MADAD Sarra
LOUATY Nour-El Houda
Classe 31.A

Rapport de Base de Données.
Projet Pokémon.

Note de début : j'ai pris le soin de vous mettre **en gras et en vert tous les numéros des questions** dans le rapport si vous souhaitez les visualiser rapidement ! :)

On commence par créer les tables :

- Equipe(idEquipe, nom, couleur)
- Joueur (pseudonyme, personnage, sexe, niveau, #idEquipe)
- Pokemon (idPokemon, nom, espece, pointCombat, #pseudonyme)
- Emplacement (idEmplacement, latitude, longitude)
- Arene (idArene, nom, #idEmplacement)
- Defense(#idEquipe, #idArene, dateControle)
- Apparition (#idPokemon, #idEmplacement, horaire, duree)

NB : J'ai volontairement changé les noms des clés primaires pour plus de clarté et éviter les confusions.
NB2 : La clé primaire de la table joueur est pseudonyme, on l'a retrouve sous le même nom en clé étrangère de la table Pokémon.
NB3 : le type date est de type jj-mmm-aaaa et les int sont remplacés par des decimal(10,0) ou autre pour plus de liberté sur les opérations arithmétiques futures.
NB4 : les seules erreurs ressortant dans le codes sont celles à expliquer dans le sujet, j'ai mis les commentaires qui vont avec.

On remplit les tables avec les données présentées ainsi qu'avec les données demandés dans les **questions 3, 4, 5, 6 et 7**.

Voici les tables terminées, affichés avec un SELECT * FROM table :
Vous pourrez voir le code PL/SQL utilisés pour créer mes tables et insérer les valeurs avec le fichier .sql, triés par question.

PSEUDONYME	IDEQUIPE	PERSONNAGE	SEXE	NIVEAU
Shadow	1	Smith	F	15
Root	2	Alice	F	15
Admin	1	Bob	M	1
Moustache	-	-	-	-
Flavius	2	Ruth	M	20
Asterix	1	Ruth	M	5

IDEQUIPE	NOM	COULEUR
1	Intuition	Jaune
2	Sagesse	Bleu
3	Bravoure	Rouge

Table Joueur et Équipe. (on voit bien que le niveau des joueurs féminins dans la table Joueur ont été mis à 15 (**question 7)**

IDPOKEMON	PSEUDONYME	NOM	ESPECE	POINTCOMBAT
1	Shadow	Bulbizarre	Graine	1071
25	Root	Pikachu	Souris	887
103	Admin	Noadkoko	Fruitpalme	190
150	Root	Mewtwo	Génétique	4144
107	Shadow	Tygnon	Puncheur	204
19	Admin	Rattata	Souris	20
39	Moustache	Rondoudou	Bouboule	4145

IDEMPLACEMENT	LATITUDE	LONGITUDE
1	49.0350369	2.0696998
2	48.857848	2.295253
3	-74.0445	40.6892

Table Pokémon et Emplacement.

IDARENE	IDEMPLACEMENT	NOM
1	3	Liberte
2	1	Lune
3	2	Star

IDARENE	IDEQUIPE	DATECONTROLE
1	1	10-OCT-16
2	1	01-SEP-16
3	2	10-OCT-16

Table Arène et Défense.

IDPOKEMON	IDEMPLACEMENT	HORAIRE	DUREE
25	3	25-OCT-16	3
1	2	09-OCT-16	10
107	3	02-OCT-16	5
103	1	25-OCT-16	15
25	1	01-SEP-16	20

Table Apparition.

Pour la **question 8**, on souhaite supprimer les pokémons dont l'espèce contient le mot « fruit ».
Comme SQL est sensible à la casse, il nous faut prendre en compte toutes les situations. Il faut donc tester avec le mot fruit en début de mot, en fin de mot, en milieu de mot, avec et sans majuscule.
DELETE FROM Pokemon WHERE espece LIKE 'fruit%';
DELETE FROM Pokemon WHERE espece LIKE 'Fruit%';
DELETE FROM Pokemon WHERE espece LIKE '%fruit%';

Néanmoins, nous avons une erreur et nous ne pouvons pas supprimer.
C'est normal que ce tuple est référencé dans d'autres tables, il y a donc des soucis de contrainte.
On ne peut pas supprimer ce tuple car elle contient une clé primaire qui a des 'enfants'. Il faut donc d'abord supprimer les tuples qui font référence à cette clé (ses 'enfants') dans les autres tables avant de supprimer celle-ci.

C'est la même chose pour la **question 9**. On nous demande de supprimer le joueur Admin.
DELETE FROM Joueur WHERE pseudonyme LIKE 'Admin';

On ne peut pas supprimer ce tuple car elle contient une clé primaire qui a des 'enfants'. Il faut donc d'abord supprimer les tuples qui font référence à cette clé (ses 'enfants') dans les autres tables avant de supprimer celle-ci.

À la **question 10**, on nous demande de sortir l'Arène dont le mot contient « Lune ». Ici encore et comme SQL est sensible à la casse, on prend en compte toutes les combinaisons.
SELECT * FROM Arene WHERE nom LIKE 'Lune%'
OR nom LIKE 'lune%'

OR nom LIKE '%lune%'

On a donc :

IDARENE	IDEMPLACEMENT	NOM
2	1	Lune

Pour la **question 11**, on nous demande les pokémons dont le nom commence par « p » sans tenir de la casse. Même principe pour que les deux précédents :
SELECT * FROM Pokemon WHERE nom LIKE 'P%'
OR nom LIKE 'p%'

On a donc :

IDPOKEMON	PSEUDONYME	NOM	ESPECE	POINTCOMBAT
25	Root	Pikachu	Souris	887

Pour la **question 12**, on souhaite les pseudonymes des joueurs qui ne contiennent pas la lettre « a ». Même principe pour la casse :
SELECT * FROM Joueur WHERE pseudonyme NOT LIKE '%a%'
AND pseudonyme NOT LIKE 'A%';

On a donc :

PSEUDONYME	IDEQUIPE	PERSONNAGE	SEXE	NIVEAU
Root	2	Alice	F	15

Pour la **question 13**, on veut trier les pokémons selon leurs points de combat par ordre décroissant.
SELECT * FROM Pokemon
ORDER BY pointCombat DESC;

On a donc :

IDPOKEMON	PSEUDONYME	NOM	ESPECE	POINTCOMBAT
39	Moustache	Rondoudou	Bouboule	4145
150	Root	Mewtwo	Génétique	4144
1	Shadow	Bulbizarre	Graine	1071
25	Root	Pikachu	Souris	887
107	Shadow	Tygnon	Puncheur	204
103	Admin	Noadkoko	Fruitpalme	190
19	Admin	Rattata	Souris	20

Pour la **question 14**, on veut calculer la durée moyenne d'apparition des pokémons. On utilise la fonction d'agrégation AVG qui calcule la moyenne.
SELECT AVG (duree) FROM Apparition;

On a donc :

AVG(DUREE)
10.6

Pour la **question 15**, on veut compter le nombre d'apparition des pokémons en octobre 2016. On se rappelle que le type date est de type jj-mmm-aaaa. Le type dans le SELECT est quant à lui légèrement différent, il faut faire attention.
SELECT COUNT(*) FROM Apparition WHERE horaire LIKE '%%-OCT-16';

On a donc :

COUNT(*)
4

Pour la **question 16**, il faut donner les noms des pokémons qui sont de la même espèce que Pikachu. Pour cela, on fait une requête imbriquée.
SELECT * FROM Pokemon WHERE espece LIKE (SELECT espece FROM Pokemon WHERE nom = 'Pikachu');

On a donc :

IDPOKEMON	PSEUDONYME	NOM	ESPECE	POINTCOMBAT
25	Root	Pikachu	Souris	887
19	Admin	Rattata	Souris	20

Pour la **question 17**, on veut les joueurs dont le niveau est supérieur au niveau moyen des joueurs. On fait donc une requête imbriquée avec la fonction d'agrégation AVG.
SELECT * FROM Joueur WHERE niveau > (SELECT AVG (niveau) FROM Joueur);

On a donc :

PSEUDONYME	IDEQUIPE	PERSONNAGE	SEXE	NIVEAU
Shadow	1	Smith	F	15
Root	2	Alice	F	15
Flavius	2	Ruth	M	20

On peut vérifier la requête précédente en comparant avec le niveau moyen des joueurs.
SELECT AVG (niveau) FROM Joueur;

AVG(NIVEAU)
11.2

Ils sont bien tous au dessus de 11,2. C'est bon !

Pour la **question 18**, on souhaite savoir quel est le pokémon le plus fort, donc avec le maximum de points de combat.
SELECT * FROM Pokemon WHERE pointCombat = (SELECT MAX (pointCombat) FROM Pokemon);

On a donc Rondoudou :

IDPOKEMON	PSEUDONYME	NOM	ESPECE	POINTCOMBAT
39	Moustache	Rondoudou	Bouboule	4145

Pour la **question 19**, on souhaite savoir combien d'arène l'équipe Intuition a contrôlé.
SELECT count(*) FROM Defense d, Equipe e WHERE e.nom LIKE 'Intuition' AND d.idEquipe=e.idEquipe;

On trouve donc :

COUNT(*)
2

Pour la **question 20**, on veut la date de la première apparition de Tygnon.
SELECT horaire FROM Apparition a, Pokemon p WHERE p.nom LIKE 'Tygnon' AND a.idPokemon = p.idPokemon;

On a donc :

HORAIRE
02-OCT-16

Pour la **question 21**, on veut les joueurs qui ont participé à la défense de l'arène Lune.
SELECT pseudonyme FROM Joueur j, Arene a, Defense d WHERE a.nom LIKE 'Lune' AND d.idArene = a.idArene AND d.idEquipe = j.idEquipe;

On a donc :

PSEUDONYME
Shadow
Admin
Asterix

Pour la **question 22**, on veut le nombre de pokémons de chaque joueur. On utilise ici une jointure gauche car on souhaite garder les joueurs qui n'ont pas attrapé de pokémons.
SELECT j.pseudonyme, count(p.idPokemon) FROM Joueur j left join Pokemon p ON p.pseudonyme=j.pseudonyme GROUP BY j.pseudonyme;

PSEUDONYME	COUNT(P.IDPOKEMON)
Admin	2
Root	2
Moustache	1
Shadow	2
Asterix	0
Flavius	0

On a donc :

Pour la **question 23**, on souhaite la moyenne des points de combat de chaque espèce de pokémons.

SELECT espece, AVG (pointCombat) FROM Pokemon GROUP BY espece;

On a donc :

ESPECE	AVG(POINTCOMBAT)
Puncheur	204
Bouboule	4145
Souris	453.5
Graine	1071
Génétique	4144
Fruitpalme	190

Pour la **question 24**, on souhaite afficher pour chaque joueur : son pseudonyme, son équipe et le nombre de pokémons qu'il possède. On fait ici une jointure complète car on souhaite garder les joueurs n'ayant pas d'équipe, ou les équipes n'ayant pas de joueurs.
SELECT j.pseudonyme, e.nom, count (p.pseudonyme)
FROM Joueur j full join Equipe e on j.idEquipe=e.idEquipe
full join Pokemon p on j.pseudonyme=p.pseudonyme
GROUP BY j.pseudonyme, e.nom;

On a donc :

PSEUDONYME	NOM	COUNT(P.PSEUDONYME)
Root	Sagesse	2
Flavius	Sagesse	0
Asterix	Intuition	0
-	Bravoure	0
Shadow	Intuition	2
Admin	Intuition	2
Moustache	-	1

Pour la **question 25**, on veut le nom et la durée totale d'apparition pour chaque pokémon.
SELECT p.nom, SUM(duree) FROM Pokemon p
FULL JOIN Apparition a ON a.idPokemon=p.idPokemon GROUP BY p.nom;

On a donc :

NOM	SUM(DUREE)
Tygnon	5
Mewtwo	-
Pikachu	23
Rattata	-
Bulbizarre	10
Noadkoko	15
Rondoudou	-

Pour la **question 26**, on veut le nom et la date de prise de contrôle la plus récente pour chaque arène.
SELECT a.nom, MAX(dateControle) FROM Arene a, Defense d WHERE d.idArene=a.idArene GROUP BY a.nom;

On a alors :

NOM	MAX(DATECONTROLE)
Star	10-OCT-16
Lune	01-SEP-16
Liberte	10-OCT-16

Pour la **question 27**, on veut le nom, l'espèce et le nombre d'apparition pour chacun des pokémons qui sont apparus au moins 2 fois.
SELECT p.nom, p.espece, count(a.idPokemon) FROM Pokemon p, Apparition a WHERE p.idPokemon=a.idPokemon HAVING count(a.idPokemon)>=2
GROUP BY p.nom, p.espece;

On a donc :

NOM	ESPECE	COUNT(A.IDPOKEMON)
Pikachu	Souris	2

Pour la **question 28,** on veut le nom, l'espèce et le nombre d'apparition de chacun des pokémons dont la durée est supérieure à 5 minutes.
SELECT p.nom, p.espece, a.duree FROM Pokemon p, Apparition a WHERE p.idPokemon=a.idPokemon AND a.duree>5;

On a donc :

NOM	ESPECE	DUREE
Bulbizarre	Graine	10
Noadkoko	Fruitpalme	15
Pikachu	Souris	20

Procédures et fonctions :

La procédure permettant de classer les pokémons en fonction du nombre d'apparitions (0 compris) ?

```
CREATE OR REPLACE PROCEDURE Classement_Pokemon
IS
BEGIN
   FOR ligne IN (
   SELECT nom, (SELECT COUNT(*) FROM Apparition a WHERE a.idPokemon = p.idPokemon) AS
Nombre_App
   FROM Pokemon p
   ORDER BY (SELECT COUNT(*) FROM Apparition a WHERE a.idPokemon = p.idPokemon) DESC
   ) LOOP
        dbms_output.put_line('Nom Pokémon : ' || ligne.nom || ' est apparu ' || ligne.Nombre_App || ' fois.');
        END LOOP;
```

```
END;
```

La fonction permettant de trouver le nombre de joueurs possèdent des pokémons qui sont placés dans l'emplacement avec la latitude la plus septentrionale (la plus haute) ?

```
CREATE OR REPLACE FUNCTION Nombre_Joueur
RETURN NUMBER
AS
   nombre NUMBER :=0;
BEGIN
   FOR ligne IN (
   SELECT j.pseudonyme FROM Joueur j, Pokemon p, Emplacement e, Apparition a
   WHERE j.pseudonyme = p.pseudonyme AND p.idPokemon = a.idPokemon AND e.idEmplacement =
a.idEmplacement
   AND e.latitude = (SELECT MAX(e2.latitude) FROM Emplacement e2)
   ) LOOP
      nombre := nombre + 1;
   END LOOP;
   RETURN(nombre);
END;
```

La fonction permettant de trouver l'équipe qui a pris le contrôle d'une arène plus souvent ?

```
CREATE OR REPLACE FUNCTION Equipe_Dominante
RETURN VARCHAR2
AS
   nom_Equipe VARCHAR2(20);
   num_Intuition NUMBER := 0;
   num_Sagesse NUMBER := 0;
   num_Bravoure NUMBER := 0;
BEGIN
   FOR row IN (SELECT idEquipe FROM Defense) LOOP
   IF row.idEquipe = 1 THEN
      num_Intuition := num_Intuition + 1;
   ELSIF row.idEquipe = 2 THEN
      num_Sagesse := num_Sagesse + 1;
   ELSE
      num_Bravoure := num_Bravoure + 1;
   END IF;
END LOOP;

IF num_Intuition > num_Sagesse AND num_Intuition > num_Bravoure THEN nom_Equipe := 'Intuition';
ELSIF num_Sagesse > num_Intuition AND num_Sagesse > num_Bravoure THEN nom_Equipe := 'Sagesse';
ELSIF num_Bravoure > num_Sagesse AND num_Bravoure > num_Intuition THEN nom_Equipe := 'Bravoure';
ELSE nom_Equipe := 'NULL';
END IF;
RETURN(nom_Equipe);
END;
```

La fonction permettant de trouver la plage des dates auxquelles les pokémons de l'équipe de la question précédente ont apparu ?

```
CREATE OR REPLACE FUNCTION Plage_de_Date(nom_Equipe VARCHAR2)
RETURN VARCHAR2
AS
   date_Pokemon VARCHAR2(50) := '';

BEGIN
   FOR ligne IN (SELECT horaire FROM Joueur j, Pokemon p, Equipe e, Apparition a
   WHERE e.nom = nom_Equipe AND j.idEquipe = e.idEquipe AND j.pseudonyme = p.pseudonyme AND
p.idPokemon = a.idPokemon)
   LOOP
      date_Pokemon := TO_CHAR(ligne.horaire)|| ' ' || date_Pokemon;
END LOOP;
RETURN(date_Pokemon);
END;
```