

# Projet de session



420-KGB-JQ -Gestion et exploitation de bases de données  
Semaine # 10

<b>Procédures stockées.....</b>	<b>1</b>
1. Création d'une base de données SQL Server.....	2
Création d'une base de données .....	2
Création des tables de la bd TennisX.....	2
Ajouter les données suivantes .....	4
Importer les données des tables .....	4
Relations entre les tables .....	5
Schéma relationnel de la bd .....	5
2. Création des vues .....	6
3. Les procédures stockées.....	7
Création de procédures stockées .....	7
4. Les procédures stockées et le traitement des erreurs.....	8
Mise au point des procédures stockées .....	9

- À remettre le lundi 19 décembre
- Travail individuel
- Pondération 30% de la note finale

# 1. Création d'une base de données SQL Server


## Création d'une base de données

1. Créer une base de données *TennisCodePerm* (ex : *TennisTrec....*)
  - 1.1. *TennisTrec* avec une taille de 10 Mo et une augmentation de 1 Mo, grandeur illimitée.
  - 1.2. *TennisTrec\_Log*, qui est le fichier journal, avec une taille de 5 Mo et une augmentation de 10%, taille maximale de 25 Mo.
2. Supprimer la base de données
3. Vérifier le bon fonctionnement de votre script en détruisant la base de données puis en la recréant avec votre script.

## Création des tables de la bd TennisX


Ajouter au script précédent les instructions qui permettent de créer les tables suivantes :

1. Création de 6 tables Joueurs. Chaque table a la définition suivante :

Table tblJoueurs???					
Description	Type de données	Taille	Obligatoire	Défaut	Contraintes
Numéro du joueur	 Entier	standard	Oui		
Nom du joueur	texte	30, variable	Oui		
Prénom du joueur	texte	30, variable	Oui		
Points du joueur	entier	standard	Non	0	≥ 0
Code de nationalité	texte	3, fixe	non		


- 1.1. tblJoueursAfrique
- 1.2. tblJoueursAmeriqueNord
- 1.3. tblJoueursAmeriqueSud
- 1.4. tblJoueursAsie
- 1.5. tblJoueursEurope
- 1.6. tblJoueursOceanie

2. Création de la table tblNationalite avec la définition suivante :

Table tblNationalite					
Description	Type de données	Taille	Obligatoire	Défaut	Contraintes
Code de nationalité	 texte	3, fixe	Oui		3 lettres *
Nom du pays	texte	50, variable	Oui		
Code de zone	entier	standard	Oui		


\* Validation généralement effectuée au niveau du programme d'application.

3. Création de la table tblZoneGeographique avec la définition suivante :

Table tblZoneGeographique					
Description	Type de données	Taille	Obligatoire	Défaut	Contraintes
Code de zone	 entier	standard	Oui		Entre 1 et 6 *
Nom de zone	texte	40, variable	Oui		





\* Validation généralement effectuée au niveau du programme d'application.

4. Création de la table tblTournoi avec la définition suivante :


Table tblTournoi					
Description	Type de données	Taille	Obligatoire	Défaut	Contraintes
Numéro du tournoi	 No auto	standard	Oui		
Nom du tournoi	texte	50, variable	Oui		
Ville du tournoi	texte	50, variable	Oui		
Pays du tournoi	texte	30, variable	Oui		
Bourse du tournoi	Monétaire		Non	50000	≥ 50 000 *

\* Validation généralement effectuée au niveau du programme d'application.


5. Création de la table tblResultats avec la définition suivante :

Table tblResultats					
Description	Type de données	Taille	Obligatoire	Défaut	Contraintes
Numéro du joueur	 entier	standard	Oui		
Numéro du tournoi	 entier	standard	Oui		
Code de résultat	 entier	standard	Oui		
Gains	Monétaire		Non		
Année du tournoi	 entier	standard	Oui		
Nombre de points	entier	standard	Non		



6. Création de la table tblTypeResultats avec la définition suivante :

Table tblTypeResultats					
Description	Type de données	Taille	Obligatoire	Défaut	Contraintes
Code de résultat	 entier	standard	Oui		
Nom du résultat	texte	30, variable	Oui		
Nombre de points	entier	standard	Oui		> 0
Taux de la bourse	réel	5 chiffres, 2 décimales	Oui		> 0 et ≤ 1

7. Création de la table tblCommanditaires avec la définition suivante :

Table tblCommanditaires						
.Description		Type de données	.Taille	.Obligatoire	.Défaut	Contraintes
Numéro commanditaire	du	 entier	standard	Oui		
Nom commanditaire	du	texte	50, variable	Oui		

8. Création de la table tblTournoiCommandite avec la définition suivante :

Table tblTournoiCommandite						
.Description		Type de données	.Taille	.Obligatoire	.Défaut	Contraintes
Numéro commanditaire	du	 entier	standard	Oui		
No du tournoi		 entier	standard	Oui		

## Ajouter les données suivantes

1. Ajouter les données suivantes à la table tblZoneGeographique, ajouter à votre script.

codeZone	nomZone
1	Afrique
2	Amérique du Nord
3	Amérique du Sud
4	Asie
5	Europe
6	Océanie

2. Ajouter les données suivantes à la table tblTournoi, , ajouter à votre script.

noTournoi	nomTournoi	villeTournoi	paysTournoi	BourseTotal
1	Internationaux du Canada	Montréal	Canada	750000
2	Roland Garros	Paris	France	1250000
3	Wimbledon	Londres	Angleterre	1000000

## Importer les données des tables

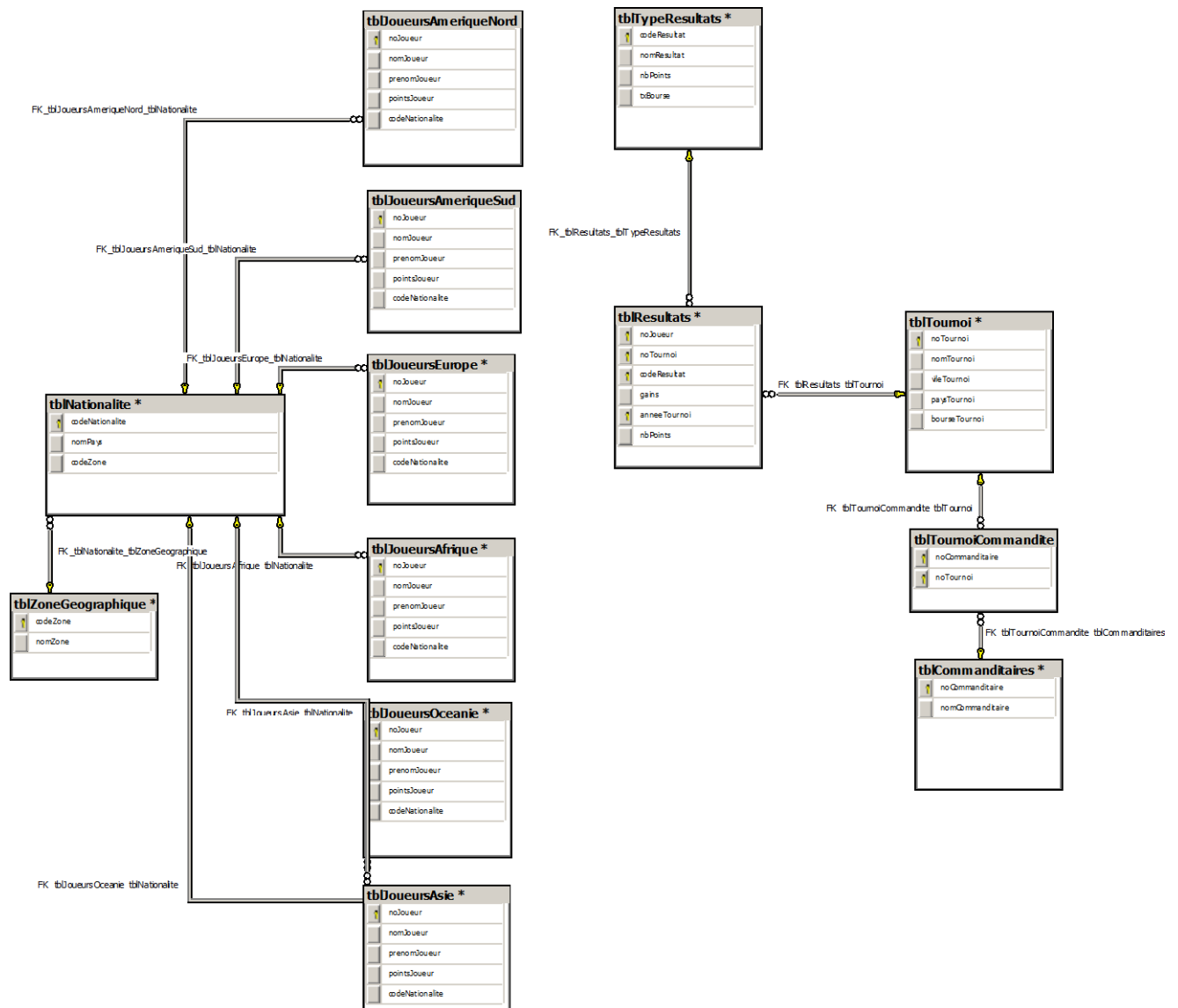
Sur le réseau, il y a un fichier Excel contenant les données des autres tables de la base de données : DonneesBdTennis.

## Relations entre les tables

Régler les contraintes de clés étrangères pour définir les relations entre les différentes tables, telles que représentées par le schéma suivant :

## Schéma relationnel de la bd

Créer le diagramme relationnel de votre base de données.



---

## 2. Création des vues

---

Effectuer la mise au point d'un script pour créer les objets de base de données suivants. (commandes T-SQL).

1. Nom : VueListeJoueurs  
Les joueurs sont répartis dans 6 tables. On voudrait consulter la liste de tous les joueurs en consultant une seule table « virtuelle ». Créer une vue qui génère la liste de tous les joueurs avec tous les champs.
2. Nom : VueClassementJoueurs  
Créer une vue pour visualiser le classement des joueurs. Les informations suivantes doivent être contenues dans cette vue : nom du joueur, prénom du joueur, total des gains, total des points.
3. Nom : VueNbJoueursParContinent  
Créer une vue pour visualiser le nombre de joueurs par continent. Les informations suivantes doivent être contenues dans cette vue : codeZone, nomZone, NombreJoueurs.
4. Nom : VueNbNationaliteParContinent  
Créer une vue pour visualiser le nombre de nationalités par continent. Les informations suivantes doivent être contenues dans cette vue : codeZone, nomZone, nombreNationalite
5. Nom : VueMoyenneBourseTournoi  
Créer une vue pour calculer la moyenne des bourses totales des tournois.
6. Nom : VueResultatsJoueurs  
Créer une vue pour visualiser les résultats de tous les joueurs, même ceux qui n'ont pas encore cumulé de gains / points. Les informations suivantes doivent être contenues dans cette vue : noJoueur, nomJoueur, prenomJoueur, nomTournoi, nomResultat, anneeTournoi, gains, nbPoints.

### 3. Les procédures stockées

#### Travail à faire:

- ❑ Procéder à la mise au point des procédures stockées suivantes.
- ❑ Conserver un script permettant de tester chaque procédure (.sql).

#### Tournoi de tennis Répartition des gains

Vainqueur	28%															
Finale	16%															
Demi	8%								8%							
Quart	4%				4%				4%				4%			
Huitièmes	2%		2%		2%		2%		2%		2%		2%		2%	
Seizièmes	0,5%	0,5%	0,5%	0,5%	0,5%	0,5%	0,5%	0,5%	0,5%	0,5%	0,5%	0,5%	0,5%	0,5%	0,5%	0,5%

#### Répartition des points

Vainqueur	200															
Finale	50															
Demi	20								20							
Quart	10				10				10				10			
Huitièmes	5		5		5		5		5		5		5		5	
Seizièmes	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

#### Création de procédures stockées

1. Procédure `psResultatsJoueurs`  
Créer une procédure qui récupère le nombre de points, les gains d'un joueur à partir de la table `tblResultat` en fonction de
  - 1.1. 3 paramètres en entrée : `noJoueur`, `noTournoi`, `anneeTournoi`
  - 1.2. 2 paramètres en sortie : `nbPoints`, `gains`
2. Procédure `psZoneGeographique`  
Créer une procédure qui récupère le `codeZone` de la `tblNationalite` en fonction du code nationalité.
  - 2.1. 1 paramètre en entrée : `codeNationalite`

2.2. 1 paramètre en sortie : codeZone

3. Procédure *psNomTableJoueur*

Créer une procédure qui récupère le nom de la table dans laquelle un joueur doit être inscrit selon son code nationalité. La procédure utilise la procédure *psZoneGeographique* qui retourne le code de la zone géographique selon le code de la nationalité

3.1. 1 paramètre en entrée : codeNationalite

3.2. 1 paramètre en sortie : nomTableJoueur

4. Procédure *psAjoutJoueur*

Créer une procédure qui permet d'ajouter un joueur dans la table correspondante à sa zone géographique. Les joueurs sont répartis en 6 zones géographiques.

4.1. 3 paramètres en entrée : nomJoueur, prenomJoueur, codeNationalite

4.2. Cette procédure utilise 2 procédures : *psNomTableJoueur* qui permet de récupérer le nom de la table en fonction du code de nationalité et *sp\_executeSQL* qui exécute une instruction SQL passée en paramètres

5. Procédure *psCalculGains*

Créer une procédure qui calcule les gains en fonction du résultat et de la bourse du tournoi

5.1. 2 paramètres en entrée : codeResultat, noTournoi

5.2. 1 paramètre en sortie : gains

6. Procédure *psCalculPoints*

Créer une procédure qui calcule les points en fonction du résultat.

6.1. 1 paramètre en entrée : codeResultat

6.2. 1 paramètre en sortie : points

7. Procédure *psAjoutResultat*

Créer une procédure qui insert un résultat dans la table *tblResultat*

7.1. 4 paramètres en entrée : noJoueur, noTournoi, codeResultat, anneeTournoi

7.2. Cette procédure utilise 2 procédures : *psCalculGains*, *psCalculPoints*

---

## 4. Les procédures stockées et le traitement des erreurs

---

### Travail à faire:

- ❑ Procéder à la mise au point des procédures stockées en tenant compte des erreurs possibles et des situations anormales.



- ❑ À remettre un script contenant les procédures et les instructions de test pour correction.

## Mise au point des procédures stockées

---

1. Procédure *psResultatsJoueurs*
  - 1.1. Ajuster votre procédure pour tenir compte des éléments suivants : on a fourni un ensemble de paramètres pour lesquels il n'y a pas d'enregistrement dans la table *tblResultat* ou encore parce que ce joueur n'a pas participé à ce tournoi ;
  - 1.2. Votre script de test (qui simule le programme d'application) devra afficher le résultat escompté ou un message d'erreur approprié.
2. Procédure *psZoneGeographique*
  - 2.1. Ajuster votre procédure : prévoir le cas où l'on donne un code de nationalité qui n'existe pas.
  - 2.2. Votre script de test devra produire le résultat escompté ou encore afficher un message d'erreur approprié (code de nationalité inexistant).
3. Procédure *psNomTableJoueur*
  - 3.1. Cette procédure doit prévoir le cas que la procédure *psZoneGeographique* qu'elle a appelé lui retourne un code lui indiquant que le code de nationalité était inexistant.
  - 3.2. Votre script de test devra produire le résultat escompté ou encore afficher un message d'erreur approprié (code de nationalité inexistant).
4. Procédure *psAjoutJoueur*

Créer une procédure qui permet d'ajouter un joueur dans la table correspondante à sa zone géographique. Les joueurs sont répartis en 6 zones géographiques.

  - 4.1. Cette procédure doit prévoir le cas que la procédure *psNomTableJoueur* lui retourne un code de nationalité inexistant
  - 4.2. Votre script de test devra produire le résultat escompté ou encore afficher un message d'erreur approprié (code de nationalité inexistant).
5. Procédure *psCalculGains*
  - 5.1. Faire exécuter la procédure, retourner succès ou échec
  - 5.2. Votre script de test affiche le message approprié.
6. Procédure *psCalculPoints*
  - 6.1. Faire exécuter la procédure, retourner succès ou échec
  - 6.2. Votre script de test affiche le message approprié.
7. Procédure *psAjoutResultat*
  - 7.1. Faire exécuter la procédure, retourner succès ou échec
  - 7.2. Votre script de test affiche le message approprié.

## 5. Les déclencheurs

### Travail à faire:

- ❑ Procéder à la création des déclencheurs /triggers suivants
  - ❑ Construire également un script qui permettra de tester chaque déclencheur. Remettre pour correction.
1. Déclencheur AjoutTournoiCommanditaire  
Créer un déclencheur sur la table tblTournoi qui insert un enregistrement dans la table TournoiCommandite avec la liste de tous les commanditaires pour chaque insertion de Tournoi.  
Note : Tous les commanditaires commanditent un nouveau tournoi.
  2. Déclencheur Ajout Commanditaire Tournoi  
Créer un déclencheur sur la table tblCommanditaires qui insert un enregistrement dans la table TournoiCommandite avec la liste de tous les tournois pour chaque insertion de commanditaire.  
Note : Si on ajoute un commanditaire alors il commanditera tous les tournois.
  3. Déclencheur AjoutVueListeJoueurs  
Créer un déclencheur sur la vue VueListeJoueurs qui permet de saisir un nouveau joueur dans la vue et de l'insérer dans la bonne table.  
Note : Faites appel à votre procédure stockée psAjoutJoueur.
  4. Déclencheur SupprVueListeJoueurs  
Créer un déclencheur sur la vue VueListeJoueurs qui permet de supprimer un joueur à partir de cette vue et de supprimer les résultats du joueur dans la table tblResultat et de supprimer le joueur de sa table d'origine.
  5. Déclencheur ModifPointsVuelisteJoueurs  
Créer un déclencheur sur la vue VueListeJoueurs qui permet de modifier les points d'un joueur à partir de cette vue et de modifier les points dans la table d'origine.
  6. Déclencheur AjoutTotalJoueur  
Créer un déclencheur sur la table tblResultat qui ajoute le nombre de points au total du joueur lors de l'insertion d'un nouveau résultat dans la table tblResultat.
  7. Déclencheur SupprTotalJoueur  
Créer un déclencher sur la table tblResultat qui soustrait le nombre de points au total du joueur lors de la suppression d'un résultat dans la table tblResultat.