**Tables SQL:**

1. **Artist**

CREATE TABLE ARTIST

( ID NUMBER NOT NULL,

NAME VARCHAR2 (60 BYTE) NOT NULL,

CONSTRAINT ARTIST\_PK PRIMARY KEY (ID)

**);**

1. **SONGSCATEGORY**

CREATE TABLE SONGSCATEGORY

( CATEGORYID NUMBER NOT NULL,

CATEGORYNAME VARCHAR2 (60 BYTE) NOT NULL,

CONSTRAINT SONGSCATEGORY\_PK PRIMARY KEY

(CATEGORYID)

);

1. **Songs**

CREATE TABLE SONGS

( SONGID NUMBER NOT NULL ENABLE,

SONGNAME VARCHAR2(50 BYTE) NOT NULL ENABLE,

CATEGORYID NUMBER NOT NULL ENABLE,

ARTISTID NUMBER,

CONSTRAINT SONGS\_PK PRIMARY KEY (SONGID),

CONSTRAINT SONGCATEGORY FOREIGN KEY (CATEGORYID)

REFERENCES SONGSCATEGORY (CATEGORYID) ENABLE,

CONSTRAINT IDARTIST FOREIGN KEY (ARTISTID)

REFERENCES ARTIST (ID) ENABLE

);

1. **SUBSCRIPTIONPLAN:**

CREATE TABLE SUBSCRIPTIONPLAN

( SUBNAME VARCHAR2 (60 BYTE) NOT NULL,

PRICE NUMBER NOT NULL,

DURATION NUMBER NOT NULL,

CONSTRAINT SUBSCRIPTIONPLAN\_PK PRIMARY KEY (SUBNAME)

);

1. **User Info:**

CREATE TABLE USERINFO

( USERID NUMBER NOT NULL,

USERNAME VARCHAR2 (30 BYTE) NOT NULL,

STARTDATE DATE NOT NULL,

SUBSCRIPTIONNAME VARCHAR2 (60 BYTE) DEFAULT 'regular' NOT NULL,

CONSTRAINT USERINFO\_PK PRIMARY KEY (USERID),

CONSTRAINT SUBSCRIPTIONNAME FOREIGN KEY (SUBSCRIPTIONNAME) REFERENCES SUBSCRIPTIONPLAN (SUBNAME)

);

1. **Favorite list:**

CREATE TABLE FAVOURITLIST

( SONGID NUMBER NOT NULL,

USERID NUMBER NOT NULL,

CONSTRAINT FAVOURITLIST\_PK PRIMARY KEY (SONGID, USERID),

CONSTRAINT USERID FOREIGN KEY ("USERID")

REFERENCES USERINFO (USERID),

CONSTRAINT FAVOURITLIST\_SONGS\_FK1 FOREIGN KEY (SONGID) REFERENCES SONGS (SONGID)

);

1. **Artist list**

CREATE TABLE ARTISTLIST

( ARTISTID NUMBER NOT NULL,

USERID NUMBER NOT NULL,

CONSTRAINT ARTISTLIST\_PK PRIMARY KEY (ARTISTID, USERID),

CONSTRAINT ARTISTID FOREIGN KEY (ARTISTID)

REFERENCES ARTIST (ID),

CONSTRAINT ARTISTLIST\_USERINFO\_FK1 FOREIGN KEY (USERID) REFERENCES USERINFO (USERID)

);

1. **ATRIST\_SONGS**

CREATE TABLE ATRIST\_SONGS

( ARTISTID NUMBER NOT NULL,

SONGID NUMBER NOT NULL,

CONSTRAINT ATRIST\_SONGS\_PK PRIMARY KEY (ARTISTID, SONGID),

CONSTRAINT ARTIST\_ID FOREIGN KEY (ARTISTID)

REFERENCES ARTIST (ID),

CONSTRAINT SONG\_ID FOREIGN KEY (SONGID)

REFERENCES SONGS (SONGID)

);

**Insert commands**

* **ARTIST**

INSERT INTO ARTIST (ID, NAME) VALUES ('1', 'Maher Zain')

INSERT INTO ARTIST (ID, NAME) VALUES ('2', 'Samy youssef')

INSERT INTO ARTIST (ID, NAME) VALUES ('3', 'Dema bashar')

INSERT INTO ARTIST (ID, NAME) VALUES ('4', 'El Menshawy')

INSERT INTO ARTIST (ID, NAME) VALUES ('5', 'El hosary')

INSERT INTO ARTIST (ID, NAME) VALUES ('6', 'Ahmed abo zied')

INSERT INTO ARTIST (ID, NAME) VALUES ('7', 'Ahmed samir')

* SONGSCATEGORY

INSERT INTO SONGSCATEGORY (CATEGORYID, CATEGORYNAME) VALUES ('1', 'kids')

INSERT INTO SONGSCATEGORY (CATEGORYID, CATEGORYNAME) VALUES ('2', 'podcast')

INSERT INTO SONGSCATEGORY (CATEGORYID, CATEGORYNAME) VALUES ('3', 'quran')

INSERT INTO SONGSCATEGORY (CATEGORYID, CATEGORYNAME) VALUES ('4', 'english ')

* Songs

INSERT INTO SONGS (SONGID, SONGNAME, CATEGORYID) VALUES ('1', 'mariem', '3','5')

INSERT INTO SONGS (SONGID, SONGNAME, CATEGORYID) VALUES ('2', 'youssef', '3','4')

INSERT INTO SONGS (SONGID, SONGNAME, CATEGORYID) VALUES ('3', 'always be there', '4','1')

INSERT INTO SONGS (SONGID, SONGNAME, CATEGORYID) VALUES ('4', 'supplication', '4','2')

INSERT INTO SONGS (SONGID, SONGNAME, CATEGORYID) VALUES ('5', 'how to learn', '2','6')

INSERT INTO SONGS (SONGID, SONGNAME, CATEGORYID) VALUES ('6', 'dialogue with advanced mind', '2','7')

INSERT INTO SONGS (SONGID, SONGNAME, CATEGORYID) VALUES ('7', 'alphabetic song', '1','3')

INSERT INTO SONGS (SONGID, SONGNAME, CATEGORYID) VALUES ('8', 'mariem', '3','5')

* Subscription Plan:

INSERT INTO SUBSCRIPTIONPLAN (SUBNAME, PRICE, DURATION) VALUES ('regular', '0', '0')

INSERT INTO SUBSCRIPTIONPLAN (SUBNAME, PRICE, DURATION) VALUES ('monthly', '50', '30')

INSERT INTO SUBSCRIPTIONPLAN (SUBNAME, PRICE, DURATION) VALUES ('yearly', '1000', '365')

* User Info

INSERT INTO USERINFO (USERID, USERNAME, STARTDATE, SUBSCRIPTIONNAME) VALUES ('1', 'sama', TO\_DATE('17-SEP-21', 'DD-MON-RR'), 'yearly')

INSERT INTO USERINFO (USERID, USERNAME, STARTDATE, SUBSCRIPTIONNAME) VALUES ('2', 'nouran', TO\_DATE('27-JUN-21', 'DD-MON-RR'), 'yearly')

INSERT INTO USERINFO (USERID, USERNAME, STARTDATE, SUBSCRIPTIONNAME) VALUES ('3', 'menna', TO\_DATE('20-APR-19', 'DD-MON-RR'), 'regular')

INSERT INTO USERINFO (USERID, USERNAME, STARTDATE, SUBSCRIPTIONNAME) VALUES ('4', 'nada', TO\_DATE('11-APR-21', 'DD-MON-RR'), 'regular')

INSERT INTO USERINFO (USERID, USERNAME, STARTDATE, SUBSCRIPTIONNAME) VALUES ('5', 'youmna', TO\_DATE('20-APR-22', 'DD-MON-RR'), 'monthly')

INSERT INTO USERINFO (USERID, USERNAME, STARTDATE, SUBSCRIPTIONNAME) VALUES ('6', 'abeer', TO\_DATE('10-APR-22', 'DD-MON-RR'), 'monthly')

* Artist list:

INSERT INTO ARTISTLIST (ARTISTID, USERID) VALUES ('1', '2')

INSERT INTO ARTISTLIST (ARTISTID, USERID) VALUES ('4', '2')

INSERT INTO ARTISTLIST (ARTISTID, USERID) VALUES ('7', '2')

INSERT INTO ARTISTLIST (ARTISTID, USERID) VALUES ('6', '1')

INSERT INTO ARTISTLIST (ARTISTID, USERID) VALUES ('3', '1')

INSERT INTO ARTISTLIST (ARTISTID, USERID) VALUES ('5', '1')

INSERT INTO ARTISTLIST (ARTISTID, USERID) VALUES ('2', '1')

* Favorite list:

INSERT INTO FAVOURITLIST (SONGID, USERID) VALUES ('1', '2')

INSERT INTO FAVOURITLIST (SONGID, USERID) VALUES ('3', '2')

INSERT INTO FAVOURITLIST (SONGID, USERID) VALUES ('7', '6')

INSERT INTO FAVOURITLIST (SONGID, USERID) VALUES ('5', '1')

INSERT INTO FAVOURITLIST (SONGID, USERID) VALUES ('2', '3')

PROCEDURE

1. ***GETCATEGORYNAME***

CREATE OR REPLACE PROCEDURE GETCATEGORYNAME

( category\_Id IN NUMBER, category\_Name OUT VARCHAR2)

AS

BEGIN

select categoryname

into category\_Name

from songscategory

where category\_Id = categoryid ;

END GETCATEGORYNAME;

1. ***GetArtistName***

CREATE OR REPLACE PROCEDURE GETARTISTNAME

(song\_Id IN NUMBER, artistName OUT VARCHAR2) AS

BEGIN

select ar.name

into artistname

from artist ar , atrist\_songs ars

Where ars.artistid = ar.id and song\_id = ars.songid;

END GETARTISTNAME;

1. ***GetSongs***

CREATE OR REPLACE

PROCEDURE GETSONGS

(artistid in NUMBER , CID out sys\_refcursor)

AS

BEGIN

open CID for

select s.songname

from atrist\_songs ars , songs s

where ars.songid = s.songid and artistid = ars.artistid;

END GETSONGS;