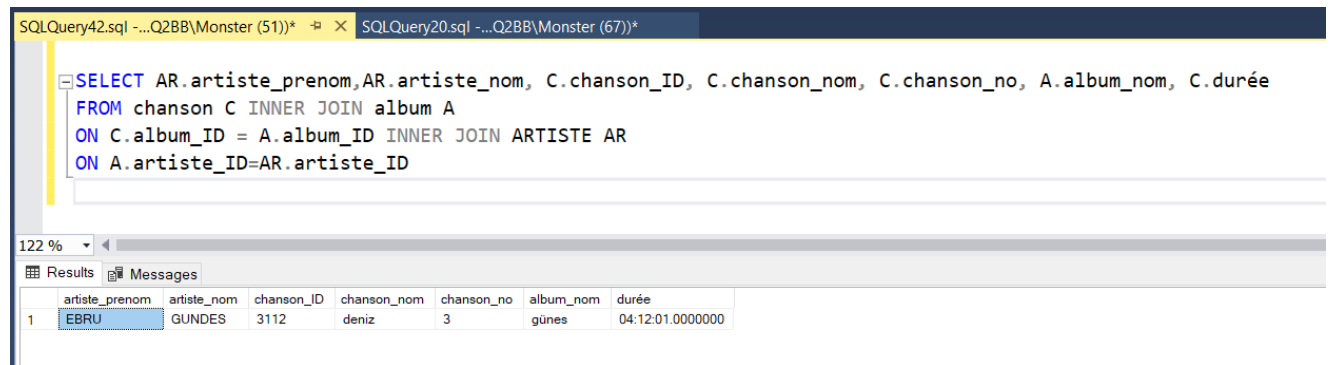


## REQUETES DE LISTES:

1)

```
SELECT AR.artiste_prenom, AR.artiste_nom, C.chanson_ID,  
C.chanson_nom, C.chanson_no, A.album_nom, C.durée  
FROM chanson C INNER JOIN album A  
ON C.album_ID = A.album_ID INNER JOIN ARTISTE AR  
ON A.artiste_ID=AR.artiste_ID
```



The screenshot shows a SQL query window with the following query:

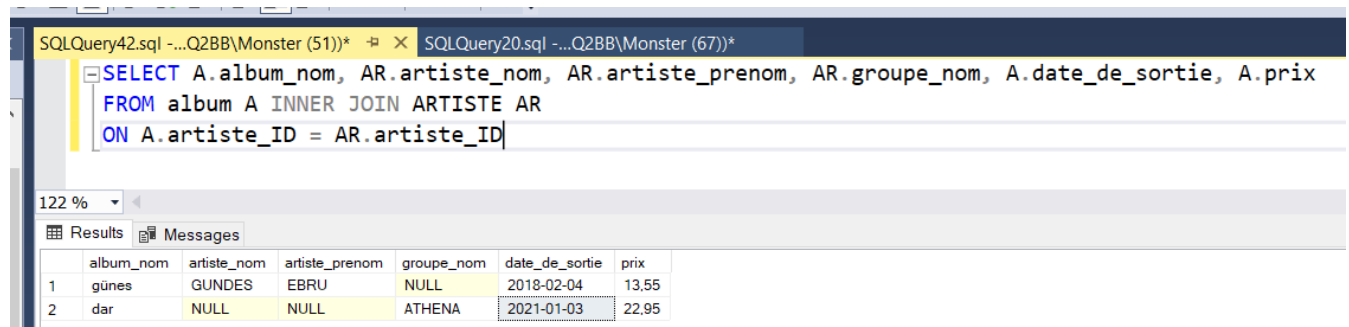
```
SELECT AR.artiste_prenom, AR.artiste_nom, C.chanson_ID, C.chanson_nom, C.chanson_no, A.album_nom, C.durée  
FROM chanson C INNER JOIN album A  
ON C.album_ID = A.album_ID INNER JOIN ARTISTE AR  
ON A.artiste_ID=AR.artiste_ID
```

The results pane shows a single row of data:

	artiste_prenom	artiste_nom	chanson_ID	chanson_nom	chanson_no	album_nom	durée
1	EBRU	GUNDES	3112	deniz	3	günes	04:12:01.0000000

2)

```
SELECT A.album_nom, AR.artiste_nom, AR.artiste_prenom,  
AR.groupe_nom, A.date_de_sortie, A.prix  
FROM album A INNER JOIN ARTISTE AR  
ON A.artiste_ID = AR.artiste_ID
```



The screenshot shows a SQL query window with the following query:

```
SELECT A.album_nom, AR.artiste_nom, AR.artiste_prenom, AR.groupe_nom, A.date_de_sortie, A.prix  
FROM album A INNER JOIN ARTISTE AR  
ON A.artiste_ID = AR.artiste_ID
```

The results pane shows two rows of data:

	album_nom	artiste_nom	artiste_prenom	groupe_nom	date_de_sortie	prix
1	günes	GUNDES	EBRU	NULL	2018-02-04	13.55
2	dar	NULL	NULL	ATHENA	2021-01-03	22.95

3)

```
SELECT C.client_nom, C.client_prenom, C.client_tel, A.ville,  
A.quartier  
FROM CLIENT C INNER JOIN [dbo].[ADRESSE] A  
ON C.adresse_ID= A.adresse_ID
```

SQLQuery42.sql -...Q2BB\Monster (51))\* SQLQuery20.sql -...Q2BB\Monster (67))\*

```

SELECT C.client_nom, C.client_prénom, C.client_tel, A.ville, A.quartier
FROM CLIENT C INNER JOIN [dbo].[ADRESSE] A
ON C.adresse_ID= A.adresse_ID

```

122 %

Results Messages

	client_nom	client_prénom	client_tel	ville	quartier
1	YILMAZ	ALI	5325460906	ISTANBUL	BEBEK
2	DONMEZ	YESIM	5526548796	ISTANBUL	CADDEBOSTAN
3	DONMEZ	SEVIM	5458760905	ISTANBUL	ETILER
4	CORNER	ALEX	5644426789	IZMIR	KARSIYAKA

4)

```

SELECT S.studio_nom, S.studio_tel, A.ville, A.quartier
FROM STUDIO S INNER JOIN [dbo].[ADRESSE] A
ON S.adresse_ID= A.adresse_ID

```

SQLQuery42.sql -...Q2BB\Monster (51))\* SQLQuery20.sql -...Q2BB\Monster (67))\*

```

SELECT S.studio_nom, S.studio_tel, A.ville, A.quartier
FROM STUDIO S INNER JOIN [dbo].[ADRESSE] A
ON S.adresse_ID= A.adresse_ID

```

122 %

Results Messages

	studio_nom	studio_tel	ville	quartier
1	NEWYORK STUDIO	5337222827	ISTANBUL	ETILER
2	CALIFORNIA STUDIO	5455678986	ISTANBUL	BEBEK
3	JERSEY STUDIO	5534568763	IZMIR	KARSIYAKA
4	TOKYO STUDIO	5457865759	ISTANBUL	KADIKOY

5)

```

SELECT A.auteur_nom, A.auteur_prénom, A.auteur_tel, AD.ville,
AD.quartier
FROM auteur A INNER JOIN [dbo].[ADRESSE] AD
ON A.adresse_ID= AD.adresse_ID

```

SQLQuery42.sql -...Q2B8\Monster (51))\* SQLQuery20.sql -...Q2B8\Monster (67))\*

```

SELECT A.auteur_nom, A.auteur_prénom, A.auteur_tel, AD.ville, AD.quartier
FROM auteur A INNER JOIN [dbo].[ADRESSE] AD
ON A.adresse_ID= AD.adresse_ID

```

122 %

Results Messages

	auteur_nom	auteur_prénom	auteur_tel	ville	quartier
1	YILMAZ	DOA	5312503010	ISTANBUL	YILDIZ
2	LIPA	DUA	5322344534	IZMIR	KARSIYAKA
3	AKKAYA	SINEM	5355423498	ISTANBUL	BESIKTAS
4	CELIKEL	EMRE	5435768765	ISTANBUL	KADIKOY
5	YILMAZ	IRMAK	5387650908	ISTANBUL	BEBEK

## PROCEDURES STOCKEES :

J'ai créé 4 procédures stockées pour la base de données. La première procédure est pour l'insertion d'une chanson à la table « chanson ». La deuxième procédure est pour insérer une commande à la table « commande » à condition que 'La date de mise d'une commande soit avant de la date d'accomplissement'. (C'est une contrainte d'intégrité sémantique). La troisième procédure est mise à jour de la table "album". La quatrième procédure est suppression d'un artiste de la table « Artiste ».

### REQUÊTES POUR L'INSERTION :

1)

```

CREATE PROCEDURE [dbo].[insert_chanson]
@chanson_ID int,
@chanson_nom nvarchar(50),
@chanson_no int,
@durée time(7),
@album_ID int,
@auteur_ID int,
@genre_ID int
AS
BEGIN
SET NOCOUNT ON
INSERT INTO chanson
(
chanson_ID,
chanson_nom,
chanson_no,
durée,
album_ID,
auteur_ID,
genre_ID
)

```

VALUES

```
(  
@chanson_ID, @chanson_nom, @chanson_no, @durée, @album_ID,  
@auteur_ID, @genre_ID  
)  
END
```

EXEC insert\_chanson

```
@chanson_ID= 432,  
@chanson_nom = 'tan',  
@chanson_no=5,  
@durée= '03:52:06',  
@album_ID=32,  
@auteur_ID=1,  
@genre_ID=5;
```

	chanson_ID	chanson_nom	chanson_no	durée	genre_ID	album_ID	auteur_ID
1	432	tan	5	03:52:06.0000000	5	32	1
2	3112	deniz	3	04:12:01.0000000	4	19	3

2)

create proc insert\_commande

```
(  
@commande_ID int,  
@date_de_mise datetime,  
@date_de_accomplissement datetime,  
@facture_ID int,  
@client_ID int)  
AS BEGIN
```

BEGIN TRAN Add\_Commande

insert into COMMANDE

```
(commande_ID,date_de_mise,[date_d'accomplissement],facture_ID,client  
_ID)
```

```
VALUES(@commande_ID,@date_de_mise,@date_de_accomplissement,@fac  
ture_ID,@client_ID)
```

SELECT \* FROM COMMANDE

```
IF @date_de_mise > @date_de_accomplissement  
BEGIN
```

```

        PRINT('La date de mise d'une commande doit être
avant de la date d'accomplissement.')
    ROLLBACK TRAN Add_Commande
    SELECT * FROM COMMANDE
END
ELSE
BEGIN
    COMMIT TRAN Add_Commande
    SELECT * FROM COMMANDE
END
END

```

SQLQuery55.sql -...Q2BB\Monster (68))\* SQLQuery52.sql -...Q2BB\Monster (74))\* SQL

```

EXEC insert_commande
@commande_ID =5,
@date_de_mise='20211012 23:02:12',
@date_de_accomplissement='20210912 23:02:12',
@facture_ID=13456,
@client_ID =4;

```

122 %

Results Messages

(1 row affected)

(4 rows affected)

La date de mise d'une commande doit être avant de la date d'accomplissement.

(3 rows affected)

Completion time: 2022-01-05T01:56:07.4925346+03:00

122 %

Results Messages

	commande_ID	date_de_mise	date_d'accomplissement	facture_ID	client_ID
1	1	2021-03-21 13:21:01.000	2021-04-25 09:05:03.000	1555	3
2	2	2021-01-21 13:28:01.000	2021-02-05 10:21:03.000	12345	1
3	3	2021-04-25 13:56:01.000	2021-05-17 15:12:03.000	14567	4
4	5	2021-10-12 23:02:12.000	2021-09-12 23:02:12.000	13456	4

	commande_ID	date_de_mise	date_d'accomplissement	facture_ID	client_ID
1	1	2021-03-21 13:21:01.000	2021-04-25 09:05:03.000	1555	3
2	2	2021-01-21 13:28:01.000	2021-02-05 10:21:03.000	12345	1
3	3	2021-04-25 13:56:01.000	2021-05-17 15:12:03.000	14567	4

## REQUÊTES POUR LA MISE A JOUR :

1)

```
SELECT TOP (1000) [album_ID]
, [album_nom]
, [date_de_sortie]
, [prix]
, [studio_ID]
, [artiste_ID]
FROM [Societe_d'Enregistrement].[dbo].[ALBUM]
```

122 %

Results Messages

	album_ID	album_nom	date_de_sortie	prix	studio_ID	artiste_ID
1	19	günes	2018-02-04	13,55	1	2
2	32	dar	2021-01-03	22,95	3	3

```
CREATE PROCEDURE update_album
(@album_ID int,
@album_nom nvarchar(50),
@date_de_sortie date,
@prix money,
@studio_ID int,
@artiste_ID int)
AS
BEGIN
UPDATE ALBUM
SET
album_nom = @album_nom,
date_de_sortie = @date_de_sortie,
prix =@prix, studio_ID = @studio_ID, artiste_ID = @artiste_ID
WHERE album_ID = @album_ID
END
```

```
EXEC update_album

@album_ID = 32,
@album_nom='dar',
@date_de_sortie= '20210103',
@prix =19.99,
@studio_ID=3,
@artiste_ID=3

SELECT * FROM ALBUM
```

22 %

Results Messages

	album_ID	album_nom	date_de_sortie	prix	studio_ID	artiste_ID
1	19	günes	2018-02-04	13,55	1	2
2	32	dar	2021-01-03	19,99	3	3

## REQUÊTE POUR LA SUPPRESSION :

1)

GO

CREATE PROC delete\_artiste

@artiste\_ID int

AS

BEGIN

DELETE ARTISTE WHERE artiste\_ID= @artiste\_ID

END

```
SELECT TOP (1000) [artiste_ID]
, [artiste_prenom]
, [artiste_nom]
, [groupe_nom]
FROM [Societe_d'Enregistrement].[dbo].[ARTISTE]
```

122 %

	artiste_ID	artiste_prenom	artiste_nom	groupe_nom
1	1	CAN	BONOMO	NULL
2	2	EBRU	GUNDES	NULL
3	3	NULL	NULL	ATHENA
4	4	ALICIA	KEYS	NULL
5	5	NULL	NULL	MAROON 5
6	6	ADAM	LEVINE	NULL
7	7	BILLIE	EILISH	NULL
8	9	DUA	LIPA	NULL
9	10	NULL	NULL	MANGA
10	15	Uzay	San	NULL
11	17	NULL	NULL	ADAMLAR
12	28	NULL	NULL	AYNA

SQLQuery24.sql - ...Q2/B8\monster (677) SQLQuery2

```
EXEC delete_artiste
@artiste_ID=28;

SELECT* FROM ARTISTE
```

122 %

	artiste_ID	artiste_prenom	artiste_nom	groupe_nom
1	1	CAN	BONOMO	NULL
2	2	EBRU	GUNDES	NULL
3	3	NULL	NULL	ATHENA
4	4	ALICIA	KEYS	NULL
5	5	NULL	NULL	MAROON 5
6	6	ADAM	LEVINE	NULL
7	7	BILLIE	EILISH	NULL
8	9	DUA	LIPA	NULL
9	10	NULL	NULL	MANGA
10	15	Uzay	San	NULL
11	17	NULL	NULL	ADAMLAR

## REQUÊTES POUR LE DÉCLENCHEUR:

1)

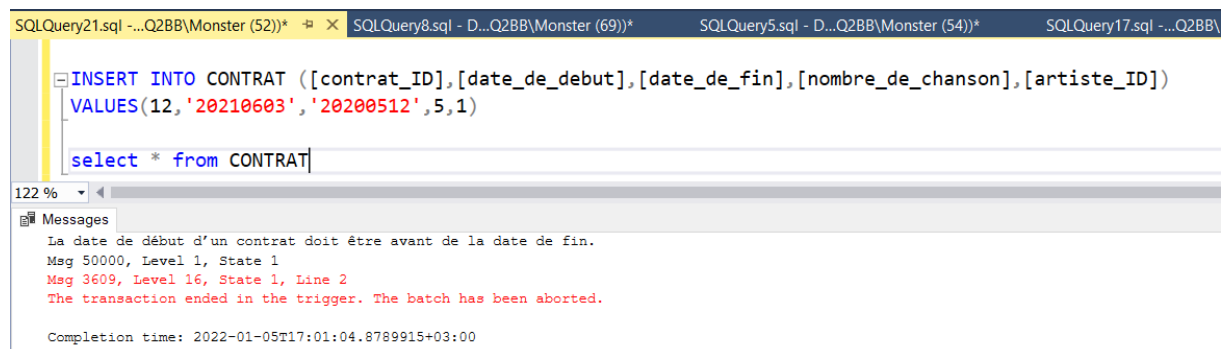
```
USE [Societe_d'Enregistrement]
GO
```

```
CREATE TRIGGER contrat_warning
ON CONTRAT
FOR INSERT
AS
BEGIN
```

```
    DECLARE @date_de_fin date
    DECLARE @date_de_debut date
    SELECT @date_de_debut= date_de_debut, @date_de_fin= date_de_fin
FROM inserted
```

```
    IF @date_de_debut >= @date_de_fin
        BEGIN
            RAISERROR('La date de début d'un contrat doit être
avant de la date de fin.',1,1)
            ROLLBACK TRAN
        END
    ELSE
        BEGIN
            PRINT(@date_de_debut)
            PRINT(@date_de_fin)
        END
END
```

END





SQLQuery21.sql - ...Q2BB\Monster (52))\* X SQLQuery8.sql - D...Q2BB\Monster (69))\* SQLQuery5.sql - D...Q2BB\Monster (54))\* SQLQuery17.sql - ...Q2

```

INSERT INTO CONTRAT ([contrat_ID],[date_de_debut],[date_de_fin],[nombre_de_chanson],[artiste_ID])
VALUES(122,'20200512','20210603',5,1)
select * from CONTRAT

```

122 %

Results Messages

	contrat_ID	date_de_debut	date_de_fin	nombre_de_chanson	artiste_ID
1	122	2020-05-12	2021-06-03	5	1
2	351	2016-01-21	2018-04-13	5	2
3	432	2015-05-21	2021-05-13	11	5
4	832	2020-05-09	2022-04-02	3	3

2)

```

GO
CREATE TRIGGER inserted_artiste_add_membres
ON ARTISTE
AFTER INSERT
AS
BEGIN
    DECLARE @groupe_nom nvarchar(50)
    SELECT @groupe_nom = A.groupe_nom FROM ARTISTE AS A

    IF @groupe_nom IS NULL
    BEGIN
        PRINT('Vous devez insérer artiste à la table
membres')
    END
    ELSE
    BEGIN
        PRINT('Vous devez insérer les membres de groupe à la
table membres')
    END
END
GO

```

SQLQuery10.sql - ...Q2BB\Monster (63))\* X SQLQuery8.sql - D...Q2BB\Monster (69))\* SQLQuery5.sql - D...Q2BB\

```

INSERT INTO ARTISTE(artiste_ID, artiste_prenom, artiste_nom,groupe_nom)
VALUES(17, NULL, NULL, 'ADAMLAR')

```

122 %

Messages

Vous devez insérer les membres de groupe à la table membres

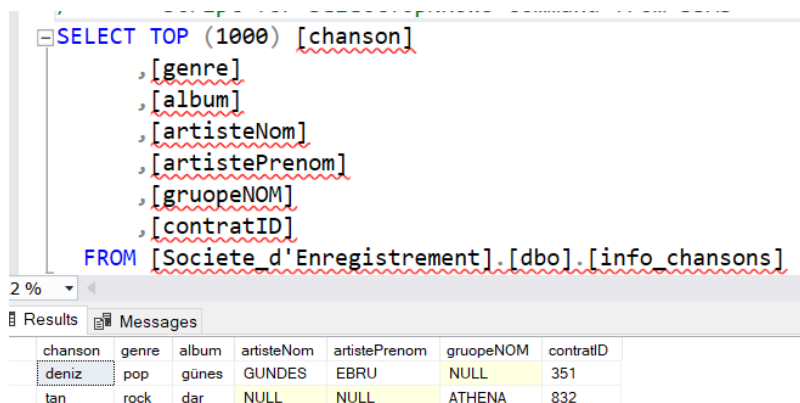
(1 row affected)

Completion time: 2022-01-05T15:30:37.4120760+03:00

## VUE :

J'ai utilisé 5 tables pour créer la vue : genre, chanson, album, artiste et contrat. En utilisant ces 5 tables, j'ai créé une vue qui donne le nom, le genre, l'artiste et l'album des chansons et le contratID pour l'artiste.

```
CREATE VIEW info_chansons AS
SELECT
C.chanson_nom AS chanson,
G.genre_nom AS genre,
A.album_nom AS album,
AR.artiste_nom AS artisteNom,
AR.artiste_prenom AS artistePrenom,
AR.groupe_nom AS groupeNOM,
CO.contrat_ID AS contratID
FROM
genre G INNER JOIN chanson C
ON G.genre_ID = C.genre_ID INNER JOIN album A
ON C.album_ID = A.album_ID INNER JOIN ARTISTE AR
ON A.artiste_ID = AR.artiste_ID INNER JOIN CONTRAT CO
ON AR.artiste_ID = CO.artiste_ID
```



The screenshot shows a SQL query window with the following text:

```
SELECT TOP (1000) [chanson]
      , [genre]
      , [album]
      , [artisteNom]
      , [artistePrenom]
      , [groupeNOM]
      , [contratID]
FROM [Societe_d'Enregistrement].[dbo].[info_chansons]
```

Below the query window, the 'Results' tab is active, displaying a table with 7 columns and 3 rows of data:

chanson	genre	album	artisteNom	artistePrenom	groupeNOM	contratID
deniz	pop	günes	GUNDES	EBRU	NULL	351
tan	rock	dar	NULL	NULL	ATHENA	832

## INDEX :

J'ai créé un index pour la colonne "chanson\_nom" de la table "chanson".

```
CREATE INDEX ix_chanson_nom
ON chanson(chanson_nom);
```