Abdelhakim EL HAFID

SGBD1 Contrôle
Exercice 2

```
if exists (select * from sys.databases where name='gestion_projet')
drop database gestion_projet;
```

```
--A. Créer la base de données :
create database gestion_projet;
use gestion_projet;
```

```
--1. Créer les tables et les contraintes :
create table Servicee (
	NumService int primary key identity(1,1),
	NomService varchar(30),
	DateCreation date
);
create table Employe (
	Matricule int primary key identity(1,1),
	Nom varchar(30),
	Prenom varchar(30),
	DateNaissance date,
	Adresse varchar(20),
	Salaire money,
	Grade varchar(20),
	NumService int foreign key(NumService) references Servicee(NumService)
);
create table Projet (
	NumProjet int primary key identity(1,1),
	NomProjet varchar(300),
	Lieu varchar(50),
	NbrLimiteTaches int,
	NumService int foreign key references Servicee (NumService)
);
create table Tache(
	NumTache int primary key,
	NomTache varchar(300),
	DateDebut date,
	DateFin date,
	Cout int,
	NumProjet int foreign key references Projet (NumProjet),
);
create table Travaille (
	Matricule int foreign key references Employe (Matricule),
	NumTache int foreign key references Tache (NumTache),
	NombreHeures int,
	primary key(Matricule, NumTache)
);
```

```
-- Insértions des données de test :
--Servicee (NumService, NomService, DateCreation)
insert into Servicee values
( 'Direction', '1991-11-25'),
( 'Secrétariat', '1999-01-18'),
( 'Ressources', '1999-02-20'),
( 'Développement', '1991-11-25'),
( 'Communication', '2000-11-11');
```

```sql
--Employe (Matricule, Nom, Prenom, DateNaissance, Adresse, Salaire, Grade, NumService)
insert into Employe values
( 'El madani', 'Hassan', '1991-11-25', 'Medina 7', 1000000000, 'Maître', 1),
( 'Mandor', 'Nour', '1999-01-18', 'Hayat Street 2', 100000, 'Or', 2),
( 'Korman', 'Mehdi', '2008-11-25', 'Rakiz 111', 10000, 'Or', 3),
( 'Zator', 'Abdelhadi', '2000-11-25', 'Romaya Avenue 12', 20000, 'Platine', 3),
( 'El maâlam', 'Hasna', '2002-11-25', 'Magid AB50', 200000, 'Diamond', 4),
( 'Safi', 'Karima', '2016-01-27', 'MID 9', 5000, 'Bronze', 5);
--Projet (NumProjet, NomProjet, Lieu, NbrLimiteTaches, NumService)
insert into Projet values
( 'Application mobile de gestion de temps', 'Département Principal', 500, 4),
( 'RPG en ligne multiplateforme', 'Département Principal', 1000, 4),
( 'Carapasse géante', 'Usine Principal', 2000, 3),
( 'Projet X', null, null, null);
--Tache(NumTache, NomTache, DateDebut, DateFin, Cout, NumProjet)
insert into Tache values
(1, 'Conception du modèle de données', '2016-01-01', '2016-11-11', 8000, 1),
(2, 'Character design du personnage joueur', '2016-07-17', '2016-11-27', 17000, 2),
(3, 'Programmation de l''environement', '2016-01-18', '2016-08-19', 300000, 2),
(4, 'Construction des bases de support', '2014-03-01', '2016-06-10', 9000000, 3),
(5, 'Conception des modèles introductifs', '2013-10-08', '2015-01-09', 17700, 3);
--Travaille (Matricule, NumTache, NombreHeures)
insert into Travaille values
(5, 1, 7560), (3, 2, 5789), (4, 2, 1287),
(4, 3, 1580), (5, 3, 100), (3, 3, 7890),
(6, 4, 120), (2, 4, 1240), (5, 5, 5899);
```

```sql
-- Les contraintes :

--La contrainte CK_Employe_dateNaissance : l'âge de l'employé doit être supérieur à 18.
-- Méthode 1 (moins précis car les mois et jours supplémentaire seront ignorés)
alter table Employe add constraint CK_Employe_dateNaissance
check ( datediff(year,DateNaissance,getdate()) >= 18 );
-- Méthode 2 (plus précis car l'année contient exactement 365.25 jours)
alter table Employe add constraint CK_Employe_dateNaissance
check ( (datediff(day,DateNaissance,getdate())/365.25) >= 18 );

--La contrainte CK_Tache_duree : une tâche a une durée minimale de 3 jours.
( Durée = Date_fin – Date_debut )
alter table Tache add constraint CK_Tache_duree
check( datediff(day,DateDebut,DateFin) >= 3);

--La contraint CK_Tache_cout : le côut minimal d'une tâche est de 1000DH par jour.
( cout >= (Date_fin – Date_debut)j x 1000 )
alter table Tache add constraint CK_Tache_cout
check ( cout >= datediff(day,DateDebut,DateFin)*1000 and cout >= 1000 )

--Les clés primaires sont incrémentées automatiquement sauf le numéro de la tâche.
--Déjà fait lors de la création des tables ☺

--le nom du projet doit être encodé en français et sensible à la casse.
alter table Employe alter column
Nom varchar(30) collate French_CS_AS; --Cette collation est sensible à la casse et aux
accents et insensible aux caractères Kana et à la largeur du code de représentation des
caractères.
```

--2. Ajouter le champ calculé âge à la table Employé.

```
alter table Employe add Age as
floor( datediff(day,DateNaissance,getdate()) / 365.25 ); --Arrondissement
```

--B. Requêtes de sélection :

--1. Afficher les employés dont le nom commence avec « El » et ne se termine pas par une lettre entre a et f, trier la liste par date de naissance.

```
-- Méthode 1
select * from Employe where Nom like 'El%[^a-f]' order by DateNaissance;
-- Méthode 2
select * from Employe where Nom like 'El%[g-z]' order by DateNaissance;
```

--2. Afficher les noms des taches (en majuscule) qui prendrons fin ce mois ci.

```
select upper(NomTache) as 'Nom de la tâche' from Tache
where month(DateFin) = month(getdate());
```

--3. Compter le nombre de grades différents de l'entreprise.

```
select count(distinct Grade) as 'Nombre des différents grades' from Employe;
```

--4. Afficher les employés qu'ont participé à un projet affecté à un service différent où il travaille.

```
select Emp.*
from Employe Emp
inner join Travaille Tra on Tra.Matricule = Emp.Matricule
inner join Tache Tch on Tch.NumTache = Tra.NumTache
inner join Projet Prj on Tch.NumProjet = Prj.NumProjet
where Emp.NumService != Prj.NumService;
```

--5. Afficher les projets avec une tache de durée inférieure à 30jours et une autre supérieure à 60jours (Durée d'une tache = Date de Fin – date de début)

```
select * from Projet where NumProjet in (
       select distinct NumProjet from Tache
       where datediff(day, DateDebut, DateFin) < 30
)
and NumProjet in (
       select distinct NumProjet from Tache
       where datediff(day, DateDebut, DateFin) > 60
);
```

--6. Afficher la masse horaire travaillée cette année (travaille débuter et terminer cette année) par projet (Masse horaire = somme (nombre_heure))

```
select Tch.NumProjet, sum(NombreHeures) as 'Masse horaire de cette année'
from Travaille Tra inner join Tache Tch on Tra.NumTache = Tch.NumTache
where year(DateDebut) = year(getdate()) and year(DateFin) = year(getdate())
group by NumProjet;
```

--7. Afficher le matricule et le nom des employés qui ont participé à la réalisation de plusieurs projets.

```
select Employe.Matricule, Employe.Nom
from Employe
inner join Travaille on Travaille.Matricule = Employer.Matricule
inner join Tache on Tache.NumTache = Travaille.NumTache
group by Employe.Matricule, Employe.Nom
having count(distinct Tache.NumProjet) >= 2;
```

--8. Afficher le matricule, le nom, la date d'anniversaire et l'adresse des employés qui vont fêter leur anniversaire la semaine prochaine.

```
select Matricule, Nom, dateadd(year, Age, DateNaissance) as 'Anniversaire', Adresse
from Employe
where dateadd(year, Age, DateNaissance)
between
	dateadd(day, -datepart(weekday,getdate())+1+7, getdate())
and
	dateadd(day, -datepart(weekday,getdate())+1+7+6, getdate());
```

--9. Afficher le(s) projet(s) qui se composent du plus grand nombre de tâches.

```
drop view TachesParProjet;
create view TachesParProjet as
	select NumProjet, count(distinct NumTache) as 'Nombre de tâches' from Tache
	group by NumProjet;

select * from Projet where NumProjet in (
	select NumProjet from Tache
	group by NumProjet
	having count(distinct NumTache) in
	(select max([Nombre de tâches]) from TachesParProjet)
);
```

--10. Afficher la durée de réalisation par projet (La durée de réalisation d'un projet = la date de fin de la dernière tache de ce projet – la date de début de la première tâche du projet (utiliser Min et Max))

```
select NumProjet, datediff(day, min(DateDebut), max(DateFin)) as 'Durée de réalisation'
from Tache
group by NumProjet;
```

--C. Requêtes de mise à jour :

--1. Modifier les salaires des employés selon la règle suivante:

- sans modification pour les employés âgés de moins de 58 ans,
- augmentation de 0.5% pour les employés âgés entre 58 et 60 ans,
- augmentation de 5% pour les employés âgés de plus que 60 ans.

```
update Employe set Salaire += (
	case
		when Age between 58 and 60 then 0.5*Salaire/100
		when Age > 60 then 5*Salaire/100
		else Salaire
	end
);
```

--2. Supprimer les taches non réalisées (une tache non réalisée est une tache dont la date de fin est dépassée sans qu'elle contienne un travail).

```
delete from Tache where DateFin < getdate()
and NumTache not in (select distinct NumTache from Travaille);
```

--D. Gérer la sécurité de la base de données :

--1. Créer les deux profils de connexion suivants:

- profil SQL server : CnxGestionnaire

```
create login CnxGestionnaire with password='abcd1234';
```

- profil Windows : ChefProjet-PC\ChefProjet

```
create login [ChefProjet-PC\ChefProjet] from windows;
```

--2. Créer un utilisateur au niveau de la base de données gestion_projet pour chaque profil:

- Gestionnaire

```
create user Gestionnaire from login CnxGestionnaire;
```

- ChefProjet

```
create user ChefProjet from login [ChefProjet-PC\ChefProjet];
```

--3. Attribuer les autorisations suivantes aux utilisateurs concernés:

- Gestionnaire :
  - le droit de mise à jour (insertion, modification et suppression) de toutes les tables sauf la table « employé ».

```
grant insert,update,delete on Servicee to Gestionnaire;
grant insert,update,delete on Projet to Gestionnaire;
grant insert,update,delete on Tache to Gestionnaire;
grant insert,update,delete on Travaille to Gestionnaire;
```

- ChefProjet
  - le droit de suppression de toutes les tables sauf la table « Employé »

```
grant delete on Servicee to ChefProjet;
grant delete on Projet to ChefProjet;
grant delete on Tache to ChefProjet;
grant delete on Travaille to ChefProjet;
```

  - le droit de modification du champ « adresse » de la table « Employé » (coupler avec un vue)

```
grant update (Adresse) on Employe to ChefProjet;
create view Vue_Adresse as
       select Adresse from Employe;
grant update on Vue_Adresse to ChefProjet;
```