

```

<?xml version="1.0" encoding="UTF-8"?>
<bande-dessinée>
<titre>Moi, ce que j'aime, c'est les monstres</titre>
<auteur>
<prenom>Emil</prenom>
<nom>Ferris</nom>
<nationalite>US</nationalite>
<année-naissance>1962</année-naissance>
</auteur>
<!-- Pour le résumé, on choisit d'utiliser une représentation en HTML. Pour cela, on peut
utiliser le namespace du XHTML. -->
<résumé xmlns:html="http://www.w3.org/1999/xhtml">
<html:p>
<html:em>Karen Reyes</html:em> est une jeune fille de dix ans passionnée de monstres
qui habite à Chicago à la fin des années 1960.
Suite à la mort de sa voisine <html:em>Anka Silverberg</html:em>, Karen va décider de
mener l'enquête et découvrir un passé compliqué au coeur de l'Allemagne Nazi.
</html:p>
</résumé>
<personnages>
<personnage protagoniste="true">
<prenom>Karen</prenom>
<nom>Reyes</nom>
</personnage>
<personnage>
<prenom>Diego</prenom>
<nom>Reyes</nom>
<surnom>Deeze</surnom>
<lien>Frère</lien>
</personnage>
<personnage>
<prenom>Marvela</prenom>
<nom>Reyes</nom>
<lien>Mère</lien>
</personnage>
<personnage>
<prenom>Anka</prenom>
<nom>Silverberg</nom>
<lien>Voisine</lien>
</personnage>
<personnage>
<prenom>Sandy</prenom>
<lien>Camarade de classe</lien>
</personnage>
<personnage>
<nom>Gronan</nom>
<lien>Voisin</lien>
</personnage>
<personnage>
<prenom>Seamus</prenom>
<nom>Chuggez</nom>

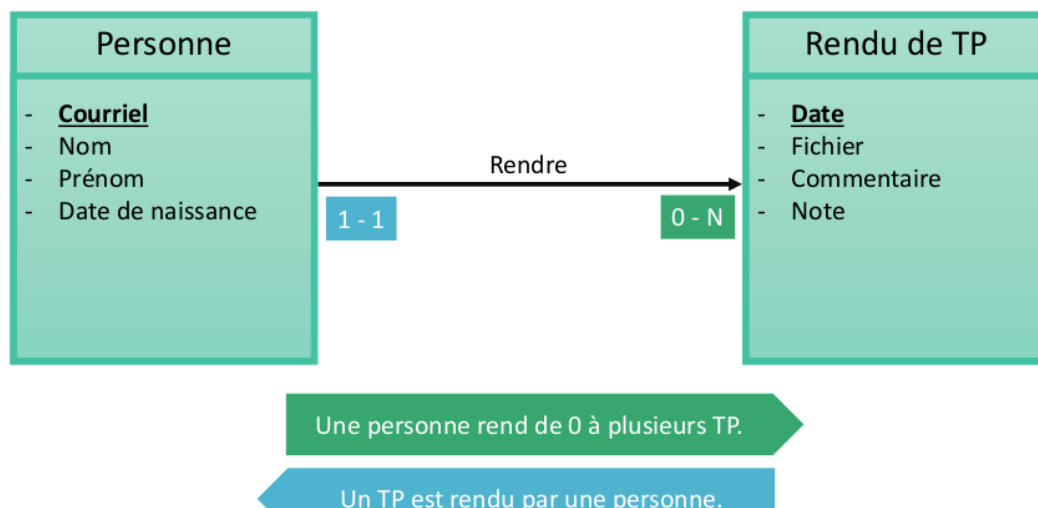
```

```

<nom>M. Chugg</nom>
<lien>Voisin</lien>
</personnage>
</personnages>
<éditions>
<édition originale="true" pays="US">
<isbn>978-1606999592</isbn>
<titre>My favorite thing is monsters</titre>
<éditeur>Fantagraphics Books</éditeur>
<date-publication>2017-02-28</date-publication>
<nombre-pages>386</nombre-pages>
</édition>
<édition pays="FR">
<isbn>979-1090724471</isbn>
<titre>Moi, ce que j'aime, c'est les monstres</titre>
<éditeur>Monsieur Toussaint Louverture</éditeur>
<date-publication>2018-08-23</date-publication>
<nombre-pages>416</nombre-pages>
</édition>
</éditions>
<récompense prix="Prix Eisner" catégorie="Meilleur album" année="2018" />
<récompense prix="Festival d'Angoulême" catégorie="Fauve d'or" année="2019" />
</bande-dessinée>

```

Cardinalité d'une relation



-- Artistes

```
SELECT * FROM artiste ORDER BY nom, prenom LIMIT 1;
SELECT * FROM artiste WHERE nationalite = 'FR' ORDER BY nom DESC, prenom DESC LIMIT 1;
SELECT prenom, nom FROM artiste WHERE prenom LIKE 'Jean%';
SELECT nom FROM artiste WHERE prenom IS NULL;
SELECT nationalite, count(*) FROM artiste GROUP BY nationalite HAVING count(*) > 1;
```

-- Usagers

```
SELECT count(*) FROM usager WHERE DAYOFWEEK(date_abonnement)=1;
SELECT nom, count(*) FROM usager group by nom order by count(*) desc;
SELECT * FROM usager WHERE DATE_ADD(date_abonnement, INTERVAL 1 YEAR) < NOW();
SELECT prenom, nom, DATE_ADD(date_abonnement, INTERVAL 1 YEAR) AS
date_renouvellement
FROM usager
WHERE DATE_ADD(date_abonnement, INTERVAL 1 YEAR) BETWEEN NOW() AND
DATE_ADD(NOW(), INTERVAL 1 WEEK)
ORDER BY date_abonnement;
```

-- Livres

```
SELECT DISTINCT(editeur) FROM livre WHERE rayonnage='Science-Fiction' ORDER BY
editeur;
select count(*) from livre WHERE serie in('Blueberry', 'Mister Blueberry');
SELECT titre, serie FROM livre WHERE serie IS NOT NULL and num_serie=1 AND rayonnage
in ('Comics', 'Bande dessinée', 'Manga');
```

-- emprunt

```
SELECT count(*) FROM emprunt WHERE date_rendu_reel IS NULL;
SELECT YEAR(date_rendu_attendu), count(*) FROM emprunt
WHERE date_rendu_reel > date_rendu_attendu
OR date_rendu_reel IS NULL AND date_rendu_attendu < NOW()
GROUP BY YEAR(date_rendu_attendu);
select id_usager, count(*) from emprunt GROUP BY id_usager, id_livre ORDER BY count(*)
DESC LIMIT 1;
```

-- mises à jour

```
INSERT INTO artiste(nom, prenom, nationalite) VALUES ('Le Tendre', 'Serge', 'FR');
UPDATE participation
SET id_artiste = (SELECT id FROM artiste WHERE nom='Le Tendre')
WHERE id_livre IN (SELECT id from livre WHERE serie = "La Quête de l'oiseau du temps")
AND role = 'Scénariste';
```

```
ALTER TABLE `artiste` ADD COLUMN `pseudonyme` VARCHAR(100) NULL AFTER
`nationalite`;
UPDATE artiste SET pseudonyme = 'J.K. Rowling' WHERE nom='Rowling';
UPDATE artiste SET pseudonyme = 'Boulet', nom = 'Roussel', prenom='Gilles' WHERE nom
= 'Boulet';
UPDATE artiste SET pseudonyme = 'Léo', nom = 'Oliveira (de)', prenom='Luiz Eduardo'
WHERE nom = 'Leo';
UPDATE artiste SET pseudonyme = 'Moebius' WHERE nom = 'Giraud';
```

```
UPDATE participation SET id_artiste = (SELECT id FROM artiste WHERE
pseudonyme='Mœbius')
WHERE id_artiste = (SELECT id FROM artiste WHERE nom='Mœbius');
DELETE FROM artiste WHERE nom='Mœbius';
```

-- complexe

```
SELECT id_livre, titre, count(*) FROM emprunt
JOIN livre ON emprunt.id_livre = livre.id
GROUP BY id_livre
ORDER BY count(*) desc LIMIT 10;
```

```
SELECT titre, serie FROM livre
LEFT JOIN emprunt ON livre.id = emprunt.id_livre
WHERE emprunt.id_livre IS NULL;
```

```
SELECT avg(nombre_emprunt) FROM (
SELECT nom, count(*) as nombre_emprunt FROM emprunt
JOIN usager ON emprunt.id_usager = usager.id
GROUP BY nom
) emprunt_famille;
```

```
SELECT nom FROM livre_genre
JOIN livre ON livre_genre.id_livre = livre.id
JOIN genre ON livre_genre.id_genre = genre.id
WHERE livre.titre = 'Le comte de Monte-Cristo I';
```

Afficher les genres des livres écrits par *Neil Gaiman*.

```
SELECT DISTINCT(genre.nom) FROM livre_genre
JOIN livre ON livre_genre.id_livre = livre.id
JOIN genre ON livre_genre.id_genre = genre.id
JOIN participation ON livre.id = participation.id_livre
JOIN artiste ON participation.id_artiste = artiste.id
WHERE artiste.nom='Gaiman';
```

```
SELECT DISTINCT(id_artiste), artiste.prenom, artiste.nom FROM participation
JOIN artiste ON participation.id_artiste = artiste.id
WHERE artiste.nom != 'Millar'
AND participation.id_livre IN
(
SELECT DISTINCT(livre.id) FROM livre
JOIN participation ON livre.id = participation.id_livre
JOIN artiste ON participation.id_artiste = artiste.id
WHERE artiste.nom = 'Millar' AND artiste.prenom='Mark'
);
```

```
SELECT prenom, nom, count(*) FROM (
SELECT artiste.id, artiste.prenom, artiste.nom FROM emprunt
JOIN participation ON emprunt.id_livre = participation.id_livre
JOIN artiste ON participation.id_artiste = artiste.id
GROUP BY emprunt.id_usager, emprunt.id_livre, date_emprunt, id_artiste
```

```

) auteur
GROUP BY id
ORDER BY count(*) DESC
LIMIT 10;

```

```
DELETE FROM person WHERE id = 10 ;
```

----- Fonctions de date et de temps -----

	Fonction	Exemple	Résultat
Année	YEAR(d)	YEAR('2019-09-14 12:05:23')	2019
Mois	MONTH(d)	MONTH('2019-09-14 12:05:23')	9
Jour dans le mois	DAY(d)	DAY('2019-09-14 12:05:23')	14
Heure	HOUR(t)	HOUR('2019-09-14 12:05:23')	12
Minute	MINUTE(t)	MINUTE ('2019-09-14 12:05:23')	5
Seconde	SECONDE(t)	SECOND ('2019-09-14 12:05:23')	23
Jour dans l'année	DAYOFYEAR(d)	DAYOFYEAR('2019-09-14 12:05:23')	257
Semaine dans l'année	WEEK(d)	WEEK('2019-09-14 12:05:23')	36
Jour dans la semaine	DAYOFWEEK(d)	DAYOFWEEK('2019-09-14 12:05:23')	7

```

{
  "titre": "Moi, ce que j'aime, c'est les monstres",
  "auteur": {
    "prenom": "Emil",
    "nom": "Ferris",
    "nationalite": "US",
    "anneeNaissance": 1962
  },
  "resume": "_Karen Reyes_ est une jeune fille de dix ans passionnée de monstres
qui habite à Chicago à la fin des années 1960. Suite à la mort de sa voisine
_Anka Silverberg_, Karen va décider de mener l'enquête et découvrir un passé
compliqué au coeur de l'Allemagne Nazi.",
  "personnages": [
    {
      "prenom": "Karen",
      "nom": "Reyes",
      "protagoniste": true
    },
    {
      "prenom": "Diego",
      "nom": "Reyes",
      "surnom": "Deeze",
      "lien": "Frère"
    },
    {
      "prenom": "Marvela",

```

```

    "nom": "Reyes",
    "lien": "Mère"
  },
  {
    "prenom": "Anka",
    "nom": "Silverberg",
    "lien": "Voisine"
  },
  {
    "prenom": "Sandy",
    "lien": "Camarade de classe"
  },
  {
    "nom": "Gronan",
    "lien": "Voisin"
  },
  {
    "prenom": "Seamus",
    "nom": "Chuggez",
    "surnom": "M. Chugg",
    "lien": "Voisin"
  }
],
}

```

titre: **Moi, ce que j'aime, c'est les monstres**

auteur:

prenom: **Emil**

nom: **Ferris**

nationalite: **US**

anneeNaissance: **1962**

resume: **_Karen Reyes_ est une jeune fille de dix ans passionnée de monstres qui habite à Chicago à la fin des années 1960. Suite à la mort de sa voisine _Anka Silverberg_, Karen va décider de mener l'enquête et découvrir un passé compliqué au coeur de l'Allemagne Nazi.**

personnages:

- prenom: **Karen**

nom: **Reyes**

protagoniste: **true**

- prenom: **Diego**

nom: **Reyes**

surnom: **Deeze**

lien: **Frère**

- prenom: **Marvela**

nom: **Reyes**

lien: **Mère**

- prenom: **Anka**

nom: **Silverberg**

lien: **Voisine**

- prenom: **Sandy**

lien: **Camarade de classe**

- nom: **Gronan**

lien: **Voisin**

- prenom: **Seamus**

nom: **Chuggez**

surnom: **M. Chugg**

lien: **Voisin**

editions:
- originale: true
pays: US
isbn: 978-1606999592
titre: My favorite thing is monsters
editeur: Fantagraphics Books
datePublication: 2017-02-28
nombrePages: 386

Clé primaire

- L'identifiant d'un objet est appelé Clé primaire.
- La clé primaire peut être précisée dans la requête de création de table après la définition des colonnes avec l'instruction PRIMARY KEY.
- Il suffit de préciser la ou les colonnes constituant la clé primaire.

PRIMARY KEY ([colonne1], [colonne2])Exemple avec une clé primaire

```
CREATE TABLE person (  
  name  
  VARCHAR(50) NOT NULL,  
  firstname VARCHAR(45) NOT NULL,  
  birthdate DATE NULL,  
  admin  
  TINYINT NULL DEFAULT 0,  
  PRIMARY KEY (name, firstname)  
);
```

Auto-incrément

- Très fréquemment, la clé primaire des objets va être un champ technique géré par la base de données elle-même.
- MariaDB permet d'avoir une colonne de type entier avec un auto-incrément.
- Ainsi si aucune valeur n'est spécifiée pour cette colonne, la valeur sera un incrément de la valeur existante maximale.Exemple avec une clé primaire en auto-incrément

```
CREATE TABLE person (  
  id  
  INT NOT NULL AUTO_INCREMENT,  
  name  
  VARCHAR(50) NOT NULL,  
  firstname VARCHAR(45) NOT NULL,  
  birthdate DATE NULL,  
  admin  
  TINYINT NULL DEFAULT 0,  
  PRIMARY KEY (id)  
);
```

Relations conceptuelles vs relations pratiques

- Les relations de nos modélisations de données ne peuvent pas être

représentées telles quelles dans notre base de données.

- Il va falloir appliquer des règles pour pouvoir ajouter des colonnes et clés dans nos tables permettant de les représenter.
- Types de relation
- Suivant la cardinalité des relations, il est possible d'en extraire 3 types de relation. Ces 3 types se basent sur la cardinalité maximale de chaque côté de la relation.
- On a donc des relations :
 - De type 1-1
 - De type 1-N
 - De type N-N

Relation 1-N

- Dans une relation 1-N, le lien entre les 2 objets va être porté par l'objet « du côté N » de la relation.

Table 1

Table 2

1 -1

0 -N

- La clé primaire de la table 1 va être ajoutée dans une ou des colonnes supplémentaires de la table 2

Clé étrangère

- Pour faire le lien entre les 2 tables, une clé étrangère est ajoutée sur le champ référençant la clé primaire de l'autre table.

Colonne(s) locale(s)

CONSTRAINT [nom_cle]

FOREIGN KEY ([colonne1], [colonne2])

REFERENCES [table] ([colonne1], [colonne2])

Table distante

Colonne(s) distante(s)

- La référence doit forcément correspondre à une clé primaire. Exemple de clé étrangère

```
CREATE TABLE rendu_tp (  
  id INT NOT NULL AUTO_INCREMENT,  
  date DATE NOT NULL,  
  fichier BLOB NOT NULL,  
  id_auteur INT NOT NULL,  
  PRIMARY KEY (id),  
  CONSTRAINT fk_rendu_auteur  
  FOREIGN KEY (id_auteur) REFERENCES person (id)  
);
```

Clé étrangère et contrainte

- Une clé étrangère est une contrainte.
- La valeur de la colonne marquée comme une clé étrangère doit exister dans l'autre table.
 - En conséquence, il n'est pas possible de supprimer une ligne qui est pointée par une clé étrangère.
- Relation 1-1
- Une relation 1-1 va être construite comme une relation 1-N : une clé étrangère va être ajoutée dans l'une des 2 tables.
- Pour choisir dans quel objet positionner la clé étrangère, il faut réfléchir

à quel objet est le plus dépendant de l'autre, quel objet peut être supprimé sans impacter l'autre. Relation N-N

- Une relation N-N est un peu plus complexe à mettre en place car elle va nécessiter la création d'une table supplémentaire appelée table de correspondance.
 - La clé primaire de chacune des 2 tables est ajoutée à cette nouvelle table en tant que clé étrangère.
 - La clé primaire de la table de correspondance est la combinaison des 2 clés primaires. Relation N-N
- personne