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AspNetUserLogins](

[LoginProvider] [nvarchar](450) NOT NULL,

[ProviderKey] [nvarchar](450) NOT NULL,

[ProviderDisplayName] [nvarchar](max) NULL,

[UserId] [nvarchar](450) NOT NULL,

CONSTRAINT [PK\_AspNetUserLogins] PRIMARY KEY CLUSTERED

(

[LoginProvider] ASC,

[ProviderKey] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

ALTER TABLE [dbo].[AspNetUserLogins] WITH CHECK ADD CONSTRAINT [FK\_AspNetUserLogins\_AspNetUsers\_UserId] FOREIGN KEY([UserId])

REFERENCES [dbo].[AspNetUsers] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserLogins] CHECK CONSTRAINT [FK\_AspNetUserLogins\_AspNetUsers\_UserId]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AspNetUserClaims](

[Id] [int] IDENTITY(1,1) NOT NULL,

[UserId] [nvarchar](450) NOT NULL,

[ClaimType] [nvarchar](max) NULL,

[ClaimValue] [nvarchar](max) NULL,

CONSTRAINT [PK\_AspNetUserClaims] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

ALTER TABLE [dbo].[AspNetUserClaims] WITH CHECK ADD CONSTRAINT [FK\_AspNetUserClaims\_AspNetUsers\_UserId] FOREIGN KEY([UserId])

REFERENCES [dbo].[AspNetUsers] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserClaims] CHECK CONSTRAINT [FK\_AspNetUserClaims\_AspNetUsers\_UserId]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AspNetRoles](

[Id] [nvarchar](450) NOT NULL,

[Name] [nvarchar](256) NULL,

[NormalizedName] [nvarchar](256) NULL,

[ConcurrencyStamp] [nvarchar](max) NULL,

CONSTRAINT [PK\_AspNetRoles] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[\_\_EFMigrationsHistory](

[MigrationId] [nvarchar](150) NOT NULL,

[ProductVersion] [nvarchar](32) NOT NULL,

CONSTRAINT [PK\_\_\_EFMigrationsHistory] PRIMARY KEY CLUSTERED

(

[MigrationId] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

end

Shard

Begin

Create Table [Team]

([Id] uniqueidentifier not null,

[Name] varchar(100) not null,

[IsActive] [bit] not null default 1,

primary key([Id]));

Create Table [User]

([Id] uniqueidentifier not null,

[FirstName] varchar(100) not null,

[LastName] varchar(100) not null,

[Email] [varchar](100) NOT NULL,

[Image] [varbinary](max) NULL,

[Role] [varchar](100) NOT NULL,

[IsActive] [bit] NOT NULL default 1,

primary key([Id]));

Create Table [Asset]

([Id] [uniqueidentifier] not null,

[Name] varchar(100) not null,

[Weight] int check ([Weight] between 1 and 10) not null default 1,

[Cost] decimal(18,2) check ([Cost] between 0.00 and 9999999999999999.99) not null default 0.00,

[Rfid] [varchar](30) Not NULL UNIQUE,

primary key([Id]));

Create Table [Rank]

([Id] [uniqueidentifier] NOT NULL,

[TeamId] uniqueidentifier not null,

[week] datetime not null,

[ActualScore] bigint check ([ActualScore] between 0 and 9223372036854775807) not null default 0 ,

[TotalScore] bigint check ([TotalScore] between 0 and 9223372036854775807) not null default 0,

[ActualWeightedScore] bigint check ([ActualWeightedScore] between 0 and 9223372036854775807) not null default 0 ,

[TotalWeightedScore] bigint check ([TotalWeightedScore] between 0 and 9223372036854775807) not null default 0,

primary key([Id]),

Foreign key([TeamId]) references[Team]);

Create Table [MemberOf]

([TeamId] uniqueidentifier not null,

[UserId] [uniqueidentifier] NOT NULL,

primary key([TeamId], [UserId]),

Foreign key([UserId]) references[User],

Foreign key([TeamId]) references[Team]);

End