Question 5 : Le code sql permettant de programmer les relations :

```
set linesize 500;
alter session  set nls_date_format="DD/MM/YYYY HH24:MI";

drop table LesTickets;
drop table LesDossiers;
drop table LesDossiers;
drop table LesProgrammations;
drop table LesRepresentations;
drop table LesSpectacles;
drop table LesSieges;
drop table LesZones;
drop table LesCategories;

create table LesCategories (nomC varchar2(20) constraint c_pk primary key, prix
number(8,2) not null,
        constraint c_ck1 check(prix > 0),
        constraint c_ck2 check(nomC in ('orchestre','1er balcon', '2nd balcon',
'poulailler'))
        );

create table LesZones(noZone number(2) constraint z_pk primary key, nomC
varchar(30) not null,
        constraint z_fk foreign key (nomC) references LesCategories(nomC)
        );

create table LesSieges (noPlace number(4), noRang number(4), noZone
number(2),
        constraint s_pk primary key (noPlace,noRang),
        constraint s_fk foreign key (noZone) references LesZones(noZone)
        );

create table LesSpectacles(noSpec number(4) constraint sp_pk primary key,
nomS varchar(30) not null, duree number(4));
```

```
create table LesRepresentations(noSpec number(4), dateRep date,
        constraint r_pk primary key (noSpec, dateRep),
        constraint r_fk foreign key (noSpec) references LesSpectacles(noSpec)
        --constraint p_fk2 foreign key (dateRep) references
LesProgrammations(dateDebut)
        );

create table LesProgrammations( noSpec number(4), dateDebut date, dateFin
date, type varchar(30),
        constraint p_pk primary key (noSpec),
        constraint p_fk foreign key (noSpec) references LesSpectacles(noSpec)
        );

create table LesDossiers(noDossier number(5) constraint d_pk primary key,
montant number(6,2) not null,
        constraint d_ck check(montant > 0)
        );

create table LesTickets(noSerie number(4) constraint t_pk1 primary key, noSpec
number(4), dateRep date, noPlace number(4), noRang number(4), dateEmission
date not null, noDossier number(4),
        constraint t_pk2 unique (noSpec, dateRep, noPlace, noRang),
        constraint t_fk1 foreign key (noPlace,noRang) references
LesSieges(noPlace,noRang),
        constraint t_fk2 foreign key (noDossier) references
LesDossiers(noDossier),
        constraint t_fk3 foreign key (noSpec) references LesSpectacles(noSpec),
        --constraint t_fk4 foreign key (dateRep,NOSPEC) references
LesRepresentations(dateRep, NOSPEC),
        constraint t_ck2 check(dateEmission < dateRep),
        constraint t_ck3 check(noDossier > 0),
        constraint t_ck1 check(noSpec > 0),
        constraint t_ck4 check(noPlace > 0),
        constraint t_ck5 check(noRang > 0)
        );
```

```
insert into LesCategories select * from theatre.LesCategories;
insert into LesZones select * from theatre.LesZones;
insert into LesSieges select * from theatre.LesSieges;
insert into LesSpectacles select * from theatre.LesSpectacles;
insert into LesRepresentations select * from theatre.LesRepresentations;
insert into LesDossiers (noDossier, montant) select noDossier, sum(prix) as montant from theatre.LesTickets natural join theatre.LesSieges natural join theatre.LesZones
natural join theatre.LesCategories group by noDossier;
insert into LesTickets select * from theatre.LesTickets;
```

Question 6 : Les contraintes d'intégrités qui ne sont pas spécifié dans la question 4 et qui ne sont pas exprimés en SQL et à charge de l'utilisateur sont :

- Si spectacle est créer, il lui faut au moins une représentation
- Si une représentation est supprimée, les tickets pour celle-ci aussi
- Si un spectacle est supprimé, ces représentations aussi

Question 7 :

Pour le premier programme une des erreurs est lors de la première requête, aucun numéro de spectacle n'est donné en values (on part du principe que 120 étant la durée du spectacle) alors que c'est une clef primaire même si 120 était la clef no_spec, il manquerai la durée de la representation.

Dans un second temps , la table les representations faisant référence au spectacle, vu que la requete n'est pas passé, elle ne pass pas ici. On le voit très bien que cette contrainte a était imbriqué car la seconde requete est « executé » uniquement si la première est bonne (if et else).

Les erreur sont remonté ligne 19 et 35 du code, les rollback 21 et 37.

Dans la deuxième version on relève les même erreur sql mais désormais, PhP affiche le code et le message d'erreur qu'indique le serveur Oracle lorsque la requete est rejeté.

Dans la troisième version, on relève les même erreurs sql . Le php gère differement les erreur indiqué par le serveur Oracle lorsque la requete est rejeté. Désormais, un message d'erreur spécifique si le code « 1 » est remonté, (objet déjà présent dans la base de donnés). Ainsi, nous pouvons voir qu'en connaissant les différents code d'erreur d'Oracle, nous pouvons traité les erreur plus facilement.