

EFM SGBD1 2016

Compte Rendu

Créer la base de données :

```
create database ManifestationDb;  
use ManifestationDb;
```

Créer les tables :

```
create table Organisme (  
    id_org int primary key,  
    nom_org varchar(50),  
    typee varchar(50),  
    adresse varchar(100)  
);  
create table Lieu (  
    id_lieu int primary key,  
    nom varchar(50),  
    typee varchar(50),  
    adresse varchar(100)  
);  
create table MANIFESTATION (  
    id_ma int primary key,  
    nom_ma varchar(50),  
    date_ma date,  
    tarif money,  
    id_organisme int foreign key references Organisme (id_org) on delete set null on update  
cascade,  
    id_lieu int foreign key references Lieu (id_lieu) on delete set null on update cascade  
);  
create table Intervenant (  
    num_int int primary key,  
    nom varchar(50),  
    prénom varchar(50),  
    coordonnées varchar(100),  
    id_org int foreign key references Organisme (id_org) on delete set null on update cascade  
);  
create table INTERVENIR (  
    id_ma int foreign key references Manifestation (id_ma),  
    id_int int foreign key references Intervenant (num_int),  
    nb_heures int  
    primary key (id_ma, id_int)  
);  
create table FEEDBACK_visiteur (  
    id_vis int primary key,  
    texte varchar(100),  
    id_ma int foreign key references Manifestation (id_ma) on delete set null on update cascade  
);
```

Insérer des enregistrements :

```
insert into Organisme values  
(1, 'Organium Trix', 'Gentil', 'Jabjoj Area 9'),  
(2, 'Ragex Camp', 'Méchant', 'Binaryum 11'),  
(3, 'Lolalox ORGX', 'Gentil', 'Simaax Yiban 199');  
insert into Lieu values
```

```
(1, 'E-Corp Studio', 'Studio', 'Gin X-232'),
(2, 'Green of Hope', 'Jardin public', 'Razembul Got CT N10'),
(3, 'High SQ', 'Terrain publique', 'SDX Area 9 RT 19');
insert into MANIFESTATION values
(1, 'Manifestation de la liberté', '02-11-2016', 3000, 2, 1),
(2, 'Manifestation de la paix', '01-01-2016', 1000, 1, 1),
(3, 'Manifestation de la vengeance', '30-12-2016', 4000, 2, 3),
(4, 'Manifestation du réveil', '01-01-2017', 10000, 3, 2);
insert into Intervenant values
(1, 'El madani', 'Hassan', 'm@lwd21341', 1),
(2, 'El madani', 'Kenza', 'knz@121@DF341', 1),
(3, 'Saâdani', 'Mimoun', 'akjqsdf_ès', 2),
(4, 'Inami', 'Loubna', 'sdkjwdlqs', 2),
(5, 'Okab', 'Soror', 'm@qddsqd121341', 2),
(6, 'X', 'TheFighter', 'm@1213kqdsq41', 3);
insert into INTERVENIR values
(1, 2, 5), (2, 1, 11), (3, 1, 7), (4, 1, 3), (4, 3, 15),
(1, 3, 15), (2, 2, 10), (1, 1, 9), (4, 2, 5), (3, 2, 17);
insert into FEEDBACK_visiteur values
(1, 'Très intéressant', 1), (2, 'Nous devons agir', 1),
(3, 'Enlevons ces chaînes', 1), (4, 'Pour la liberté', 2),
(5, 'Oh yeah mais attention', 2), (6, 'De quelle liberté parlez-vous ?', 2),
(7, 'Je suis libre', 2), (8, 'OK!', 2);
```

Exercice 1 (9,5 pts) :

On considère que les tables MANIFESTATION, Organisme, Lieu, intervenant sont déjà créés.

1. Créer les tables FEEDBACK_visiteur et INTERVENIR sachant que : (2pts)

```
--Nb_heures : supérieur ou égale à 1 (1pt)
alter table INTERVENIR add constraint c_Nb_heures
check (Nb_heures >= 1);

--Texte : obligatoire (0,5pts)
alter table FEEDBACK_visiteur
alter column Texte varchar(100) not null;

--Id_vis : se compose de 2 lettres suivies de 5 chiffres (2pts)
alter table FEEDBACK_visiteur add constraint c_Id_Vis
check (Id_Vis like '[A-Z][A-Z][0-9][0-9][0-9][0-9][0-9]');
```

2. Modifier la table Manifestation pour que :

```
--La date prend par défaut la date système (2pts)
alter table Manifestation add constraint c_date
default getdate() for date_ma;

--Le tarif doit être positif (1pt)
alter table Manifestation add constraint c_tarif
check (tarif > 0);

--Ajouter un champ calculé TarifTTC = tarif + 20% (TVA) (1pt)
alter table Manifestation
add tarifTTC as tarif + (0.2*tarif);
```

Exercice 2 (23,5 pts) :

Formuler en SQL les requêtes suivantes :

1. Liste des manifestations indiquant les noms de l'organisme et du lieu triée par ordre chronologique inverse (1pt)

```
select id_ma, nom_ma, date_ma, tarif, nom_org, nom
from MANIFESTATION, Organisme, Lieu
where MANIFESTATION.id_organisme = Organisme.id_org
and MANIFESTATION.id_lieu = Lieu.id_lieu
order by date_ma desc;
```

2. Liste des manifestations qui vont avoir lieu dans 10 jours exactement (1pt)

```
select * from MANIFESTATION
where convert(date, dateadd(day, 10, getdate())) = date_ma;
```

3. Nombre de feedback par manifestation (1pt)

```
select id_ma, count(*) as 'Nombre de feedback' from FEEDBACK_visiteur
group by id_ma;
```

4. Liste des organismes des intervenants qui n'interviennent dans aucune manifestation (2pts)

```
select * from Organisme where id_org in (
    select id_org from Intervenent where num_int not in (
        select num_int from INTERVENIR
    )
)
```

5. Les intervenants qui sont intervenus dans toutes les manifestations (3pts)

```
select * from Intervenent where num_int in (
    select id_int from INTERVENIR
    group by id_int
    having count(*) = (select count(*) from MANIFESTATION)
)
```

6. Les intervenants qui font partie d'un organisme qui a organisé au moins 3 manifestations et qui sont intervenus sur au moins une manifestation pendant 3 heures (3pts)

```
select * from Intervenent where id_org in (
    select id_org from Organisme where id_org in (
        select id_organisme from MANIFESTATION
        group by id_organisme
        having count(*) >= 3
    )
)
and num_int in (
    select id_int from INTERVENIR
    where nb_heures = 3
)
```

7. L'intervenant ou Les intervenants qui ont fait le plus grand nombre d'heures totales d'intervention (3pts)

```
create view SommeNbrHeures
as
    select id_int, sum(nb_heures) as somme_heures
    from INTERVENIR
    group by id_int

select * from Intervenent where num_int in (
    select id_int from SommeNbrHeures where somme_heures = (
        select max(somme_heures) from SommeNbrHeures
    )
)
```

8. Liste des organismes des intervenants qui ont fait un nombre d'heures d'interventions inférieur à la moyenne du nombre d'heures d'interventions (3pts)

```
select * from Organisme where id_org in (
    select id_org from Intervenant where num_int in (
        select id_int from INTERVENIR
        group by id_int
        having sum(nb_heures) < (select avg(nb_heures) from INTERVENIR)
    )
);
```

9. Le lieu ou bien les lieux les plus utilisées (2 pts)

```
select * from Lieu where id_lieu in (
    select id_lieu from MANIFESTATION
    group by id_lieu
    having count(*) >= (
        select top 1 count(*) as nbr_manif
        from MANIFESTATION
        group by id_lieu
        order by nbr_manif desc
    )
);
```

10. Tous les couples possibles des intervenants qui sont dans le même organisme : Un couple ne peut pas être constitué d'une seule personne...) (2pts)

```
select concat(A.nom, ' ', A.prénom) as 'Premier intervenant',
       concat(B.nom, ' ', B.prénom) as 'Deuxième intervenant',
       A.id_org as 'Organisme commun'
from Intervenant A, Intervenant B
where A.num_int != B.num_int and A.id_org = B.id_org;
```

11. Créer une vue sur les enregistrements de la table lieu (0,5pts)

```
--Créer la vue
create view Vue_Lieu as
    select * from Lieu;

--Utiliser cette vue pour insérer un nouvel lieu (0,5pts)
insert into Vue_Lieu values (4, 'Nouveau lieu', 'Type lieu', 'Adresse lieu');

--Utiliser cette vue pour modifier l'adresse du lieu 234 (0,5pts)
update Vue_Lieu set adresse = 'Nouvelle adresse' where id_lieu = 234;

--Utiliser cette vue pour supprimer les lieux qui ont un type dont le nombre de caractères est inférieur ou égale à 3 (1pt)
delete from Vue_Lieu where len(typee) <= 3;
```

Exercice 3 (7 pts) :

1. Ajouter les utilisateurs suivants à la base de données événementielle en leur accordant les permissions spécifiées (3pts)

Login	Permissions
User1	Propriétaire de la base
User2	Consultation des champs : nom, prénom et id_org de l'Intervenant Mise à jour de la table organisme

```
--Ajouter les utilisateurs
create login login1 with password = 'abc123';
create login login2 with password = 'def456';
```

```
create user user1 for login login1;
create user user2 for login login2;

--Rendre User1 propriétaire de la base
exec sp_addrolemember 'db_owner', 'user1';

--Permettre à User2 de consulter les champs : nom, prénom et id_org de l'Intervenant
create view Vue_Intervenant as
    select nom, prénom, id_org from Intervenant;
grant select on Vue_Intervenant to user2;

--Permettre à User2 de mettre à jour de la table organisme
grant update on Organisme to user2;
```

2. Interdire à user1 la suppression dans la table intervenant (1pt)

```
revoke delete on Intervenant to user1;
```

3. Créer un rôle personnalisé « Mode » qui permet : (2pts)

```
--Créer le rôle
create role Mode;

--Permettre la mise à jour de la date ou tarif d'une manifestation
create view Vue_Manifestation as
    select date_ma, tarif from Manifestation;
grant update on Vue_Manifestation to Mode;

--Permettre la consultation de la table Manifestation
grant select on Manifestation to Mode;
```

4. Ajouter user2 à « Mode » (0,5 pts)

```
exec sp_addrolemember Mode, user2;
```

5. Supprimer l'utilisateur user2 du rôle « Mode » (0,5pts)

```
exec sp_droprolemember Mode, user2;
```