

Project M1 - Advanced Databases

Network theatrical companies

Mohamed ZENATI
Anis IMANI

Objectif du projet

Le projet consistait à modéliser un réseau de compagnies théâtrales à l'aide de différents outils.

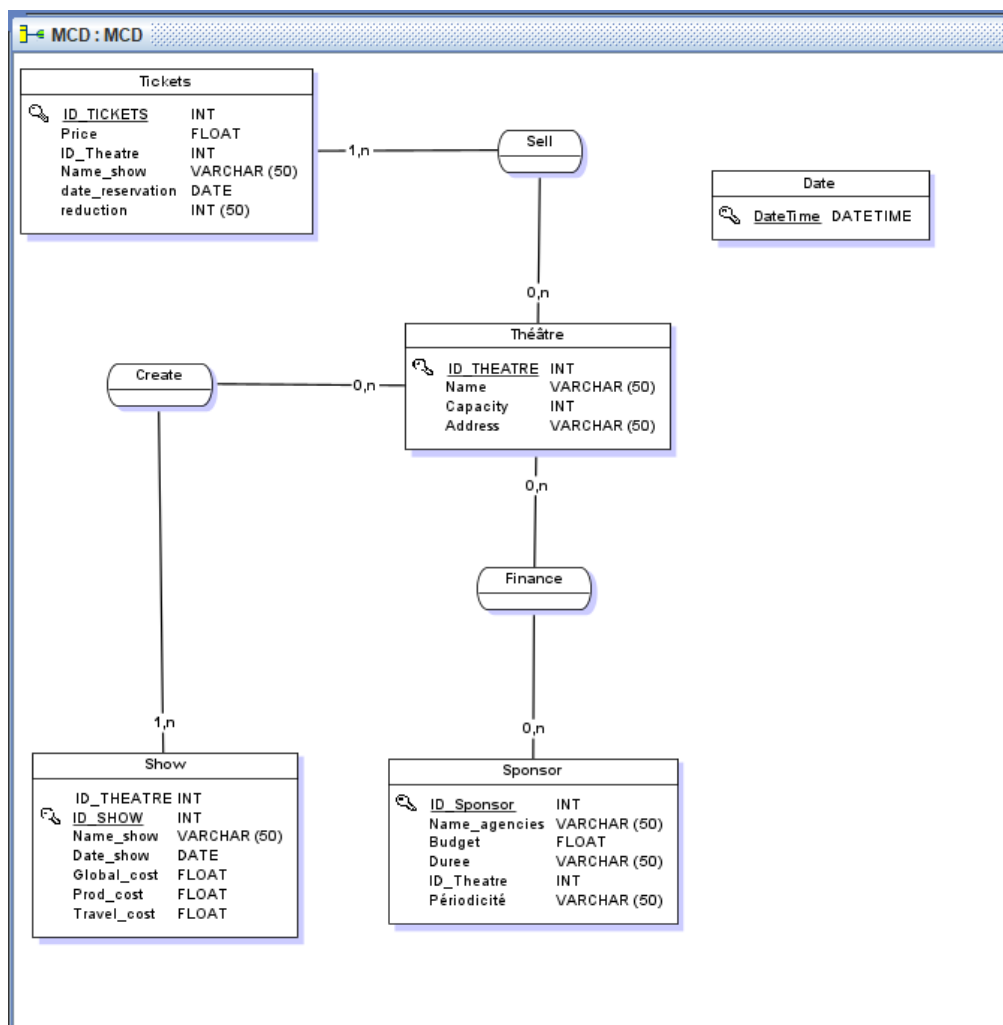
Outils utilisés

Jmerise – Oracle live – SQLite3

Étape du projet

1. Création d'un MCD sur Jmerise
2. Création de table sur SQL Workbench
3. Insertion des données pour chaque tables
4. Mise en place de requêtes et Trigger répondant à l'attente du manager

1. Création d'un MCD sur Jmerise

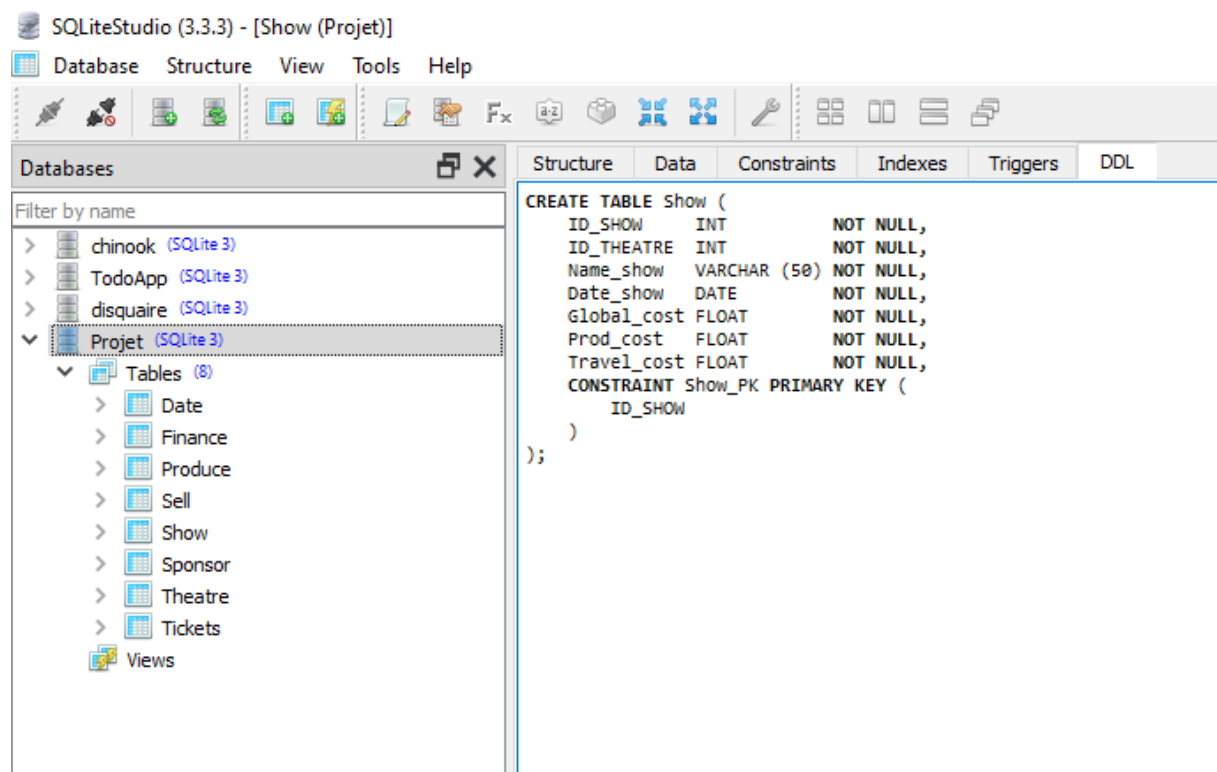


Le MCD est un modèle de donnée qui permet de décrire les données utilisées par le système d'information. Ici, le MCD a été réalisé sur JMerise. Nous avons créé 5 entités dont chacune possède ses propres attributs. Un attribut est une propriété d'une entité. Chaque entité est reliée à une autre à l'aide de cardinalités. Une cardinalité représente une association.

2. Création des tables

Une fois le MCD fait, il nous suffisait de créer les tables.

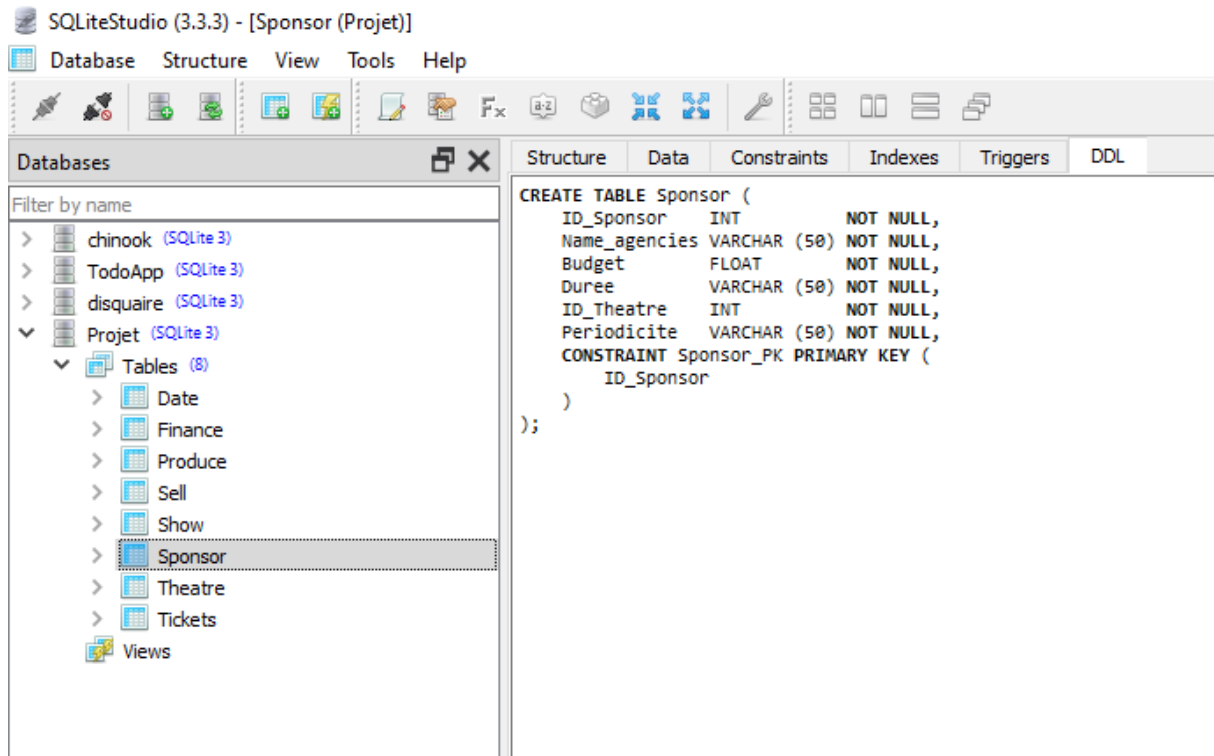
Table Show :



La table « Show » décrit les performances et les représentations des différentes compagnies de théâtre.

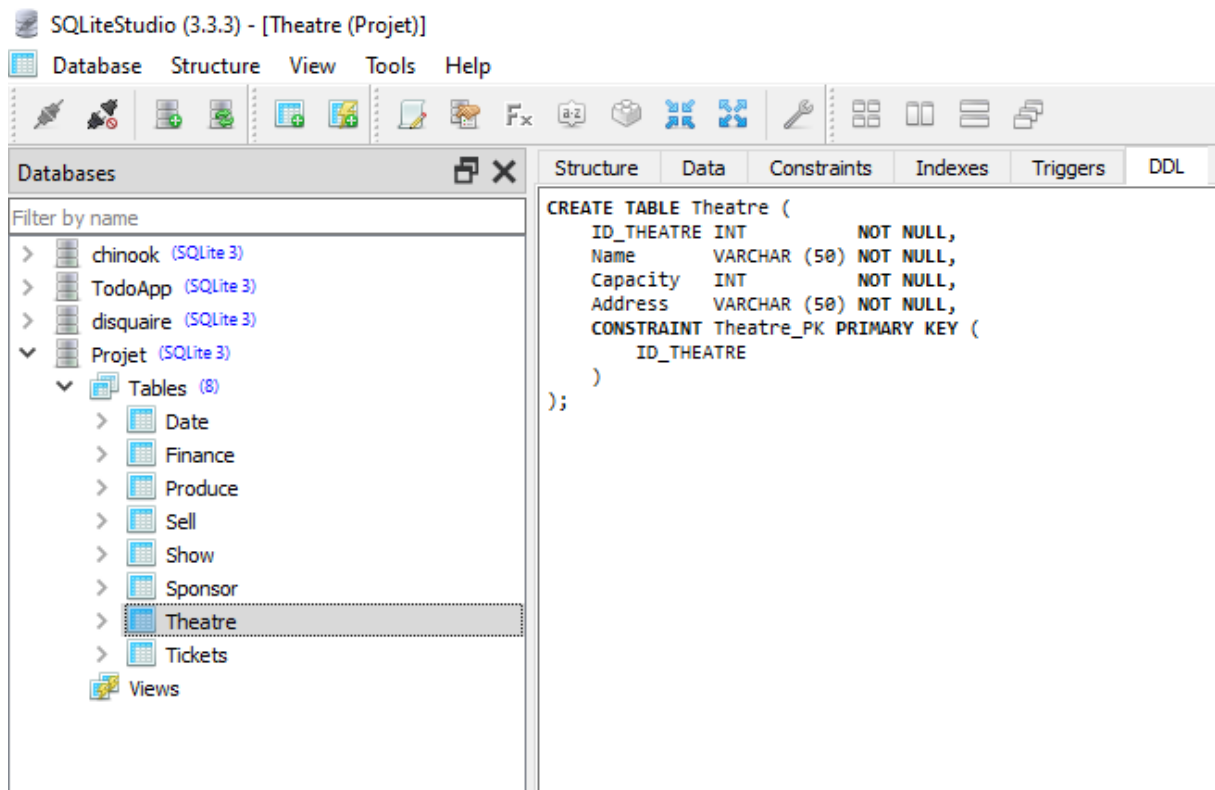
Cette table est définie par plusieurs attribut qui sont l'ID du show qui est la clé primaire, l'ID du théâtre qui organise la représentation, le nom du show, la date du show, le coût global que le théâtre fourni pour que le show ait lieu, le coût de production qui est versé dans l'achat de matériel et aux acteurs, le coût de déplacement lorsqu'un théâtre effectue un show dans un théâtre qui n'est pas le sien.

Table Show :



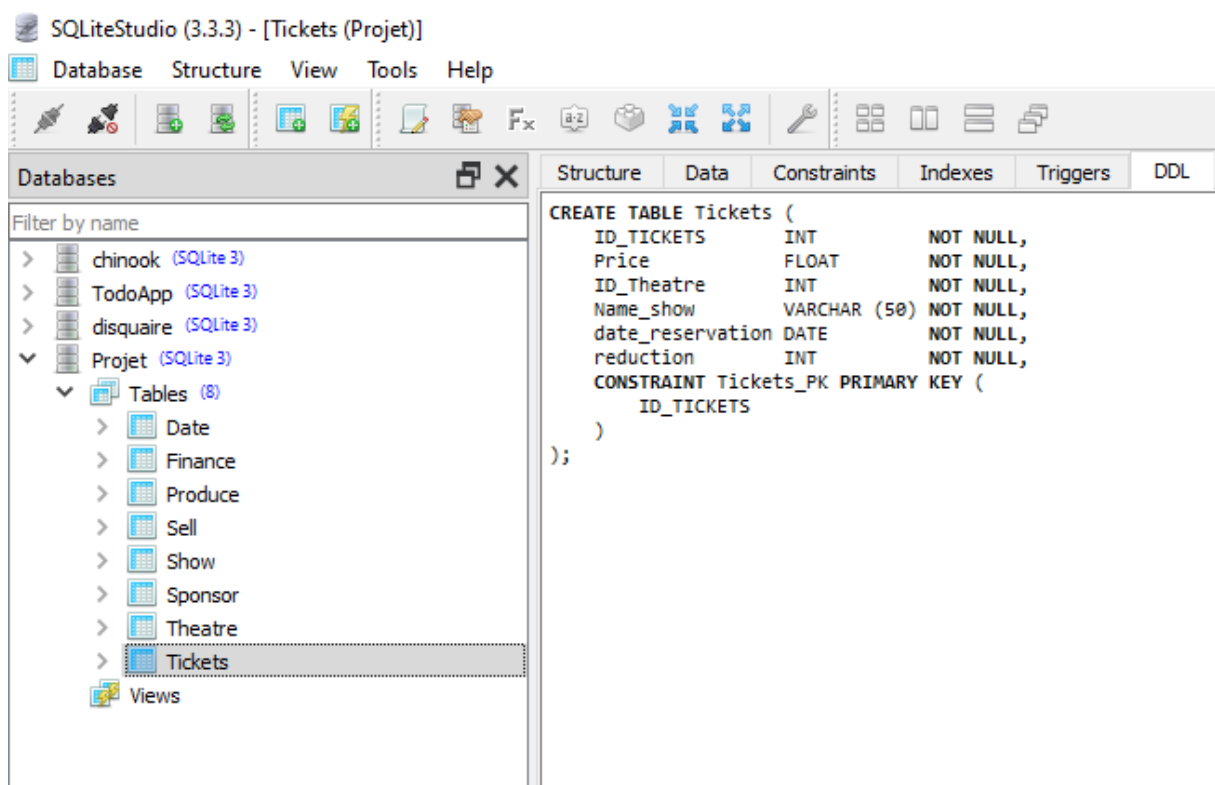
La table « Sponsor » nous donne des détails sur les agences qui aident financièrement. Les détails qui nous sont donnés sont l'ID de chaque sponsor, leurs noms, le budget qu'ils accordent, sur quel durée le budget est accordé, l'ID des théâtres que les agences aident financièrement et la période qui correspond à la fréquence dont le budget total est réparti.

Table Théâtre :

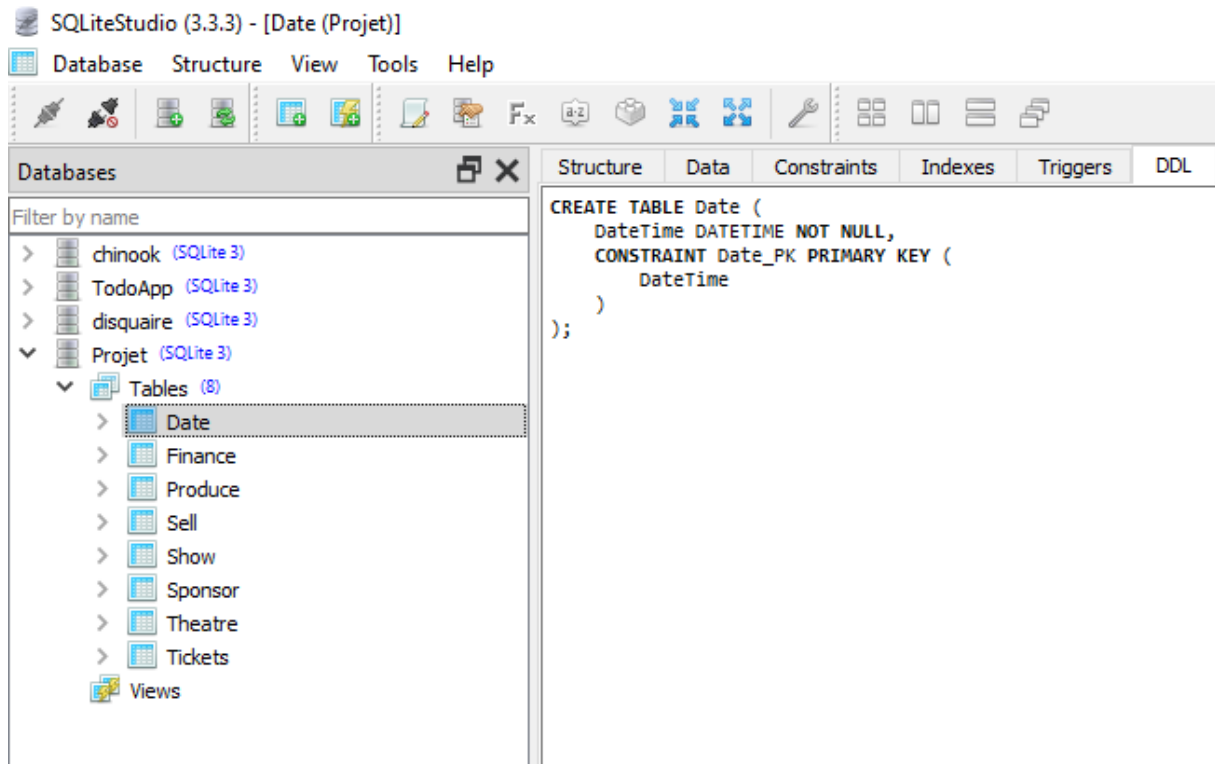


La table « Théâtre » représente la compagnie de théâtre. Elle est décrite par son ID qui est la clé primaire, son nom, sa capacité à accueillir les clients et une adresse.

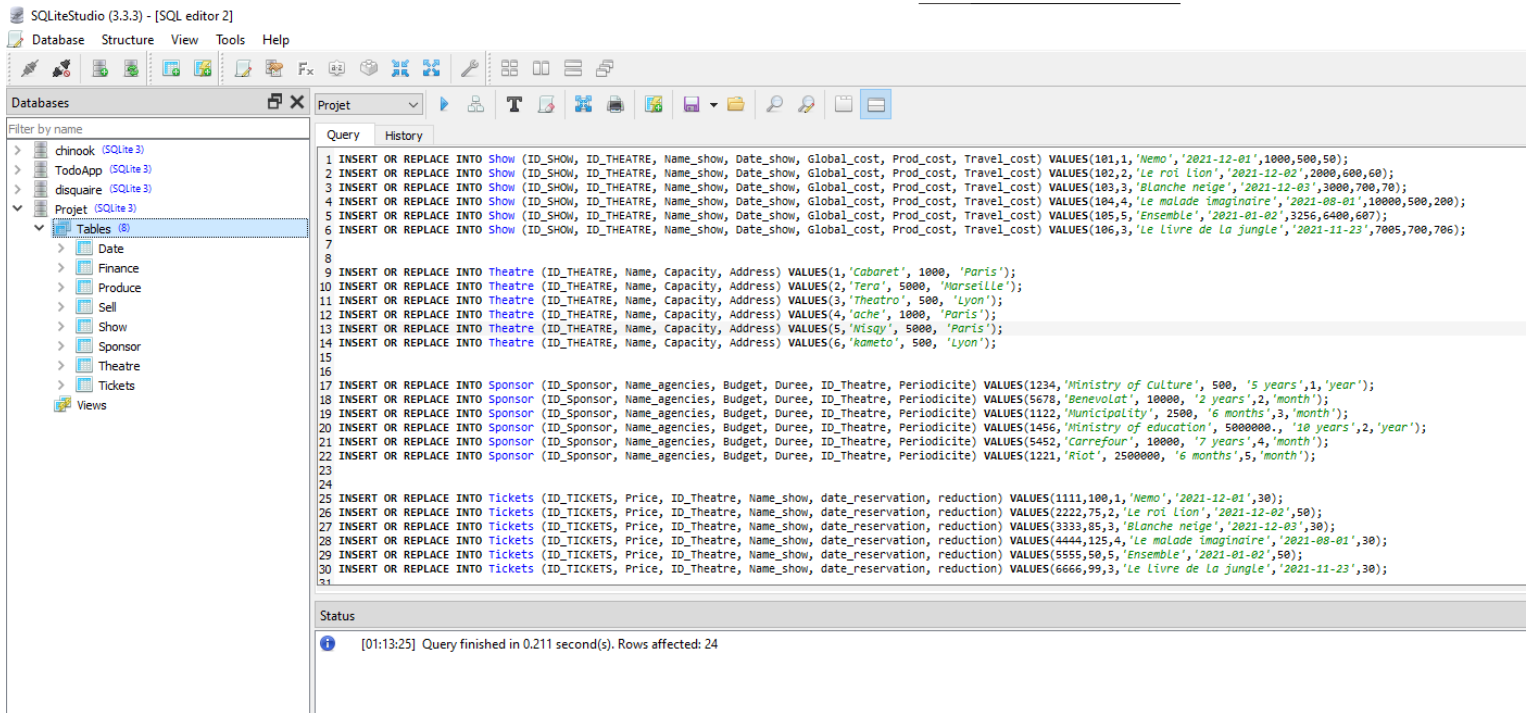
Table Tickets :



La table « Tickets » nous donne des détails sur les tickets en vente. Elle contient en clé primaire l’ID de chaque ticket, leurs prix, l’ID de chaque théâtre, nom des shows pour chaque ticket, la date de réservation et les réductions s’il y en a.

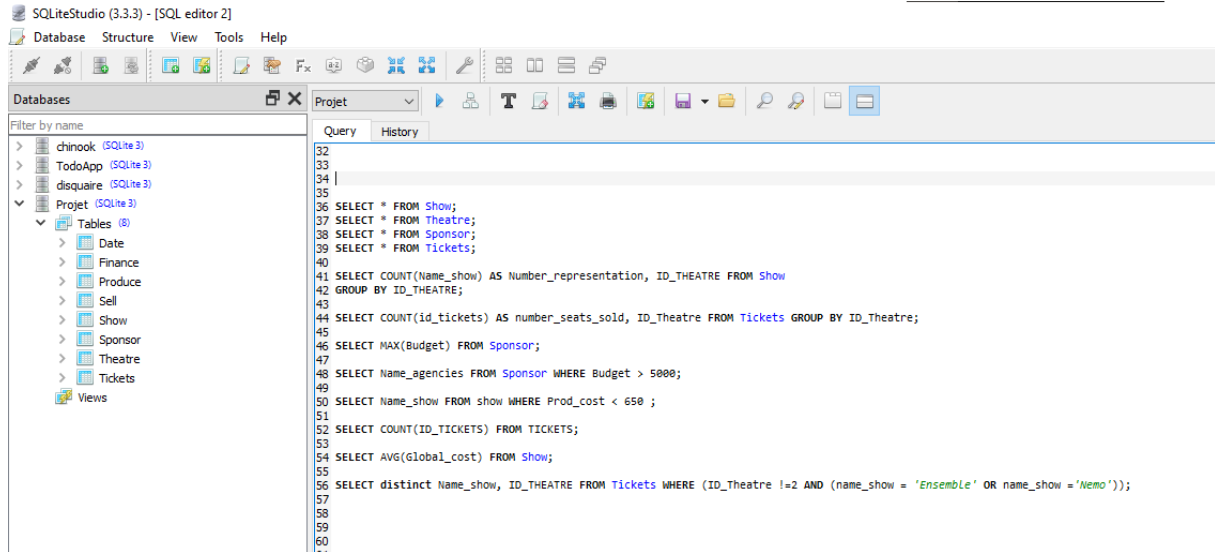


3. Insertion des données



Pour chaque table, nous avons insérés 6 lignes de données afin d’avoir différents cas possibles dans les requêtes qui sont demandés. En effet, à la main nous avons besoin de concentration pour faire en sorte d’avoir des données logiques et viable mais aussi, cette étape demandait énormément de temps. C’est pour ça que nous avons entré seulement 6 lignes de données pour chaque table, ce qui est très loin de la réalité.

4. Requêtes



```
35  
36 SELECT * FROM Show;  
37 SELECT * FROM Theatre;  
38 SELECT * FROM sponsor;  
39 SELECT * FROM Tickets;  
40
```

Ces 4 requêtes permettent d'avoir une vue d'ensemble sur les données contenu dans les tables.


```

40
41 SELECT COUNT(Name_show) AS Number_show, ID_THEATRE FROM Show
42 GROUP BY ID_THEATRE;
43

```

Grid view		Form view	
		1	
		Total rows loaded: 5	
Number show	ID THEATRE		
1	1		
2	1		
3	2		
4	1		
5	1		

Cette requête nous permet de connaitre le nombre de show joué dans chaque théâtre

```

42 GROUP BY ID_THEATRE;
43
44 SELECT COUNT(id_tickets) AS number_tickets_sold, ID_Theatre FROM Tickets GROUP BY ID_Theatre;
45

```

Grid view		Form view	
		1	
		Total rows loaded: 5	
number tickets sold	ID Theatre		
1	1		
2	1		
3	2		
4	1		
5	1		

Cette requête nous affiche le nombre de ticket vendu dans chaque théâtre

```
47
48 SELECT Name_agencies FROM Sponsor WHERE Budget > 5000;
49
```

Grid view

Form view

1

Total rows load

	Name agencies
1	Benevolat
2	Ministry of education
3	Carrefour
4	Riot

Cette requête nous présente les agences qui ont accordé un budget supérieur à 5000 euros.

```
49
50 SELECT Name_show FROM show WHERE Prod_cout < 650 ;
51
```

Grid view		Form view
		Total rows: 1
Name show		
1	Nemo	
2	Le roi lion	
3	Le malade imaginaire	

Cette requête nous montre les shows dont le coût de production est strictement inférieur à 650 euros.

```
55
56 SELECT distinct Name_show, ID_THEATRE FROM Tickets WHERE (ID_Theatre !=2 AND (name_show = 'Ensemble' OR name_show = 'Nemo'));
57
58
```

Grid view		Form view
		Total rows loaded: 2
Name show		ID THEATRE
1	Nemo	1
2	Ensemble	5

Cette requête nous permet de savoir si les shows n'ont pas été joué dans leur propre théâtre. Il suffit de changer le nom des show et l'ID théâtre pour connaître cette information sur d'autres show.

```
60
61 SELECT min(date_show), t.name_show, s.ID_THEATRE FROM Show s INNER JOIN Tickets t ON s.ID_THEATRE = t.ID_THEATRE
62 group by s.ID_THEATRE;
63
```

Grid view		Form view	
		Total rows loaded: 5	
min(date show)		name show	ID THEATRE
1	2021-12-01	Nemo	1
2	2021-12-02	Le roi lion	2
3	2021-11-23	Blanche neige	3
4	2021-08-01	Le malade imaginaire	4
5	2021-01-02	Ensemble	5

Cette requête permet de connaître le premier show effectué par chaque théâtre.

D'autres requêtes moins techniques ont été faites.

5. Trigger

Ce trigger évite les mêmes représentations à la même date.

SQL Worksheet

ClearFindActionsSaveRun

```
1 CREATE OR REPLACE TRIGGER check_date_show
2   before insert on Show
3   for each row
4   declare
5     ID_THEATRE INTEGER;
6   begin
7     select count(*) into ID_THEATRE from Show where ID_SHOW = :new.ID_SHOW and Date_show = :new.Date_show;
8
9     if ID_THEATRE > 0 then
10      RAISE_APPLICATION_ERROR(-20001,'ERROR : Not two representations at the same time. ');
11    end if;
12  end;
```

Trigger created.

Crédit en vente lors de l'achat d'un billet pour un show au théâtre

```
1 CREATE OR REPLACE TRIGGER credit_tickets
2   after insert on Tickets for each row
3   declare
4     ID_THEATRE INT;
5     sold int;
6   begin
7     select c into sold from (select count(*) as c from Tickets);
8     select ID_THEATRE into ID_THEATRE FROM Tickets where Tickets.ID_TICKETS = :new.ID_TICKETS;
9     insert into sells (transfer_id,amount,transfer_date,purpose,ID_THEATRE) values (sold,:new.price,:new.date_reservation,'sale of ticket ' || :new.ID_TICKETS || ' for the représentation : ' || :new.Name_show || ' ' Price : ' || :new.price, ID_THEATRE);
10  end;
```

Ce trigger évite d'ajouter un ticket pour un spectacle lorsqu'il est complet

```
1 CREATE OR REPLACE TRIGGER CapacityFull
2   before insert on Tickets
3   for each row
4   DECLARE
5     sold int;
6     Hall_ID INT;
7   full EXCEPTION;
8   BEGIN
9     SELECT ID_THEATRE into Hall_ID FROM Tickets WHERE Tickets.Name_show = :new.Name_show;
10    SELECT Capacity FROM Theatre WHERE Theatre.ID_THEATRE = ID_THEATRE;
11    SELECT COUNT(ID_TICKETS) into sold FROM Tickets WHERE Tickets.Name_show = :new.Name_show;
12
13    IF sold = Capacity THEN
14      RAISE full;
15    END IF;
16    EXCEPTION
17    WHEN full THEN
18      raise_application_error(-20001,'The show is full');
19  END;
```

Oracle

La syntaxe étant différente sur Oracle, voici ce qu'il faut écrire pour insérer des données sur Oracle :
(Il suffit d'ajoute « DATE » devant chaque date)

SQL Worksheet

ClearFindActionsSaveRun

```
1 INSERT INTO Show (ID_Show, ID_THEATRE, Name_show, Date_show, Global_cost, Prod_cost, Travel_cost) VALUES(101,1,'Nemo',DATE '2021-12-01',1000,500,500);
2 INSERT INTO Show (ID_Show, ID_THEATRE, Name_show, Date_show, Global_cost, Prod_cost, Travel_cost) VALUES(102,2,'Le roi lion',DATE '2021-12-02',2000,500,500);
3 INSERT INTO Show (ID_Show, ID_THEATRE, Name_show, Date_show, Global_cost, Prod_cost, Travel_cost) VALUES(103,3,'Blanche neige',DATE '2021-12-03',3000,700,700);
4 INSERT INTO Show (ID_Show, ID_THEATRE, Name_show, Date_show, Global_cost, Prod_cost, Travel_cost) VALUES(104,4,'Le malade imaginaire',DATE '2021-09-01',10000,500,500);
5 INSERT INTO Show (ID_Show, ID_THEATRE, Name_show, Date_show, Global_cost, Prod_cost, Travel_cost) VALUES(105,5,'Ensemble',DATE '2021-01-02',1006,4400,400);
6 INSERT INTO Show (ID_Show, ID_THEATRE, Name_show, Date_show, Global_cost, Prod_cost, Travel_cost) VALUES(106,6,'Le livre de la jungle',DATE '2021-11-23',7000,700,700);
7
8
9 INSERT INTO Theatre (ID_THEATRE, Name, Capacity, Address) VALUES(1,'Cabaret', 1000, 'Paris');
10 INSERT INTO Theatre (ID_THEATRE, Name, Capacity, Address) VALUES(2,'Tera', 5000, 'Marseille');
11 INSERT INTO Theatre (ID_THEATRE, Name, Capacity, Address) VALUES(3,'Theatro', 500, 'Lyon');
12 INSERT INTO Theatre (ID_THEATRE, Name, Capacity, Address) VALUES(4,'ache', 1000, 'Paris');
13 INSERT INTO Theatre (ID_THEATRE, Name, Capacity, Address) VALUES(5,'Hilary', 5000, 'Paris');
14 INSERT INTO Theatre (ID_THEATRE, Name, Capacity, Address) VALUES(6,'kaneto', 500, 'Lyon');
15
16 INSERT INTO Sponsor (ID_Sponsor, Name_agencies, Budget, Duree, ID_Theatre, Periodicite) VALUES(1129,'Ministry of Culture', 500, '5 years',1,'year');
17 INSERT INTO Sponsor (ID_Sponsor, Name_agencies, Budget, Duree, ID_Theatre, Periodicite) VALUES(5678,'Benevolat', 10000, '2 years',2,'month');
18 INSERT INTO Sponsor (ID_Sponsor, Name_agencies, Budget, Duree, ID_Theatre, Periodicite) VALUES(1122,'Municipalite', 2500, '6 months',3,'month');
19 INSERT INTO Sponsor (ID_Sponsor, Name_agencies, Budget, Duree, ID_Theatre, Periodicite) VALUES(1456,'Ministry of education', 5000000, '10 years',2,'year');
20 INSERT INTO Sponsor (ID_Sponsor, Name_agencies, Budget, Duree, ID_Theatre, Periodicite) VALUES(1456,'Ministry of education', 5000000, '10 years',2,'year');
21 INSERT INTO Sponsor (ID_Sponsor, Name_agencies, Budget, Duree, ID_Theatre, Periodicite) VALUES(1452,'Carrefour', 10000, '7 years',4,'month');
22 INSERT INTO Sponsor (ID_Sponsor, Name_agencies, Budget, Duree, ID_Theatre, Periodicite) VALUES(1221,'Riot', 2500000, '6 months',5,'month');
23
24 INSERT INTO Tickets (ID_TICKETS, Price, ID_Theatre, Name_show, date_reservation, reduction) VALUES(1111,100,1,'Nemo',DATE '2021-12-01',30);
25
```

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

© 2022 Oracle - Live SQL 22.1.1, running Oracle Database 19c Enterprise Edition - 19.0.0.0 - Database Documentation - Ask Tom - Dev Gym

Built with using Oracle APEX - Privacy - Terms of Use

Les triggers ont été fait exclusivement sur Oracle.