EFM SGBD1 2016

Compte Rendu

Créer la base de donnés :

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@1wd21341', 1),

(2, 'El madani', 'Kenza', 'knz@121@DF341', 1),

(3, 'Saâdani', 'Mimoun', 'axkjqsdf\_ès', 2),

(4, 'Inami', 'Loubna', 'sdkjwdlqs', 2),

(5, 'Okab', 'Soror', 'm@qdqsdq121341', 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 interessant', 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 Intervenant where num\_int not in (

select num\_int from INTERVENIR

)

)

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

select \* from Intervenant 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 Intervenant 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 Intervenant 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 évènementielle 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;