

MISSION 1 WARCRAFT

PAIKAN

Edris

le 13/06/2022

PARTIE 1

1) Proposer une modélisation des données.

Personnage(id,Inscription,niveau,pseudonyme,personnages mask,races id)
Classes(classes id , classes name ,classes power type,classes mask,races id)
Races(races id , races name , races side,races mask)
clef primaire: id
clef étrangère : idc en ref a classe,id
clef étrangère : idr_Race.id

2) Créer les tables races et classes en local en fonction de votre modélisation et des scripts fournis. On utilisera pour accéder à MySQL l'interface Adminer. Attention à l'erreur dans le script 2

```
CREATE TABLE IF NOT EXISTS classes (  
classes_id INT primary key,  
classes_mask INT NULL,  
classes_name VARCHAR(20) NULL,  
classes_power_type VARCHAR(11) NULL,  
races_id INT NULL  
)ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Requête exécutée avec succès, 0 ligne modifiée. (0.034 s) [Modifier](#)

```
CREATE TABLE IF NOT EXISTS races (  
races_id INT primary key,  
races_mask INT NULL,  
races_name VARCHAR(20) NULL,  
races_side VARCHAR(8) NULL  
)ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Requête exécutée avec succès, 0 ligne modifiée. (0.035 s) [Modifier](#)

3) Supprimer les colonnes races_mask et classes_mask.

```
alter table classes drop classes_mask
```

Requête exécutée avec succès, 0 ligne modifiée.

```
alter table races drop races_mask
```

Requête exécutée avec succès, 0 ligne modifiée.

4)Créer la table personnage en indiquant l'identifiant de la classe sans les contraintes de clefs étrangères. La race est déterminée par la classe. On ne doit donc pas indiquer la race dans personnage.

```
CREATE TABLE IF NOT EXISTS personnages (  
  identification INT primary key,  
  Inscription Int NULL,  
  niveau INT NULL,  
  pseudonyme VARCHAR(20) NULL,  
  races_id INT NULL  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

5)Insérer les données correspondantes au texte précédent.

```
INSERT INTO personnages VALUES  
(1,"20-02-19",60,"Daril",1),
```

```
INSERT INTO personnages VALUES  
(2,"2005-08-15",100,"Jaco",2)
```

Requête exécutée avec succès, 1 ligne modifiée.

```
INSERT INTO personnages VALUES  
(3,"2017-11-17",15,"Dunbar",7)
```

Requête exécutée avec succès, 1 ligne modifiée.

```
INSERT INTO personnages VALUES  
(4,"2018-10-18",115,"Zakyku",9)
```

Requête exécutée avec succès, 1 ligne modifiée.

6)Supprimer la race Humain. Quel problème non visible est généré ?

```
delete from races where races_id=1
```

Requête exécutée avec succès, 1 ligne modifiée.

La race Humain disparaît , mais l'utilisateur a toujours la même races id

<input type="checkbox"/> Modification	races_id	races_name	races_side
<input type="checkbox"/> modifier	2	Orc	horde
<input type="checkbox"/> modifier	3	Nain	alliance
<input type="checkbox"/> modifier	4	Elfe de la nuit	alliance
<input type="checkbox"/> modifier	5	Mort-vivant	horde
<input type="checkbox"/> modifier	6	Tauren	horde
<input type="checkbox"/> modifier	7	Gnome	alliance
<input type="checkbox"/> modifier	8	Troll	horde
<input type="checkbox"/> modifier	9	Gobelin	horde

<input type="checkbox"/> Modification	identification	Inscription	niveau	pseudonyme	races_id
<input type="checkbox"/> modifier	1	2019-09-25	60	Daril	1

7)Executer la ligne suivante ALTER TABLE `classes` ADD INDEX `races_id` (`races_id`);

```
ALTER TABLE `classes` ADD INDEX `races_id` (`races_id`)
```

Requête exécutée avec succès, 0 ligne modifiée.

8)Réinsérer Humain.

```
INSERT INTO races VALUES (1,"Humain","alliance")
```

Requête exécutée avec succès, 1 ligne modifiée.

9)Vérifier que votre champs races_id soit du meme typez dans les deux tables (sans le null également). Ajouter la contrainte d'intégrité fonctionnelle permettant de vérifier qu'une classe possède une race dans la table classe

```
ALTER TABLE classes ADD FOREIGN KEY (races_id) REFERENCES races(races_id)
```

Requête exécutée avec succès, 10 lignes modifiées. (0.117 s) [Modifier](#)

10) Supprimer la race Gobelin. Que se passe-t-il ?

```
DELETE FROM races WHERE races_id=9
```

Erreur dans la requête (1451): Cannot delete or update a parent row: a foreign key constraint fails (`beldjs`.`classes`, CONSTRAINT `classes_ibfk_1` FOREIGN KEY (`races_id`) REFERENCES `races` (`races_id`))

PARTIE 2

1) Insérer les données correspondantes au texte précédent avec la date d'inscription d'aujourd'hui. Quel problème non visible est généré ? Ne pas le résoudre.

```
INSERT INTO personnages VALUES  
(5, "2019-09-26", 50, "Saperlipopette", 6)
```

Requête exécutée avec succès, 1 ligne modifiée.

```
INSERT INTO personnages VALUES  
(6, "2019-09-26", 10, "Zigomar", 10)
```

Requête exécutée avec succès, 1 ligne modifiée.

```
INSERT INTO personnages VALUES  
(7, "2019-09-26", 105, "Sperpinette", 8)
```

Requête exécutée avec succès, 1 ligne modifiée.

L'erreur est que Saperlipopette ne peut pas être Tauren prêtre à cause de la races_id

2) Proposer une modélisation des données pour résoudre ce problème.

Personnage(id, Inscription, niveau, pseudonyme, races_id)
Classes(classes_id, classes_name, classes_power_type)
Races(races_id, races_name, races_side,)
clef primaire: id
clef étrangère : idc en ref à classe, id
clef étrangère : idr_Race.id


```
ALTER TABLE personnages ADD classes_id int(11)
```

Requête exécutée avec succès, 0 ligne modifiée.

Rajout de la colonne classes_id

3) Permettre en modifiant les tables en SQL et selon votre modélisation l'ajout des prêtres. On pensera à ajouter la clef étrangère de la classe dans la table personnage et sûrement une autre clef quelque part...

```
ALTER TABLE classes DROP races_id
```

Requête exécutée avec succès, 0 ligne modifiée.

```
ALTER TABLE personnages ADD FOREIGN KEY (classes_id) REFERENCES classes(classes_id)
```

Requête exécutée avec succès, 7 lignes modifiées. (0.165 s) [Modifier](#)

4) Ajouter les contraintes d'intégrités fonctionnelles permettant de vérifier que la race et la classe existe.

```
ALTER TABLE personnages ADD CONSTRAINT RACE_CONTRAINTE CHECK (races_id >=0)
```

Requête exécutée avec succès, 7 lignes modifiées. (0.162 s) [Modifier](#)

```
ALTER TABLE personnages ADD CONSTRAINT CLASSE_CONTRAINTE CHECK (classes_id >=0)
```

Requête exécutée avec succès, 7 lignes modifiées. (0.130 s) [Modifier](#)

PARTIE 3 :

1) Proposer une modélisation des données.

Personnage(id, Inscription, niveau, pseudonyme, races_id, classes_id, monture_id)

Classes(classes id, classes name, classes power type)

Races(races id, races name, races side,)

Mounts

(mounts_id, mounts_icon, mounts_items_id, mounts_name, mounts_quality_id, mounts_spell_id)

clef primaire: id

clef étrangère : classes_id

clef étrangère : races_id

clef étrangère : mounts_id

2) Réaliser la table monture en corrigeant le script.

```
CREATE TABLE `mounts` (
  `mounts_creature_id` int(11) NOT NULL,
  `mounts_icon` varchar(80) DEFAULT NULL,
  `mounts_item_id` int(11) DEFAULT NULL,
  `mounts_name` varchar(60) DEFAULT NULL,
  `mounts_quality_id` int(11) DEFAULT NULL,
  `mounts_spell_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`mounts_creature_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Requête exécutée avec succès, 0 ligne modifiée. (0.050 s) [Modifier](#)

```
INSERT INTO `mounts` (`mounts_creature_id`, `mounts_icon`, `mounts_item_id`, `mounts_name`, `mounts_quality_id`)
(-81772, 'inv_dogmount', 0, 'Shu Zen, la sentinelle divine', 1, 259395),
(-80358, 'inv_armoredraptor', 0, 'Ravasaure doré', 1, 255696),
(-80357, 'inv_dressedhorse', 0, 'Étalon crin-de-mer', 1, 255695),
(-79790, 'inv_lightforgedelegg_blue', 153044, 'Gangrebroyeur vengeur', 4, 254259),
(-79789, 'inv_lightforgedelegg_amethyst', 153043, 'Gangrebroyeur béni', 4, 254258),
(-79613, 'inv_ammo_bullet_07', 152912, 'Ortie d'eau douce', 4, 253711),
(-79595, 'inv_argusfelstalkermountred', 152905, 'Gueule-écumante cramoisie', 4, 253661),
(-79593, 'inv_argusfelstalkermountgrey', 152904, 'Crache-acide', 4, 253662),
(-79592, 'inv_argusfelstalkermountblue', 152903, 'Grince-dents caustique', 4, 253660),
(-79583, 'inv_mount_arcaneraven', 152901, 'Aile-ensorcelée pourpre', 4, 253639),
(-79487, 'inv_manaraymount_redfel', 152841, 'Raie de mana gangreluisante', ...)
```

Requête exécutée avec succès, 632 lignes modifiées. (0.039 s)

1) Permettre en modifiant une table l'ajout des montures.

```
ALTER TABLE personnages ADD mounts_id int(11)
```

Requête exécutée avec succès, 0 ligne modifiée. (0.009 s) [Modifier](#)

2) Attribuer la monture 29046 à Zakyku.

```
UPDATE personnages SET mounts_id=29046 WHERE identification=4
```

Requête exécutée avec succès, 0 ligne modifiée. (0.001 s) [Modifier](#)

3) Supprimer la monture 29046 de wow, la contrainte est-elle vérifiée

```
delete from mounts where mounts_creature_id=29046
```

Erreur dans la requête (1451): Cannot delete or update a parent row: a foreign key constraint fails (`beldjs`.`personnages`, CONSTRAINT `personnages_ibfk_3` FOREIGN KEY (`mounts_id`) REFERENCES `mounts` (`mounts_creature_id`))

PARTIE 4

1) Proposer une modélisation des données.

d) Table : `personnages_monture`(`mounts_perso_id`, `personnages_mounts_id`, `monture_mounts_id`)

Clef étrangère : identification
`mounts_creature_id`

2) Permettre en ajoutant une table et en modifiant la précédente l'ajout des montures.

```
CREATE TABLE IF NOT EXISTS MonturePerso (  
  mounts_perso_id INT primary key,  
  personnages_mounts_id INT(11),  
  monture_mounts_id INT(11),  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

3) Ajouter les contraintes d'intégrités fonctionnelles nécessaires.

La clé étrangère a été créée. 10:00:15 [Requête SQL](#)

```
ALTER TABLE `MonturePerso`  
ADD FOREIGN KEY (`personnages_mounts_id`) REFERENCES `personnages` (`identification`)
```

La clé étrangère a été créée. 10:06:35 [Requête SQL](#)

```
ALTER TABLE `MonturePerso`  
ADD FOREIGN KEY (`monture_mounts_id`) REFERENCES `montures` (`mounts_creature_id`) ON DELETE RESTRICT  
ALTER TABLE MonturePerso ADD CONSTRAINT PERSONNAGE_MOUNTS CHECK (personnages_mounts_id >= 0)
```

Requête exécutée avec succès, 0 ligne modifiée. (0.131 s) [Modifier](#)

```
ALTER TABLE MonturePerso ADD CONSTRAINT MONTURE_MOUNTS CHECK (monture_mounts_id >= 0)
```

Requête exécutée avec succès, 0 ligne modifiée. (0.138 s) [Modifier](#)

4) Attribuer une monture pour Daril, Dunbar et Jaco.

[Modifier](#) 09:17:46 `INSERT INTO MonturePerso VALUES (3 ,2 ,4271);`

[Modifier](#) 09:17:03 `INSERT INTO MonturePerso VALUES (2 ,3 ,3068);`

[Modifier](#) 09:15:44 `INSERT INTO MonturePerso VALUES (1 ,1 ,305);`

<input type="checkbox"/> Modification	<code>mounts_perso_id</code> ↓	<code>= rsonnages_mounts_id</code>	<code>monture_mounts_id</code>
<input type="checkbox"/> modifier	1	1	305
<input type="checkbox"/> modifier	2	3	3068
<input type="checkbox"/> modifier	3	2	4271

5) Attribuer plusieurs montures à Sperpinette et Saperlipopette .

Modifier 09:23:21 INSERT INTO MonturePerso VALUES (9 ,5 , 33316);

Modifier 09:23:03 INSERT INTO MonturePerso VALUES (8 ,5 , 31778);

Modifier 09:22:44 INSERT INTO MonturePerso VALUES (7 ,5 ,14543);

Modifier 09:21:46 INSERT INTO MonturePerso VALUES (6 ,7 ,14552);

Modifier 09:21:31 INSERT INTO MonturePerso VALUES (5 ,7 ,14548);

Modifier 09:21:23 INSERT INTO MonturePerso VALUES (4 ,7 ,14548);

<input type="checkbox"/> modifier	4	7	14333
<input type="checkbox"/> modifier	5	7	14548
<input type="checkbox"/> modifier	6	7	14552
<input type="checkbox"/> modifier	7	5	14543
<input type="checkbox"/> modifier	8	5	31778
<input type="checkbox"/> modifier	9	5	33316

