Review of the PADification database July 6, 2017

PADification database v.0.1 (according to PADification ERD)

Author: Elie Godbout

## Database Creation / Population

**Prerequisites**

* Create\_db.bat file
  + .sql files (all in the same repository)
    - Generate-PADification-DB.sql
    - Table-Create.sql
    - LatentSkill-Inserts.sql
    - AttributeAndType-Inserts.sql
    - ActiveSkill-Inserts.sql
    - LeaderSkill-Inserts.sql
    - AwokenSkill-Inserts.sql
    - AwokenSkillList-Inserts.sql
    - MonsterClass-Inserts.sql
    - EvolutionTree-Inserts.sql
    - AwokenBadges-Inserts.sql
* Drop\_db.bat file
  + .sql file
    - Drop-PADification-DB.sql
* MS SQL Server sqlcmd.exe
* SQLCMD utilities installed
* Microsoft ODBC Driver installed
* MS SQL server management studio
* SQL server (MSSQLSERVER) must be running

**Create/populate the Database**

* Double click the “create\_db.bat” file to begin execution.
* Press the Return key to continue execution.
* Creation and population process are completed automatically.

**Delete the Database**

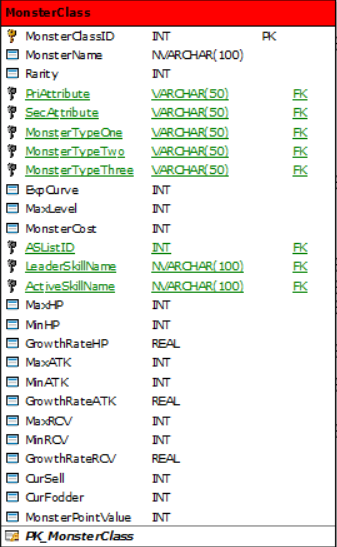
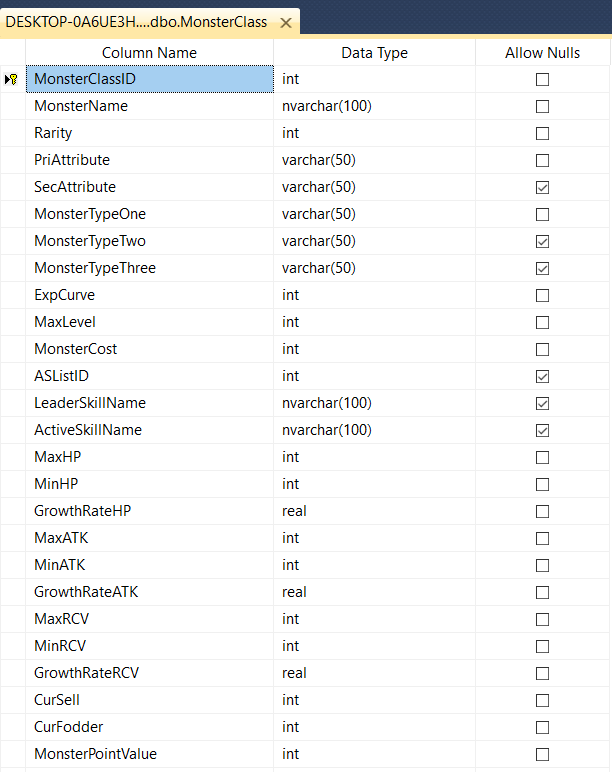
* Be sure to refresh the database as well as close any and all .sql files associated with the PADification database aside from *Drop-PADification.sql.* and make sure the PADification program/application is closed as well.
* Double click on “drop\_db.bat” file to begin the execution
* Press the Return key to continue execution.
* Deletion process is completed automatically.

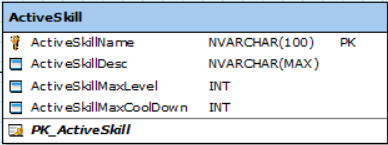
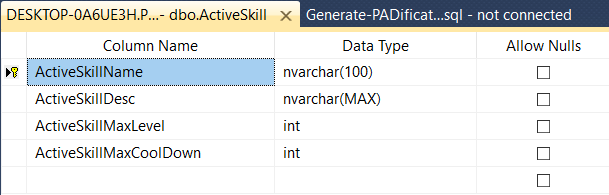
**Utilise and update the database**

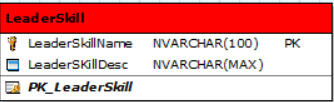
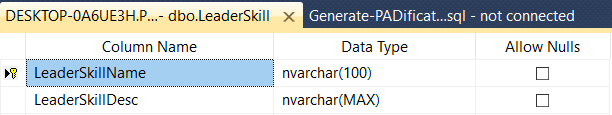
* The PADification product is a combination of MSQL database and Python functionality. Users will use the functionality defined by the Python source code in order to retrieve information and update existing information or implementing new data into the database.
* Administrators only: In the event a major update is required on the structure or the contents of the PADification database, admins or developers can access the .sql files for creating and populating the database and perform queries using MSQL server management studio.

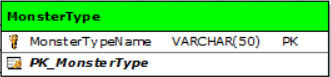
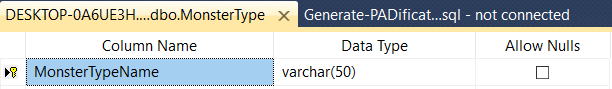
# PADification Database

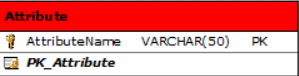
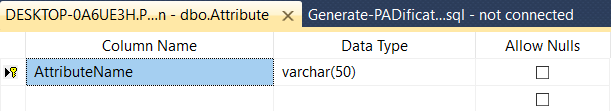
**Comparison between PADification DB, ERD & MSQL server management**

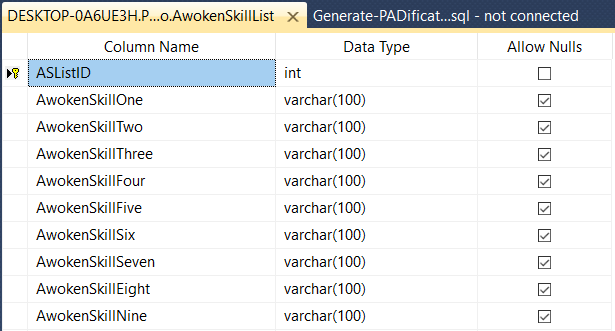
 

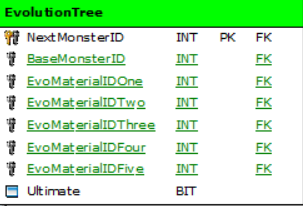
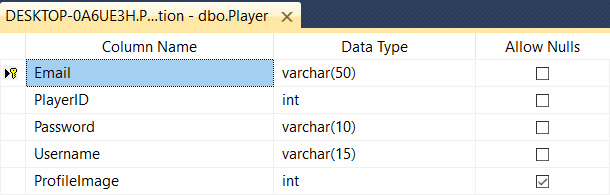
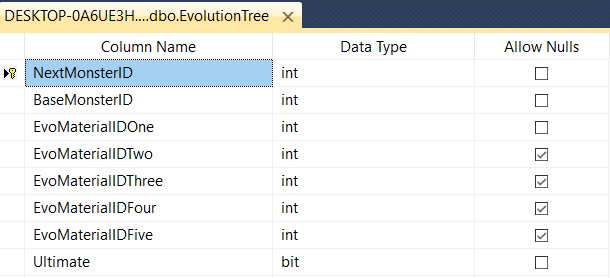


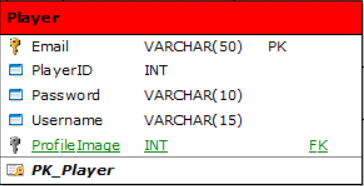


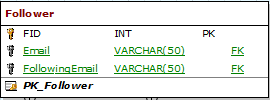
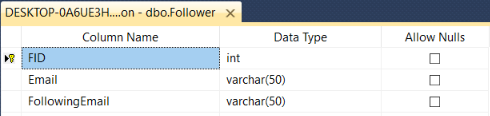


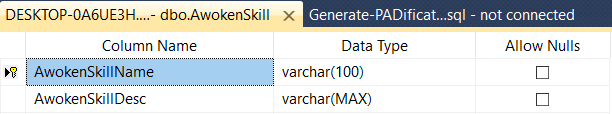


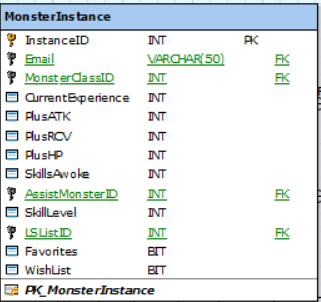
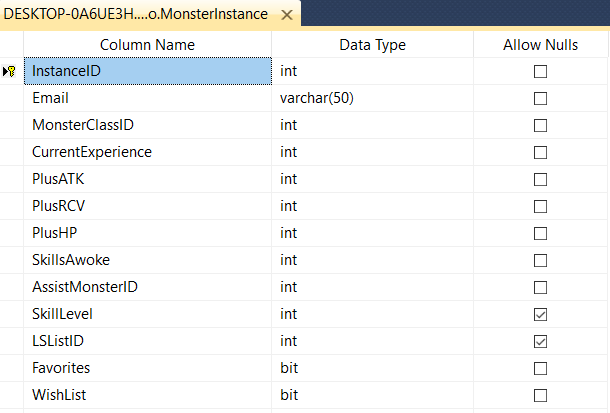


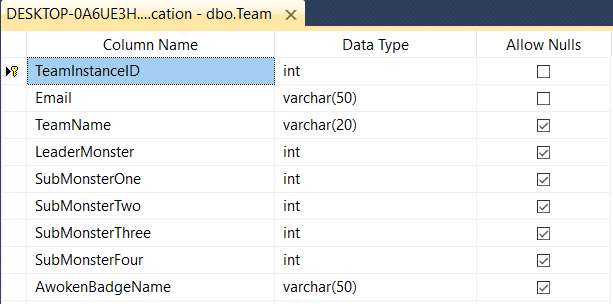
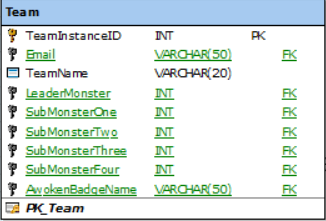
 

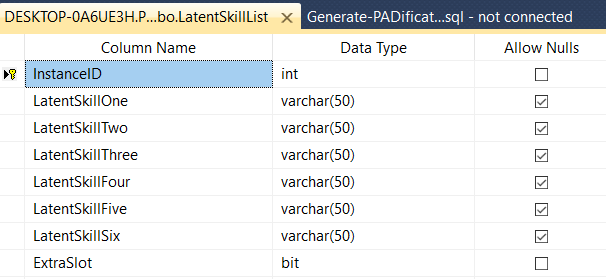


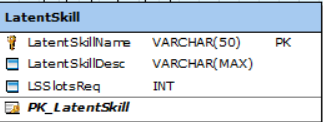
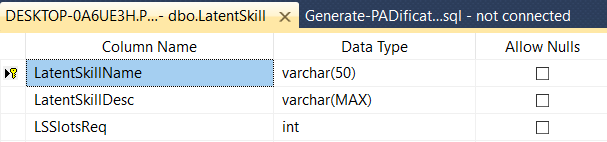
 

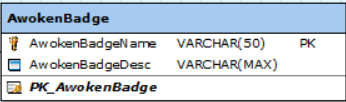
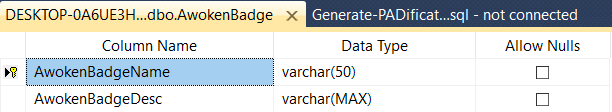




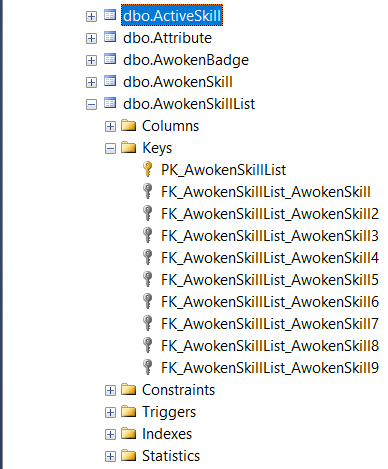






# Entity Relationships

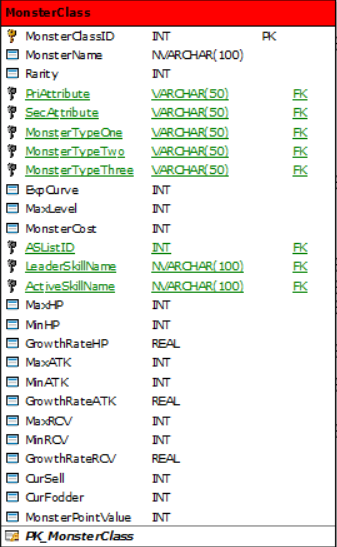
**AwokenSkillList**

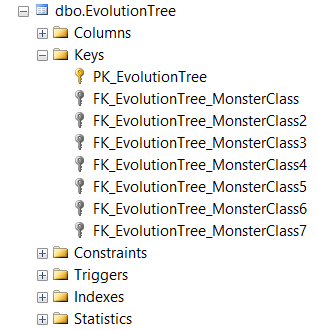
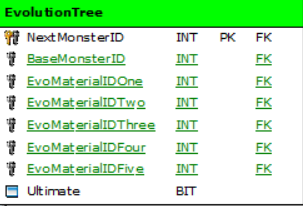


Parent table(s): AwokenSkill

AwokenSkill field: AwokenSkillName. Foreign key relationship with AwokenSkillList field(s):

* AwokenSkillOne – (NULL)
* AwokenSkillTwo – (NULL)
* AwokenSkillThree – (NULL)
* AwokenSkillFour – (NULL)
* AwokenSkillFive – (NULL)
* AwokenSkillSix – (NULL)
* AwokenSkillSeven – (NULL)
* AwokenSkillEight – (NULL)
* AwokenSkillNine – (NULL)
  + Purpose & utility: Monster may have 0 to 9 awoken skills associated to them and so, all nine fields must allow NULL. If a monster has any number of awoken skills, these must be listed in a specific order. Awoken skills will always increment from AwokenSkillOne.
    - E.g. If a monster has 3 awoken skills, that MonsterClassID will have three foreign key relationships to the ASListID in the AwokenSkillList table; AwokenSkillOne, AwokenSkillTwo & AwokenSkillThree.

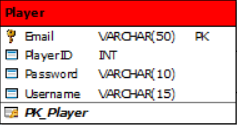
**Evolution tree**

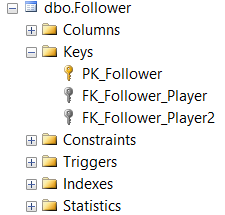
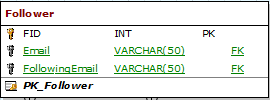


Parent table(s): MonsterClass

MonsterClass field: MonsterClassID. Foreign key relationship with EvolutionTree field(s):

* NextMonsterID - (Primary Key)
* BaseMonsterID - (NOT NULL)
* EvoMaterialIDOne - (NOT NULL)
* EvoMaterialIDTwo - (NULL)
* EvoMaterialIDThree - (NULL)
* EvoMaterialIDFour - (NULL)
* EvoMaterialIDFive - (NULL)
  + Purpose & Utility: While a monster may be able to evolve into more than one type of monster, a monster cannot be the result of two different base monsters. This is why NextMonsterID is the primary key. EvoMaterialOne is NOT NULL as it requires a minimum of one monster to perform an evolution. The type of monster and quantity of Evo-Materials required for evolution varies between monsters. This is why EvoMaterialTwo, Three and Four allow NULL.
    - E.g. Refer to the PADification SRS 5.2.1.3.11 Evolutions.

**Follower**

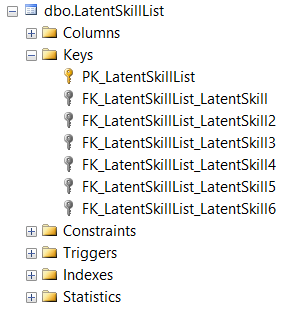
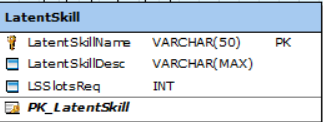


Parent table(s): Player

Player field: Username. Foreign key relationship with Follower field(s):

* Email – (NOT NULL)
* FollowingEmail – (NOT NULL)
  + Purpose & Utility: Followers are players associated to another player by interaction. As all players are identified by their Email, this field is mandatory in order to link the two accounts.
    - E.g. When a user decides to follow another player by pressing the follow button, that player’s profile and information will be saved and accessible to the user from that point forward.

**LatentSkillList**

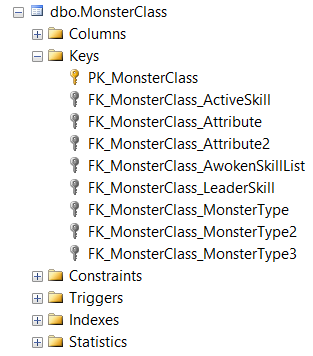
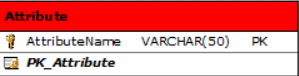
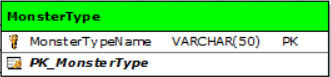
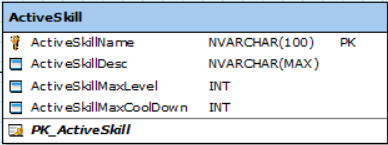
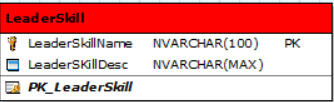


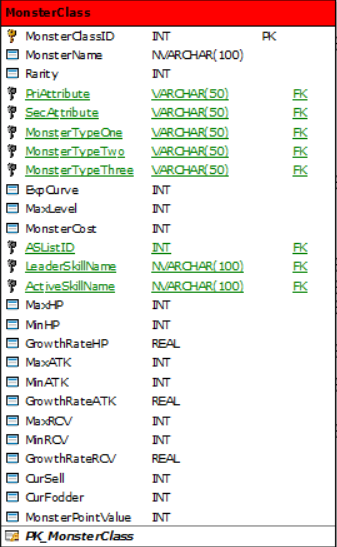
Parent table(s): LatentSkill

LatentSkill field: LatentSkillName. Foreign key relationship with LatentSkillList field(s):

* LatentSkillOne – (NULL)
* LatentSkillTwo – (NULL)
* LatentSkillThree – (NULL)
* LatentSkillFour – (NULL)
* LatentSkillFive – (NULL)
* LatentSkillSix – (NULL)
  + Purpose & Utility: All monsters have access to Latent Awoken Skills. However, no monster has one of those skills by default and so, these fields allow NULL. These slots act as a cost for equipping latent awoken skills. Certain skills require 2 slots to be equipped. LatentSkillSix is an extra slot awarded to monsters by special means. This sixth slot will not be available unless the ExtraSlot field returns TRUE.
    - E.g. Users have free range to add and remove Latent awoken skills to their monster instances as they see fit as long as they have the slots available.

**MonsterClass**





Parent table(s): Attribute, MonsterType, ActiveSkill, LeaderSkill, AwokenSkillList.

Attribute field: AttributeName. Foreign key relationship with MonsterClass field(s):

* PriAttribute – (NOT NULL)
* SecAttribute – (NULL)
  + Purpose & Utility: Every monster must have a PriAttribute. The SecAttribute is optional however, as not all monsters own a secondary Attribute.

MonsterType field: MonsterTypeName. Foreign key relationship with MonsterClass field(s):

* MonsterTypeOne – (NOT NULL)
* MonsterTypeTwo – (NULL)
* MonsterTypeThree – (NULL)
  + Purpose & Utility: Every monster must have a MonsterTypeOne. MonsterTypeTwo & MonsterTypeThree are optional as not all monsters have two or three Types.

ActiveSkill field: ActiveSkillName. Foreign key relationship with MonsterClass field(s):

* ActiveSkillName – (NULL)
  + Purpose & Utility: Not every monster has an ActiveSkill and so, this filed can be NULL.

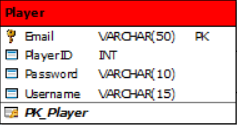
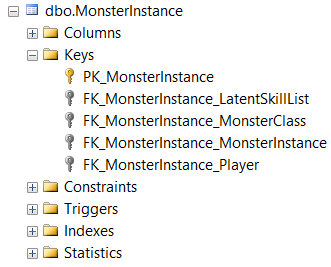
LeaderSkill field: LeaderSkillName. Foreign key relationship with MonsterClass field(s):

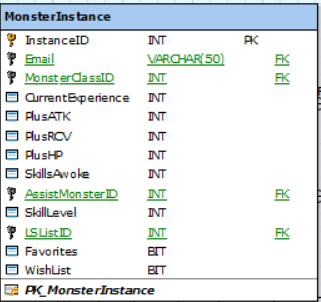
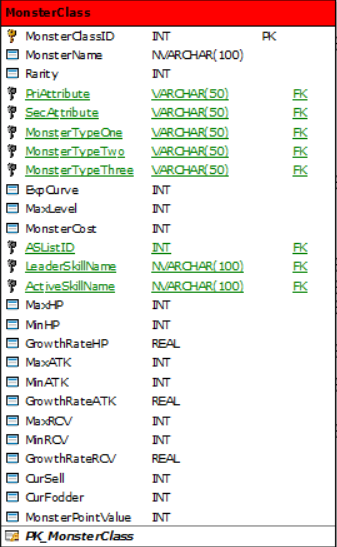
* LeaderSkillName – (NULL)
  + Purpose & Utility: Not every monster has a LeaderSkill and so, this field can be NULL.

AwokenSkillList field: ASListID. Foreign key relationship with MonsterClass field(s):

* ASListID – (NULL)
  + Purpose & Utility: Not all monsters have awoken skills and so, this field can be NULL.

**MonsterInstance**





Parent table(s): LatentSkillList, MonsterClass, Player, MonsterInstance

LatentSkillList field: LSListID. Foreign key relationship with MonsterInstance field(s):

* LSListID – (NULL)
  + Purpose & Utility: As LatentAwokenSkills can be applied and removed, this filed can be NULL. Having the LSListID NULL means the monster does not contain any Latent Awoken Skills.

MonsterClass field: MonsterClassID. Foreign key relationship with MonsterInstance field(s):

* MonsterClassID – (NOT NULL)
  + Purpose & Utility: Every monster has a predetermined ID number associated to them. This relationship is required to identify what sort of monster the player owns/is using.
    - E.g. If a user is tampering with one of their monster instances, it is important for the user and developers to know that the monster being tampered is consistently the same sort. Only some properties will be changed.

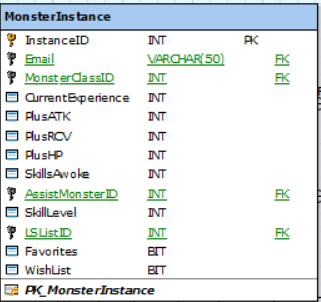
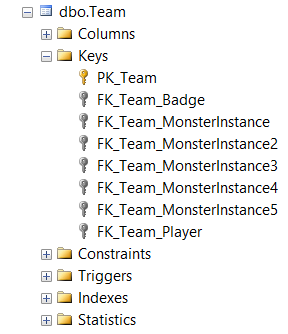
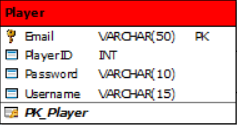
Player field: Username. Foreign key relationship with MonsterInstance field(s):

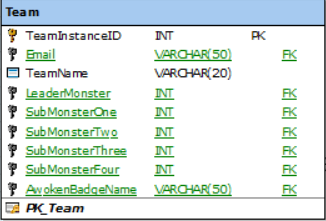
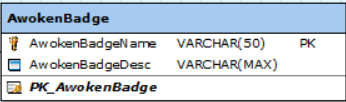
* Email – (NOT NULL)
  + Purpose & Utility: Monsters can be associated to specific users. This field identifies that link.

MonsterInstance field: AssistMonsterID. Foreign key relationship with MonsterInstance field(s):

* InstanceID – (NOT NULL)
  + Purpose & Utility: The act of assisting monsters requires to monsters owned by a user and so, the InstanceID is required to identify which specific monster is used as assist monster.

**Team**





Parent table(s): MonsterInstance, AwokenBadge, Player

MonsterInstance field: InstanceID. Foreign key relationship with Team field(s):

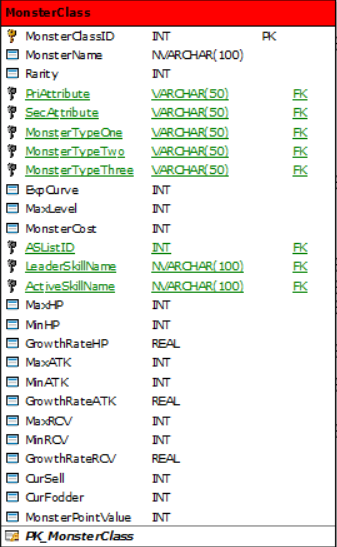
* LeaderMonster – (NOT NULL)
* SubMonsterOne – (NULL)
* SubMonsterTwo – (NULL)
* SubMonsterThree – (NULL)
* SubMonsterFour – (NULL)
  + Purpose & Utility: Teams are comprised of monster instances. Teams require a Leader in order to be created and so, it cannot be NULL. The four sub monsters are optional and so, they can be NULL.

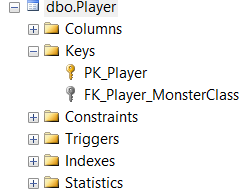
Player field: Username. Foreign key relationship with Team field(s):

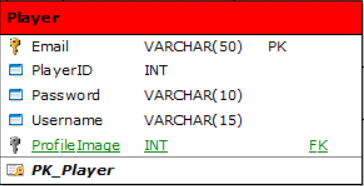
* Username – (NOT NULL)
  + Purpose & Utility: Teams are associated to their creators by the Username of the specific user.

AwokenBadge field: BadgeName. Foreign key relationship with Team field(s):

* AwokenBadgeName – (NULL)
  + Purpose & Utility: A team is not required to have an Awoken Badge and so, this field can be NULL.

**Player**



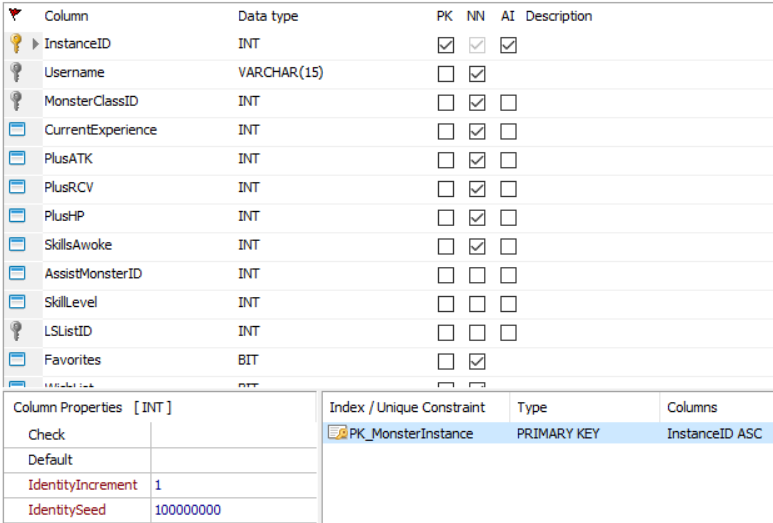
Parent table(s): MonsterClass.

MonsterClass field: MonsterClassID. Foreign key relationship with Player field(s):

* ProfileImage – (NULL)
  + Purpose & Utility: Optional personalisation for users.

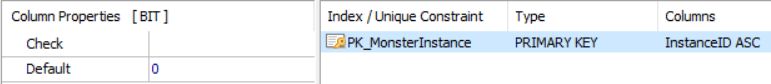
# Field with special properties

MonsterInstance

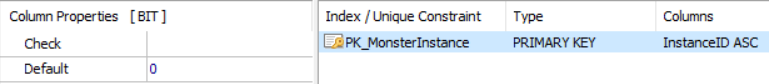
Field: InstanceID

Identity Seed: 100,000,000

Auto Increment: 1

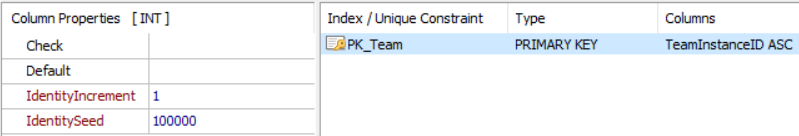
Field: Favorites

Default: 0

Field: WishList

Default: 0

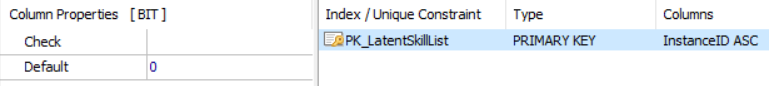
Team

Field: TeamInstanceID

Identity Seed: 100,000

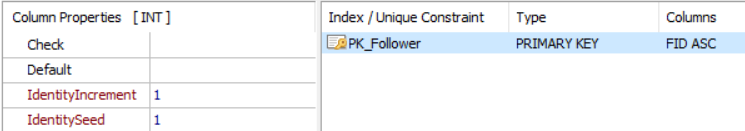
Auto Increment: 1

LatentSkillList

Field: ExtraSlot

Default: 0

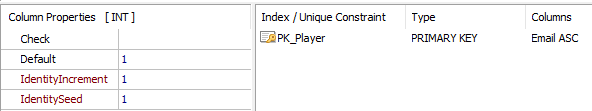
Follower

Field: FID

Identity Seed: 1

Auto Increment: 1

Player

Field: ProfileImage

Default: 1