

# Database Final Project Report

## 1. Project Summary

This project is about a music database that stores data of modern pop artists, songs, albums and all the related information.

### **It contains 24 Tables:**

- 1) Album
- 2) Artist
- 3) Artist\_Song
- 4) Song
- 5) Award
- 6) Ceremony
- 7) Certification
- 8) ChartRanking
- 9) RankingChange
- 10) Country
- 11) Genre
- 12) Label
- 13) Studio
- 14) Location
- 15) LocationChange
- 16) Musician
- 17) Instrument
- 18) Musician\_Instrument
- 19) Producer
- 20) Producer\_Song
- 21) Review
- 22) ReviewSource
- 23) Sampling
- 24) Song\_Musician.

### **8 Views:**

- 1) Album\_Info
- 2) Artist\_Info
- 3) Award\_Info
- 4) Award\_Winning\_Num
- 5) Certification\_Info

- 6) GlobalSale
- 7) Song\_Info
- 8) Studio\_Info.

Six of them, like Album\_Info ,are done using the JOIN statement to gain more readable information of the tables. 2 of them are used to calculate the sales and the award-winning data of albums.

3 Stored Procedures:

1. Best\_Selling\_Album\_Of\_Artist\_WorldWide (nameOfArtist varchar(20))
2. Best\_Selling\_Album\_Of\_Artist\_Country (nameOfArtist varchar(20), nameOfCountry varchar(20))
3. Most\_Award\_Winning\_Album\_Of\_Artist (nameOfArtist varchar(20))

## 2 Triggers:

- 1) RankingUpdate:

Insert a new record to the table RankingChange every time an album gets to a higher ranking.

- 2) StudioUpdate:

Insert a new record to the table LocationChange every time a studio changes its location.

## 2.StoredProcedure, Trigger, and View codes

### 1) Stored Procedures

```
DELIMITER //
CREATE PROCEDURE Best_Selling_Album_Of_Artist_WorldWide
(nameOfArtist varchar(20))
BEGIN
    Select * from globalsale
    Where sales = (select max(sales) from globalsale where artistName = nameOfArtist);
END//
DELIMITER ;
-----1
```

```
DELIMITER //
CREATE PROCEDURE Best_Selling_Album_Of_Artist_Country
(nameOfArtist varchar(20), nameOfCountry varchar(20))
BEGIN
    select a.*, sub.country,sub.certification,sub.sales from Album_info A
    join
    (select * from certification_info c
```

```

        where sales = (select max(sales) from certification_info c
        where c.country = nameOfCountry and c.`Artist Name` = nameOfArtist)) AS SUB
        ON SUB.ALBUM = A.AlbumTitle;
END //
DELIMITER ;
-----2

DELIMITER //
CREATE PROCEDURE Most_Award_Winning_Album_Of_Artist
(nameOfArtist varchar(20))
BEGIN
    select * FROM AWARD_WINNING_NUM
    WHERE
    `Number of Awards` = (SELECT max(`Number of Awards`) from AWARD_WINNING_NUM
WHERE ArtistName =nameOfArtist);
END //
DELIMITER ;
-----3

```

## 2) Triggers

```

delimiter //
CREATE TRIGGER RankingUpdate
BEFORE UPDATE ON ChartRanking
FOR EACH ROW
BEGIN
    set @newPosition=NEW.PeakPosition;
    set @oldPosition=OLD.PeakPosition;
    if (@newPosition < @oldPosition) THEN
        INSERT INTO RankingChange(FK_AlbumID,PeakPosition,ChangeDate)
        VALUES(NEW.FK_AlbumID,@newPosition,sysdate());
    END IF;
END //
delimiter ;
-----1

```

```

delimiter //
CREATE TRIGGER StudioUpdate
BEFORE UPDATE ON STUDIO
FOR EACH ROW
BEGIN
    set @newLocation=NEW.FK_Location;
    set @oldLocation=OLD.FK_Location;

```

```

        if (@oldLocation != @newLocation) THEN
            INSERT
            LocationChange(FK_Studio,FK_OldLocation,FK_NewLocation,ChangeDate)
            VALUES(NEW.StudioID,@oldLocation,@newLocation,sysdate());

            Update LOCATION
            SET isCurrent = 'N'
            WHERE LOCATION.LocationID=@oldLocation;

            Update LOCATION
            SET isCurrent = 'Y'
            WHERE LOCATION.LocationID=@newLocation;
        END IF;
    END //
delimiter ;
-----2

```

### 3) Views

```

CREATE VIEW album_info AS
select a.ArtistName AS `ArtistName`,
al.AlbumTitle AS `AlbumTitle`,
al.ReleaseDate AS `ReleaseDate`,
g.GenreName AS `GenreName`
from album `al`
join genre `g`
on al.FK_Genre = g.GenreID
join artist `a`
on al.FK_Artist = a.ArtistID;
-----1

```

```

CREATE VIEW Artist_INFO AS
    SELECT a.ArtistName as 'Artist Name',
           TIMESTAMPDIFF(YEAR, a.Birthday, CURDATE()) AS Age,
           C.CountryName as 'Country',
           L.LABELNAME as 'Label'
    from Artist A
    JOIN Country C ON C.CountryID=A.FK_Country
    JOIN LABEL L ON L.LABELID=A.FK_LABEL;
-----2

```

```

CREATE VIEW AWARD_INFO AS
    SELECT a.ArtistName as 'Artist Name',

```

```

        C.CeremonyName as 'Ceremony Name',
        AL.AlbumTitle as 'Album',
        aw.awardedYear as 'Awarded Year',
        aw.Category, aw.result
    from award aw
    JOIN artist a ON a.ArtistID = aw.FK_ArtistID
    JOIN Ceremony C ON C.CeremonyID = AW.FK_CeremonyID
    JOIN ALBUM AL ON AL.AlbumID = AW.FK_Album
    order by AL.AlbumTitle;

```

-----3

```

CREATE VIEW CERTIFICATION_INFO AS
    SELECT a.ArtistName as 'Artist Name',
           C.CountryName as 'Country',
           AL.AlbumTitle as 'Album',
           CT.certification as 'Certification',
           CT.CertifiedSales as 'Sales'
    from Certification CT
    JOIN Country C ON C.CountryID = CT.FK_Country
    JOIN ALBUM AL ON AL.AlbumID = CT.FK_ALBUM
    JOIN artist a ON A.ArtistID = AL.FK_Artist
    order by AL.AlbumTitle, Country;

```

-----4

```

CREATE VIEW Song_INFO AS
    SELECT A.ArtistName as 'Artist Name',
           S.SongName as 'Song Name',
           S.Length as 'Length',
           AL.AlbumTitle as 'Album',
           ST.StudioName as 'Studio'
    from Artist A
    JOIN artist_song on artist_song.FK_ARTIST = A.ArtistID
    JOIN Song S ON S.SongID = artist_song.FK_SONG
    LEFT JOIN Album AL ON AL.AlbumID = S.FK_Album
    JOIN studio ST ON ST.StudioID = S.FK_Studio
    order by A.ArtistName, AL.AlbumTitle;

```

-----5

```

CREATE VIEW Song_INFO AS
    SELECT A.ArtistName as 'Artist Name',
           S.SongName as 'Song Name',
           S.Length as 'Length',
           AL.AlbumTitle as 'Album',
           ST.StudioName as 'Studio'

```

```

        from Artist A
    JOIN artist_song on artist_song.FK_ARTIST = A.ArtistID
    JOIN Song S ON S.SongID = artist_song.FK_SONG
    LEFT JOIN Album AL ON AL.AlbumID = S.FK_Album
    JOIN studio ST ON ST.StudioID = S.FK_Studio
    order by A.ArtistName, AL.AlbumTitle;
-----6

```

```

CREATE VIEW Studio_INFO AS
    SELECT S.StudioName as 'Studio Name',
        L.Street1,
        L.Street2,
        L.City,
        L.PROVINCE,
        L.POSTCODE,
        C.COUNTRYNAME
    from STUDIO S
    JOIN LOCATION L ON L.LOCATIONID = S.FK_LOCATION
    JOIN COUNTRY C ON C.COUNTRYID = L.FK_COUNTRY*/
-----7

```

```

Create VIEW GlobalSale AS
SELECT a.*, sum(c.Sales) 'Sales' from certification_info c
join album_info a on a.AlbumTitle = c.album
group by Album;

```

```

CREATE VIEW AWARD_WINNING_NUM AS
SELECT AlbuM_INFO.*, SUB1.`Number of Awards`
    FROM
        (SELECT `Artist Name`, Album, COUNT(*) as 'Number of Awards'
        FROM award_info
        WHERE RESULT = 'Won'
        group by album) SUB1
    JOIN AlbuM_INFO
ON album_info.ALBUMTITLE = SUB1.ALBUM;
-----8

```

### 3.ER Diagram

