2/3/2016



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This assignment conducted by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Leader 1) Fikri saoudi 1141327209

2)Ahmad Hamdan 1131121467

To linda

MULTIMEDIA UNIVERSITY

**List of content**

[Business Rules 2](#_Toc442204258)

[ERD 3](#_Toc442204259)

[Data dictionary 4](#_Toc442204260)

[Data Definition Command (**DDL**) 5](#_Toc442204261)

[1.1 Create tables and Data insertion : 5](#_Toc442204262)

[2 Data Manipulation Command (DML) 16](#_Toc442204263)

[2.1 aggregate functions: 16](#_Toc442204264)

[2.2 group by and aggregate functions (count,sum) 17](#_Toc442204265)

[2.3 Select and order by and join 18](#_Toc442204266)

[2.4 Select and LIKE 19](#_Toc442204267)

[2.5 sub-query and join 19](#_Toc442204268)

[2.6 Nested query or sub query 20](#_Toc442204269)

[2.7 VIEW 21](#_Toc442204270)

[2.8 TRIGGER: 22](#_Toc442204271)

[2.9 Stored procedure : 24](#_Toc442204272)

[2.10 Displaying "top n records” 26](#_Toc442204273)

[2.11 Group by with HAVING 27](#_Toc442204274)

[2.12 auto increment by 1 for newly inserted row 28](#_Toc442204275)

# Business Rules

# 

each customer may purchase many songs

each song can be purchased by many customer

each customer may purchase many albums

each album can be purchased by many customer

each album contains many songs

each song is contained by an album

artist release many songs

songs can be released by one or more artists

# ERD

# Data dictionary

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table Name | Attribute Name | Contents/Description | Data Type | Format | Range | Nullable | PK / FK | FK Referenced Table |
| ARTIST | artist\_ID  artist\_name  artist\_gender  artist\_country  singing\_kind | artist ID  the artist name  artist sex  artist country  artist style of signing | int  Varchar (30)  Char(1)  Varchar (30)  Varchar (30) | #####  Xxxxxxx  X  Xxxxxx  Xxxxxx | 1-100 | No  No  Yes  Yes  Yes | PK |  |
| RELEASE | song\_ID  artist\_ID  date | songID  artistID  date | int  Varchar (30) | ###  Xxxxxxx |  | No  Yes | PK /FK  PK /FK | SONG  ARTIST |
| SONG | Song\_ID  Album\_ID  Song\_name  Song\_category  Duration  Song\_price  Song\_language | SongID  AlbumiD  Song name  Song type  Duration  Song price  language | int  int  Varchar (30)  Varchar (30)  Decimal(4,2)  Decimal(7,2)  Varchar (30) | ###  ###  Xxxxxx  Xxxxxx  Xxxxxx |  | No  Yes  No  Yes  Yes  Yes  yes | PK  FK | ALBUM |
| ALBUM | Albim\_ID  Name  Album\_kind |  | int  Varchar (30)  Varchar (30) | #####  Xxxxxx  ##  Xxxxxx  Xxxxxx  Xxxxxx  YYYY-MM-DD |  | No  Yes  Yes  Yes  Yes  Yes  Yes | PK |  |
| PURCHASES | Customer\_ID  Song\_ID  Date |  | int  int  date | ####  Xxxxxx  Yyyy-mm-dd |  | No  Yes  Yes | PK/FK  PK/FK | COSTUMER  SONG |
| COSTUMER | Costumer\_ID  Costumer\_name  Costumer\_birthdate  Costumer\_gender  Costumer\_phone |  | Int  Varchar (30)  Date  Chat(1)  Varchar (30) | #####  Xxxxxx  Yyyy-mm-dd  X  Xxxxxx | 0-100 | No  No  Yes  Yes  Yes | PK |  |
| PURCHASEALB | Costomer\_ID  Album\_ID  date |  | int  int  date | #####  #####  Yyyy-mm-dd | 1-5 | No  No  Yes | PK/FK  PK/FK | COSTUMER  ALBUM |

# Data Definition Command (**DDL**)

## Create tables and Data insertion :

***ARTIST TABLE :***

create table ARTIST(

ARTIST\_ID int not null,

ARTIST\_NAME varchar(30),

ARTIST\_GENDER char(1) check(ARTIST\_GENDER in('F','M')),

ARTIST\_COUNTRY varchar(30),

SINGING\_KIND varchar(30),

primary key(ARTIST\_ID)

);

***Insert data in table:***

insert into ARTIST values(111,'EMINEM','M','USA','RAP');

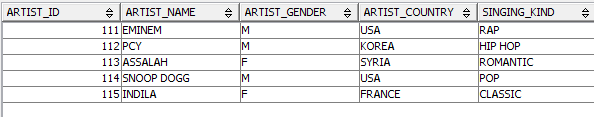
insert into ARTIST values(112,'PCY','M','KOREA','HIP HOP');

insert into ARTIST values(113,'ASSALAH','F','SYRIA','ROMANTIC');

insert into ARTIST values(114,'SNOOP DOGG','M','USA','POP');

insert into ARTIST values(115,'INDILA','F','FRANCE','CLASSIC');

select \* from ARTIST;



***ALBUM table:***

create table ALBUM(

ALBUM\_ID int not null,

ALBUM\_NAME varchar(30),

ALBUM\_KIND varchar(30),

primary key(ALBUM\_ID)

);

***Insert data in table:***

insert into ALBUM values(1991,'RECOVERY','RAP');

insert into ALBUM values(1992,'PCY-DA','HIP HOP');

insert into ALBUM values (1993,'HAYATI','ROMANTIC');

insert into ALBUM values(1994,'BUSH','RAP');

insert into ALBUM values(1995,'MINI WORLD','CLASSIC');

SELECT \* FROM ALBUM;



***SONG Table***

create table SONG(

SONG\_ID int not null,

ALBUM\_ID int,

SONG\_NAME varchar(30),

SONG\_CATEGORY varchar(30),

DURATION decimal(4,2),

SONG\_PRICE decimal(7,2),

SONG\_LANGUAGE varchar(30),

primary key(SONG\_ID),

foreign key(ALBUM\_ID)references ALBUM

);

***Insert data:***

insert into SONG values(1,1991,'NOT AFRAID','CLASSEC',3.50,30.00,'ENGLISH');

insert into SONG values(2,1991,'NO LOVE','JAZZ',4.20,12.50,'ENGLISH');

insert into SONG values(3,1991,'SOO BAD','CLASSEC',5.20,11.30,'ENGLISH');

insert into SONG values(4,1991,'DONT WORRY','HIP HOP',3.15,10.10,'ENGLISH');

insert into SONG values(5,1992,'THE DAY WILL COME','RAP',3.33,20.01,'KOREAN');

insert into SONG values(6,1992,'DREAM','CLASSEC',3.17,5.90,'ENGLISH @ KOREAN');

insert into SONG values(7,1992,'DADDY','HIP HOP',4.02,4.04,'KOREAN');

insert into SONG values(8,1992,'LOVE YOU','HIP HOP',3.16,6.80,'KOREAN');

insert into SONG values(9,1993,'AKTAR','ROMANTIC',7.75,4.11,'ARABIC');

insert into SONG values(10,1993,'HAYATI','SLOW SONG',5.02,4.50,'MIX ARABIC @ FRANCE');

insert into SONG values(11,1993,'WEHNA SAWA','DANCING SONG',4.15,8.60,'ARABIC');

insert into SONG values(12,1993,'ENTA OMRY','ARAB CLASSIC',7.10,6.81,'ARABIC');

insert into SONG values(13,1994,'THIS CITY','RAP',2.50,2.30,'ENGLISH');

insert into SONG values(14,1994,'AWAKE','CLASSEC',4.45,5.40,'ENGLISH @ ITALY');

insert into SONG values(15,1994,'RUN AWAY','PIT BOX',2.40,1.70,'ENGLISH');

insert into SONG values(16,1994,'COME BACK','PIT BOX',2.40,1.75,'ENGLISH');

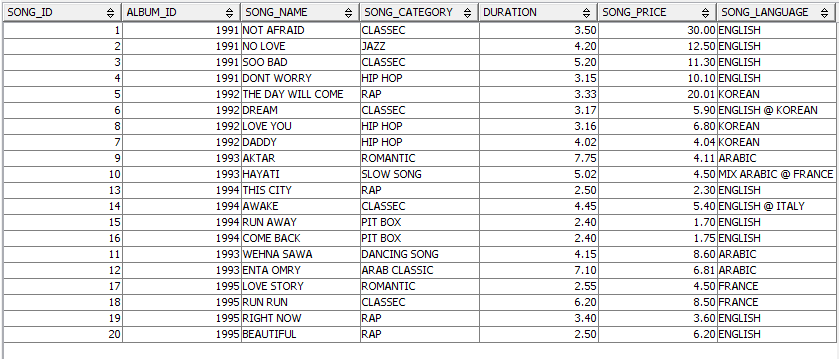
insert into SONG values(17,1995,'LOVE STORY','ROMANTIC',2.55,4.50,'FRANCE');

insert into SONG values(18,1995,'RUN RUN','CLASSEC',6.20,8.50,'FRANCE');

insert into SONG values(19,1995,'RIGHT NOW ','RAP',3.40,3.60,'ENGLISH');

insert into SONG values(20,1995,'BEAUTIFUL','RAP',2.50,6.20,'ENGLISH');

SELECT \* FROM SONG;



***CUSTOMER table:***

CREATE TABLE CUSTOMER(

CUSTOMER\_ID int not null,

CUSTOMER\_NAME varchar(30),

CUSTOMER\_BIRTHDATE date,

CUSTOMER\_GENDER char(1) check(CUSTOMER\_GENDER in ('M','F')),

CUSTOMER\_PHONE varchar(10),

PRIMARY KEY(CUSTOMER\_ID)

);

***Insert Data:***

INSERT INTO CUSTOMER VALUES(111,'FIKRI','1990-05-30','M','0112342557');

INSERT INTO CUSTOMER VALUES(112,'AMINA','1995-11-05','F','0116632565');

INSERT INTO CUSTOMER VALUES(113,'AMIN','1968-07-09','M','0132342564');

INSERT INTO CUSTOMER VALUES(114,'RIMA','1957-08-27','F','0142342565');

INSERT INTO CUSTOMER VALUES(115,'RAMI','1999-02-17','M','0162342565');

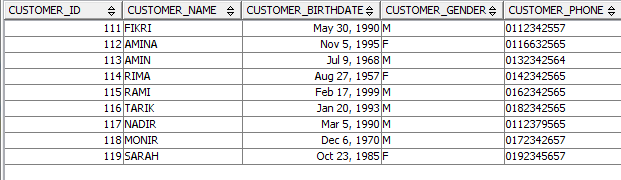
INSERT INTO CUSTOMER VALUES(116,'TARIK','1993-01-20','M','0182342565');

INSERT INTO CUSTOMER VALUES(117,'NADIR','1990-03-05','M','0112379565');

INSERT INTO CUSTOMER VALUES(118,'MONIR','1970-12-06','M','0172342657');

INSERT INTO CUSTOMER VALUES(119,'SARAH','1985-10-23','F','0192345657');

SELECT \* FROM CUSTOMER;



***PURCHASEALB table:***

CREATE TABLE PURCHASEALB(

CUSTOMER\_ID int not null,

ALBUM\_ID int not null,

DATE date,

primary key(CUSTOMER\_ID,ALBUM\_ID),

foreign key (CUSTOMER\_ID) references CUSTOMER,

foreign key(ALBUM\_ID) references ALBUM

);

***Insert Data:***

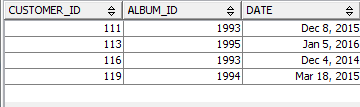
INSERT INTO PURCHASEALB VALUES(111,1993,'2015-12-08');

INSERT INTO PURCHASEALB VALUES(113,1995,'2016-01-05');

INSERT INTO PURCHASEALB VALUES(116,1993,'2014-12-04');

INSERT INTO PURCHASEALB VALUES(119,1994,'2015-03-18');

SELECT \* FROM PURCHASEALB;



***PURCHASES table:***

CREATE TABLE PURCHASES(

CUSTOMER\_ID int not null,

SONG\_ID int not null,

DATE date,

primary key(CUSTOMER\_ID,SONG\_ID),

foreign key (CUSTOMER\_ID) references CUSTOMER,

foreign key(SONG\_ID) references SONG

);

***Insert data:***

INSERT INTO PURCHASES VALUES(112,1,'2013-11-08');

INSERT INTO PURCHASES VALUES(112,15,'2015-12-08');

INSERT INTO PURCHASES VALUES(114,5,'2016-01-07');

INSERT INTO PURCHASES VALUES(115,19,'2014-12-09');

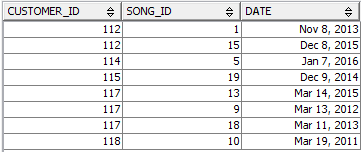
INSERT INTO PURCHASES VALUES(117,13,'2015-03-14');

INSERT INTO PURCHASES VALUES(117,9,'2012-03-13');

INSERT INTO PURCHASES VALUES(117,18,'2013-03-11');

INSERT INTO PURCHASES VALUES(118,10,'2011-03-19');

SELECT \* FROM PURCHASES;



***RELEASE table:***

CREATE TABLE RELEASE(

SONG\_ID int not null,

ARTIST\_ID int not null,

DATE date,

primary key(SONG\_ID,ARTIST\_ID),

foreign key (ARTIST\_ID) references ARTIST,

foreign key(SONG\_ID) references SONG

);

***Insert Data:***

INSERT INTO RELEASE VALUES(19,111,'2008-03-25');

INSERT INTO RELEASE VALUES(20,111,'2007-08-20');

INSERT INTO RELEASE VALUES(13,111,'2010-05-18');

INSERT INTO RELEASE VALUES(5,111,'2009-09-11');

INSERT INTO RELEASE VALUES(8,112,'2008-02-22');

INSERT INTO RELEASE VALUES(3,112,'2008-12-21');

INSERT INTO RELEASE VALUES(7,112,'2006-09-15');

INSERT INTO RELEASE VALUES(4,112,'2008-09-13');

INSERT INTO RELEASE VALUES(9,113,'2005-04-29');

INSERT INTO RELEASE VALUES(12,113,'2004-11-21');

INSERT INTO RELEASE VALUES(17,113,'2002-07-19');

INSERT INTO RELEASE VALUES(10,113,'2005-03-15');

INSERT INTO RELEASE VALUES(2,114,'2008-02-09');

INSERT INTO RELEASE VALUES(15,114,'2006-10-11');

INSERT INTO RELEASE VALUES(16,114,'2009-09-17');

INSERT INTO RELEASE VALUES(11,114,'2010-06-15');

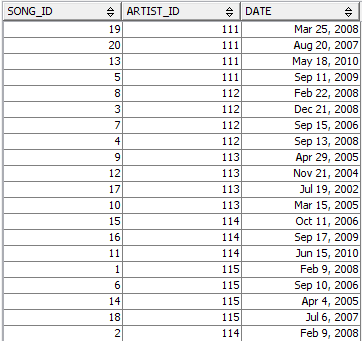
INSERT INTO RELEASE VALUES(1,115,'2008-02-09');

INSERT INTO RELEASE VALUES(6,115,'2006-09-10');

INSERT INTO RELEASE VALUES(14,115,'2005-04-04');

INSERT INTO RELEASE VALUES(18,115,'2007-07-06');

SELECT \* FROM RELEASE;



# Data Manipulation Command (DML)

## aggregate functions:

Select ARTIST\_NAME,SONG\_NAME,SONG\_PRICE,SONG\_CATEGORY,DURATION

FROM SONG,ARTIST,RELEASE

WHERE ARTIST.ARTIST\_ID = RELEASE.ARTIST\_ID

AND RELEASE.SONG\_ID = SONG.SONG\_ID

AND SONG.ALBUM\_ID<=1995

AND SONG\_PRICE=(SELECT max(SONG\_PRICE)FROM SONG);

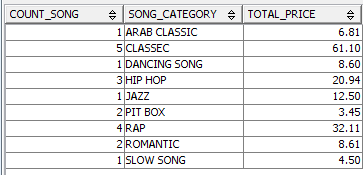


## group by and aggregate functions (count,sum)

SELECT COUNT(SONG\_NAME)AS COUNT\_SONG, SONG\_CATEGORY,SUM(SONG\_PRICE) AS TOTAL\_PRICE

FROM SONG

GROUP BY SONG\_CATEGORY;



## Select and order by and join

Select CUSTOMER\_NAME,SONG\_NAME,SONG\_PRICE,SONG\_CATEGORY,ARTIST\_NAME,PURCHASES.DATE,CUSTOMER\_PHONE

FROM SONG,CUSTOMER,PURCHASES,ARTIST,RELEASE

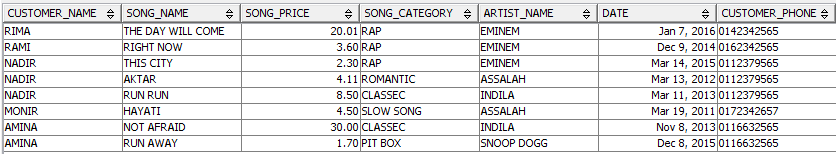
WHERE PURCHASES.SONG\_ID = SONG.SONG\_ID

AND PURCHASES.CUSTOMER\_ID = CUSTOMER.CUSTOMER\_ID

AND ARTIST.ARTIST\_ID = RELEASE.ARTIST\_ID

AND RELEASE.SONG\_ID = SONG.SONG\_ID

ORDER BY CUSTOMER\_NAME DESC;



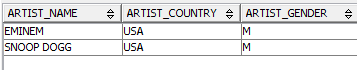
## Select and LIKE

SELECT ARTIST\_NAME ,ARTIST\_COUNTRY,ARTIST\_GENDER

FROM ARTIST

WHERE ARTIST\_GENDER='M'

AND ARTIST\_COUNTRY LIKE 'USA';



## sub-query and join

SELECT CUSTOMER\_NAME,ALBUM\_NAME,CUSTOMER\_PHONE ,DATE

FROM CUSTOMER,PURCHASEALB,ALBUM

WHERE PURCHASEALB.ALBUM\_ID = ALBUM.ALBUM\_ID

AND PURCHASEALB.CUSTOMER\_ID = CUSTOMER.CUSTOMER\_ID

AND DATE=(SELECT MIN(DATE)FROM PURCHASEALB);



## Nested query or sub query

UPDATE SONG

SET SONG\_PRICE = (SELECT AVG(SONG\_PRICE) FROM SONG)

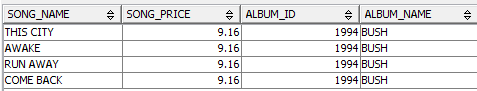
WHERE ALBUM\_ID IN (SELECT ALBUM\_ID FROM ALBUM WHERE ALBUM\_NAME = 'BUSH');

SELECT SONG\_NAME,SONG\_PRICE,ALBUM.ALBUM\_ID,ALBUM\_NAME

FROM SONG,ALBUM

WHERE ALBUM.ALBUM\_ID = SONG.ALBUM\_ID

AND ALBUM\_NAME = 'BUSH';



## VIEW

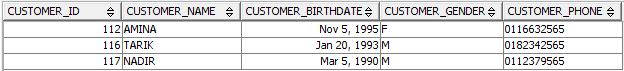
CREATE VIEW BUYERS AS

SELECT \* FROM CUSTOMER

WHERE CUSTOMER\_BIRTHDATE BETWEEN '1990-01-01' AND '1999-01-01'

AND CUSTOMER\_PHONE LIKE'%65';

SELECT \* FROM BUYERS;



## TRIGGER:

CREATE TRIGGER AUTO\_PRICE

AFTER INSERT ON SONG

FOR EACH ROW MODE DB2SQL

UPDATE SONG

SET SONG\_PRICE = 5

WHERE SONG\_CATEGORY = 'RAP'

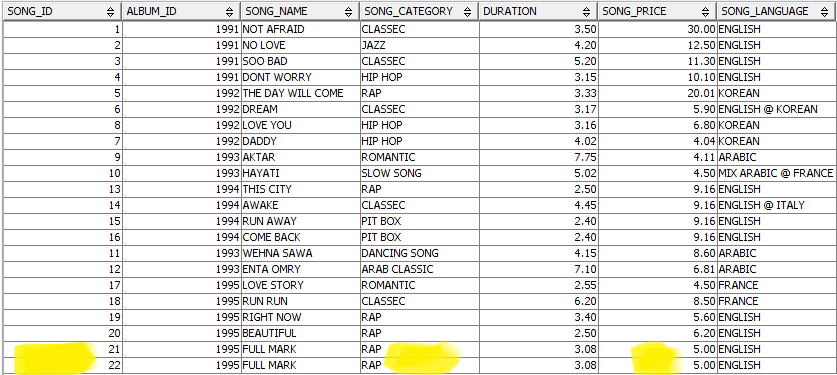
AND SONG\_PRICE IS NULL;

***Insert in table***

INSERT INTO SONG VALUES (21,1995,'FULL MARK','RAP',3.08,NULL,'ENGLISH');

INSERT INTO SONG VALUES (22,1995,'FULL MARK','RAP',3.08,NULL,'ENGLISH');

SELECT \* FROM SONG;



## Stored procedure :

CREATE PROCEDURE DISC(IN DISCOUNT INT)

BEGIN

UPDATE SONG

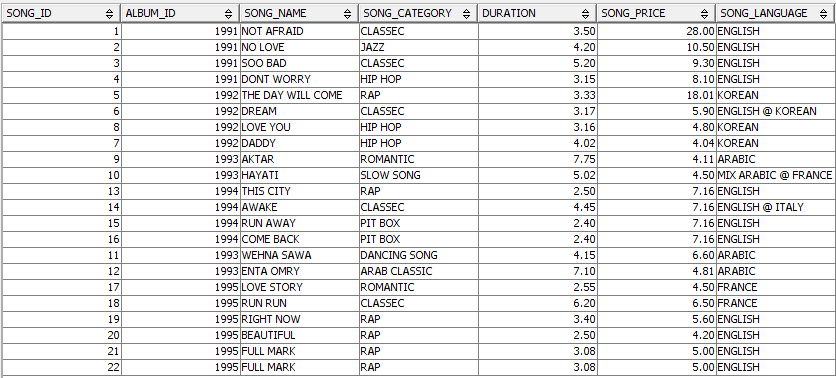
SET SONG\_PRICE = SONG\_PRICE - DISCOUNT

WHERE SONG\_PRICE > 6;

END@

CALL DISC(2);

SELECT \* FROM SONG;



## Displaying "top n records”

SELECT SONG\_NAME,SONG\_PRICE

FROM SONG

ORDER BY SONG\_PRICE DESC

LIMIT 5;



SELECT DATE,ARTIST\_NAME,SONG\_NAME,SONG\_CATEGORY

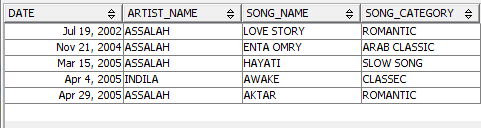
FROM RELEASE,ARTIST,SONG

WHERE ARTIST.ARTIST\_ID = RELEASE.ARTIST\_ID

AND SONG.SONG\_ID = RELEASE.SONG\_ID

ORDER BY DATE ASC

LIMIT 5;



## Group by with HAVING

SELECT ARTIST\_NAME,COUNT(SONG\_LANGUAGE) AS ENGLISH\_TYPE

FROM SONG,ARTIST,RELEASE

WHERE

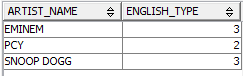
ARTIST.ARTIST\_ID=RELEASE.ARTIST\_ID

AND RELEASE.SONG\_ID=SONG.SONG\_ID

AND SONG.SONG\_LANGUAGE='ENGLISH'

GROUP BY ARTIST\_NAME

HAVING COUNT(SONG\_LANGUAGE)>1;



## auto increment by 1 for newly inserted row

**Note since table SONG has a lot of dependencies**

**We created another table with same attribute**

**To show the new trigger .**

***SONG2 TABLE***

create table SONG2(

SONG\_ID int not null,

ALBUM\_ID int,

SONG\_NAME varchar(30),

SONG\_CATEGORY varchar(30),

DURATION decimal(4,2),

SONG\_PRICE decimal(7,2),

SONG\_LANGUAGE varchar(30),

primary key(SONG\_ID),

foreign key(ALBUM\_ID)references ALBUM

);

INSERT IN TABLE ( THE SAME INPUT OF TABLE SONG )

***First way***

INSERT INTO SONG2

(SELECT \* FROM SONG2)

***Second way***

insert into SONG2 values(1,1991,'NOT AFRAID','CLASSEC',3.50,30.00,'ENGLISH');

insert into SONG2 values(2,1991,'NO LOVE','JAZZ',4.20,12.50,'ENGLISH');

insert into SONG2 values(3,1991,'SOO BAD','CLASSEC',5.20,11.30,'ENGLISH');

insert into SONG2 values(4,1991,'DONT WORRY','HIP HOP',3.15,10.10,'ENGLISH');

insert into SONG2 values(5,1992,'THE DAY WILL COME','RAP',3.33,20.01,'KOREAN');

insert into SONG2 values(6,1992,'DREAM','CLASSEC',3.17,5.90,'ENGLISH @ KOREAN');

insert into SONG2 values(7,1992,'DADDY','HIP HOP',4.02,4.04,'KOREAN');

insert into SONG2 values(8,1992,'LOVE YOU','HIP HOP',3.16,6.80,'KOREAN');

insert into SONG2 values(9,1993,'AKTAR','ROMANTIC',7.75,4.11,'ARABIC');

insert into SONG2 values(10,1993,'HAYATI','SLOW SONG',5.02,4.50,'MIX ARABIC @ FRANCE');

insert into SONG2 values(11,1993,'WEHNA SAWA','DANCING SONG',4.15,8.60,'ARABIC');

insert into SONG2 values(12,1993,'ENTA OMRY','ARAB CLASSIC',7.10,6.81,'ARABIC');

insert into SONG2 values(13,1994,'THIS CITY','RAP',2.50,2.30,'ENGLISH');

insert into SONG2 values(14,1994,'AWAKE','CLASSEC',4.45,5.40,'ENGLISH @ ITALY');

insert into SONG2 values(15,1994,'RUN AWAY','PIT BOX',2.40,1.70,'ENGLISH');

insert into SONG2 values(16,1994,'COME BACK','PIT BOX',2.40,1.75,'ENGLISH');

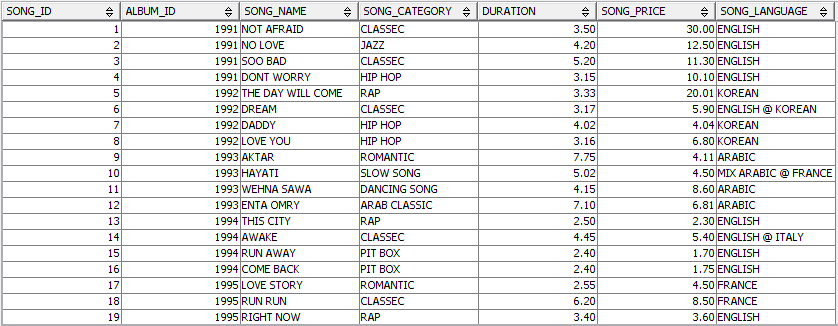
insert into SONG2 values(17,1995,'LOVE STORY','ROMANTIC',2.55,4.50,'FRANCE');

insert into SONG2values(18,1995,'RUN RUN','CLASSEC',6.20,8.50,'FRANCE');

insert into SONG2 values(19,1995,'RIGHT NOW ','RAP',3.40,3.60,'ENGLISH');

insert into SONG2 values(20,1995,'BEAUTIFUL','RAP',2.50,6.20,'ENGLISH');

select \* from song2



**Create trigger :**

CREATE TRIGGER NEWADDP

AFTER INSERT ON SONG

REFERANCING AS N

FOR EACH ROW MODE DB2SQL

UPDATE SONG

SET SONG­\_PRICE=N.SONG\_PRICE +1;

***Insert in table song2***

insert into SONG2 values(24,1994,'COME BACK','PIT BOX',2.40,39.00,'ENGLISH';

