SCHEME RELATIONALE

--USER\_SMR ( user\_id# , user\_name , user\_mail , user\_password , user\_PaypalAccount , user\_register\_date , user\_subscription) --

--PLAYLIST ( playlist\_id# , playlist\_name ) --

CONTRIBUTES ( user\_id# , playlist\_id#,number\_streams )--

--ARTIST (artist\_id# , stage\_name, artist\_last\_name , artist\_first\_name, nationality , prod\_id as FK)--

--SONG (song\_id# , song\_name, release\_date ,song\_duration) --

CONTAINS ( playlist\_id# , song\_id# ) --

--ALBUM ( album\_id# , album\_name ) --

HAS ( album\_id# , song\_id#)--

SUNG\_BY (song\_id# , artist\_id#)--

--GENRE( genre\_id# , genre\_name ) --

BELONGS\_TO ( song\_id# , genre\_id# )

--PRODUCTION\_COMPANY (prod\_id# , company\_name , mail , phone) --

CREARE TABELE

1.

CREATE TABLE USER\_SMR(

user\_id NUMBER(5) PRIMARY KEY,

user\_name VARCHAR(20) NOT NULL UNIQUE,

user\_mail VARCHAR(20) NOT NULL UNIQUE, --unique pt a avea aceeasi adresa de mail pt un singur user

user\_password VARCHAR(20) NOT NULL,

user\_PaypalAccount VARCHAR(20) , --null in caz ca e user pe abonament free

user\_register\_date DATE DEFAULT (sysdate),

user\_subscription NUMBER(1) --tip subscription(0- FREE , 1-BASIC , 2-PREMIUM)

);

2.

--PLAYLIST ( playlist\_id# , playlist\_name )

CREATE TABLE PLAYLIST(

playlist\_id NUMBER(5) PRIMARY KEY,

playlist\_name VARCHAR(55) NOT NULL

);

3.

CREATE TABLE CONTRIBUTES(

user\_id NUMBER(5),

playlist\_id NUMBER(5),

nr\_streams NUMBER(5),

CONSTRAINT PK\_Contributes PRIMARY KEY (user\_id, playlist\_id),

CONSTRAINT FK\_Contributes\_Utilizator FOREIGN KEY (user\_id) REFERENCES User\_SMR(user\_id),

CONSTRAINT FK\_Contributes\_Playlist FOREIGN KEY (playlist\_id) REFERENCES Playlist(playlist\_id)

);

4.

CREATE TABLE ARTIST(

artist\_id NUMBER(5) PRIMARY KEY,

stage\_name VARCHAR(20) NOT NULL,

artist\_last\_name VARCHAR(20), --null pt trupa

artist\_first\_name VARCHAR(20),

nationality VARCHAR(20) ,

prod\_id NUMBER(5) REFERENCES production\_company(prod\_id) ON DELETE SET NULL

);

5.

CREATE TABLE SONG(

song\_id NUMBER(5) PRIMARY KEY,

song\_name VARCHAR(20) NOT NULL,

release\_date DATE,

song\_duration NUMBER(5) -- durata exprimata in secunde

);

6.

CREATE TABLE CONTAINS(

song\_id NUMBER(5),

playlist\_id NUMBER(5),

CONSTRAINT PK\_Contains PRIMARY KEY (song\_id, playlist\_id),

CONSTRAINT FK\_Contains\_Song FOREIGN KEY (song\_id) REFERENCES Song(song\_id),

CONSTRAINT FK\_Contains\_Playlist FOREIGN KEY (playlist\_id) REFERENCES Playlist(playlist\_id)

);

7.

CREATE TABLE ALBUM(

album\_id NUMBER(5) PRIMARY KEY,

album\_name VARCHAR(20) NOT NULL

);

8.

--HAS ( album\_id# , song\_id#)

CREATE TABLE HAS(

song\_id NUMBER(5),

album\_id NUMBER(5),

CONSTRAINT PK\_Has PRIMARY KEY (song\_id, album\_id),

CONSTRAINT FK\_Has\_Song FOREIGN KEY (song\_id) REFERENCES Song(song\_id),

CONSTRAINT FK\_Has\_Album FOREIGN KEY (album\_id) REFERENCES Album(album\_id)

);

9.

--SUNG\_BY (song\_id# , artist\_id#)

CREATE TABLE SUNG\_BY(

song\_id NUMBER(5),

artist\_id NUMBER(5),

CONSTRAINT PK\_Sung\_By PRIMARY KEY (song\_id, artist\_id),

CONSTRAINT FK\_Sung\_By\_Song FOREIGN KEY (song\_id) REFERENCES Song(song\_id),

CONSTRAINT FK\_Sung\_By\_Artist FOREIGN KEY (artist\_id) REFERENCES Artist(artist\_id)

);

10. --GENRE( genre\_id# , genre\_name )

CREATE TABLE GENRE(

genre\_id NUMBER(5) PRIMARY KEY,

genre\_name VARCHAR(20) NOT NULL

);

11. --BELONGS\_TO ( song\_id# , genre\_id# )

CREATE TABLE BELONGS\_TO(

song\_id NUMBER(5),

genre\_id NUMBER(5),

CONSTRAINT PK\_Belongs\_To PRIMARY KEY (song\_id, genre\_id),

CONSTRAINT FK\_Belongs\_To\_Song FOREIGN KEY (song\_id) REFERENCES Song(song\_id),

CONSTRAINT FK\_Belongs\_To\_Genre FOREIGN KEY (genre\_id) REFERENCES Genre(genre\_id)

);

12. --PRODUCTION\_COMPANY (prod\_id# , company\_name , mail , phone)

CREATE TABLE PRODUCTION\_COMPANY(

prod\_id NUMBER(5) PRIMARY KEY,

company\_name VARCHAR(20) NOT NULL,

company\_mail VARCHAR(20) NOT NULL,

company\_phone VARCHAR(20) NOT NULL

);

INSERAM DATELE IN TABELE

1.USER\_SMR

1.

INSERT INTO USER\_SMR (USER\_ID, USER\_NAME,USER\_MAIL , USER\_PASSWORD, user\_paypalaccount, user\_register\_date,user\_subscription)

VALUES (10, 'stanescuraluca','marie02st@gmail.com','parola123','stral@paypal.com', to\_date('2019/12/10', 'RRRR-MM-DD'), 2);

2.

INSERT INTO USER\_SMR (USER\_ID, USER\_NAME,USER\_MAIL , USER\_PASSWORD, user\_paypalaccount, user\_register\_date,user\_subscription)

VALUES (11, 'popescuion','popescuion@gmail.com','parola1234','popion@paypal.com', to\_date('2020/06/10', 'RRRR-MM-DD'), 0);

3.

INSERT INTO USER\_SMR (USER\_ID, USER\_NAME,USER\_MAIL , USER\_PASSWORD, user\_paypalaccount, user\_register\_date,user\_subscription)

VALUES (101, 'marinandrei','marinand@gmail.com','pass1234','marinand@paypal.com', to\_date('2021/06/11', 'RRRR-MM-DD'), 1);

4.

INSERT INTO USER\_SMR (USER\_ID, USER\_NAME,USER\_MAIL , USER\_PASSWORD, user\_paypalaccount, user\_register\_date,user\_subscription)

VALUES (15, 'tomisanda','tomand@aol.com','pass@12','tomand@paypal.com', to\_date('2018/11/07', 'RRRR-MM-DD'), 0);

5.

INSERT INTO USER\_SMR (USER\_ID, USER\_NAME,USER\_MAIL , USER\_PASSWORD, user\_paypalaccount, user\_register\_date,user\_subscription)

VALUES (20, 'rusupaul','rusup@gmail.com','pass@112','rusup@paypal.com', to\_date('2021/01/09', 'RRRR-MM-DD'), 2);

6.

INSERT INTO USER\_SMR (USER\_ID, USER\_NAME,USER\_MAIL , USER\_PASSWORD, user\_paypalaccount, user\_register\_date,user\_subscription)

VALUES (21, 'rosualex','rosual@gmail.com','parola125','rosual@paypal.com', to\_date('2020/01/09', 'RRRR-MM-DD'), 1);

7.

INSERT INTO USER\_SMR (USER\_ID, USER\_NAME,USER\_MAIL , USER\_PASSWORD, user\_paypalaccount, user\_register\_date,user\_subscription)

VALUES (25, 'predadan','prdan@gmail.com','parola@129','prdan@paypal.com', to\_date('2019/02/05', 'RRRR-MM-DD'), 0);

8.

INSERT INTO USER\_SMR (USER\_ID, USER\_NAME,USER\_MAIL , USER\_PASSWORD, user\_paypalaccount, user\_register\_date,user\_subscription)

VALUES (27, 'alexedan','aldan@gmail.com','parola@157','aldan@paypal.com', to\_date('2016/03/07', 'RRRR-MM-DD'), 1);

9.

INSERT INTO USER\_SMR (USER\_ID, USER\_NAME,USER\_MAIL , USER\_PASSWORD, user\_paypalaccount, user\_register\_date,user\_subscription)

VALUES (29, 'vladalex','vlalex@gmail.com','parola@147','vlalex@paypal.com', to\_date('2017/02/10', 'RRRR-MM-DD'), 2);

10.

INSERT INTO USER\_SMR (USER\_ID, USER\_NAME,USER\_MAIL , USER\_PASSWORD, user\_paypalaccount, user\_register\_date,user\_subscription)

VALUES (35, 'mariusandrei','mandrei@gmail.com','parola@150','mand@paypal.com', to\_date('2019/05/11', 'RRRR-MM-DD'), 0);

2. PLAYLIST ( playlist\_id# , playlist\_name )

INSERT INTO PLAYLIST (playlist\_id, playlist\_name)

VALUES (200, 'Happysongs');

INSERT INTO PLAYLIST (playlist\_id, playlist\_name)

VALUES (201, 'Sadsongs');

INSERT INTO PLAYLIST (playlist\_id, playlist\_name)

VALUES (202, 'Rocksongs');

INSERT INTO PLAYLIST (playlist\_id, playlist\_name)

VALUES (203, '70sHits');

INSERT INTO PLAYLIST (playlist\_id, playlist\_name)

VALUES (204, '80sHits');

INSERT INTO PLAYLIST (playlist\_id, playlist\_name)

VALUES (205, 'Top10Today');

INSERT INTO PLAYLIST (playlist\_id, playlist\_name)

VALUES (206, '00sHits');

INSERT INTO PLAYLIST (playlist\_id, playlist\_name)

VALUES (207, ' AlternativeFavs');

INSERT INTO PLAYLIST (playlist\_id, playlist\_name)

VALUES (208, ' AllTimeFavs');

INSERT INTO PLAYLIST (playlist\_id, playlist\_name)

VALUES (209, 'Bestof');

3. --ALBUM ( album\_id# , album\_name )

INSERT INTO ALBUM (album\_id, album\_name)

VALUES (300, 'Is This It');

INSERT INTO ALBUM (album\_id, album\_name)

VALUES (301, 'Vladivostok');

INSERT INTO ALBUM (album\_id, album\_name)

VALUES (302, 'The Wall');

INSERT INTO ALBUM (album\_id, album\_name)

VALUES (303, 'Hot Space');

INSERT INTO ALBUM (album\_id, album\_name)

VALUES (304, 'The Works');

INSERT INTO ALBUM (album\_id, album\_name)

VALUES (305, 'Back In Black');

INSERT INTO ALBUM (album\_id, album\_name)

VALUES (306, 'AM');

INSERT INTO ALBUM (album\_id, album\_name)

VALUES (307, 'The New Abnormal');

INSERT INTO ALBUM (album\_id, album\_name)

VALUES (308, 'From Eden');

INSERT INTO ALBUM (album\_id, album\_name)

VALUES (309, 'Pastel Blues');

4. --SONG (song\_id# , song\_name, release\_date ,song\_duration)

INSERT INTO SONG (song\_id, song\_name,release\_date,song\_duration)

VALUES (400, 'Mad Sounds',to\_date('2013/09/06', 'RRRR-MM-DD'), 210);

INSERT INTO SONG (song\_id, song\_name,release\_date,song\_duration)

VALUES (401, 'Someday',to\_date('2001/07/02', 'RRRR-MM-DD'), 184);

INSERT INTO SONG (song\_id, song\_name,release\_date,song\_duration)

VALUES (402, 'Noi toti',to\_date('2019/10/01', 'RRRR-MM-DD'), 206);

INSERT INTO SONG (song\_id, song\_name,release\_date,song\_duration)

VALUES (403, 'Back in Black',to\_date('1980/10/15', 'RRRR-MM-DD'), 256);

INSERT INTO SONG (song\_id, song\_name,release\_date,song\_duration)

VALUES (404, 'Comfortably Numb',to\_date('1979/04/06', 'RRRR-MM-DD'), 376);

INSERT INTO SONG (song\_id, song\_name,release\_date,song\_duration)

VALUES (405, 'Cool Cat',to\_date('1982/05/21', 'RRRR-MM-DD'), 210);

INSERT INTO SONG (song\_id, song\_name,release\_date,song\_duration)

VALUES (406, 'Hammer To Fall',to\_date('1984/04/19', 'RRRR-MM-DD'), 266);

INSERT INTO SONG (song\_id, song\_name,release\_date,song\_duration)

VALUES (407, 'Bad Decisions',to\_date('2020/06/06', 'RRRR-MM-DD'), 300);

INSERT INTO SONG (song\_id, song\_name,release\_date,song\_duration)

VALUES (408, 'Work Song',to\_date('2014/02/09', 'RRRR-MM-DD'), 229);

INSERT INTO SONG (song\_id, song\_name,release\_date,song\_duration)

VALUES (409, 'Sinnerman',to\_date('1965/03/16', 'RRRR-MM-DD'), 622);

5. --GENRE( genre\_id# , genre\_name )

INSERT INTO GENRE(genre\_id, genre\_name)

VALUES (500, 'Rock');

INSERT INTO GENRE(genre\_id, genre\_name)

VALUES (501, 'Pop');

INSERT INTO GENRE(genre\_id, genre\_name)

VALUES (502, 'Alternative');

INSERT INTO GENRE(genre\_id, genre\_name)

VALUES (503, 'Indie');

INSERT INTO GENRE(genre\_id, genre\_name)

VALUES (504, 'Jazz');

INSERT INTO GENRE(genre\_id, genre\_name)

VALUES (505, 'Blues');

INSERT INTO GENRE(genre\_id, genre\_name)

VALUES (506, 'Folk');

INSERT INTO GENRE(genre\_id, genre\_name)

VALUES (507, 'Dance');

INSERT INTO GENRE(genre\_id, genre\_name)

VALUES (508, 'Heavy Metal');

INSERT INTO GENRE(genre\_id, genre\_name)

VALUES (509, 'Soul');

6. --PRODUCTION\_COMPANY (prod\_id# , company\_name , mail , phone)

INSERT INTO PRODUCTION\_COMPANY(prod\_id, company\_name,company\_mail,company\_phone)

VALUES (600, 'Sony','sony@gmail.com','0798467224');

INSERT INTO PRODUCTION\_COMPANY(prod\_id, company\_name,company\_mail,company\_phone)

VALUES (601, 'Warner','warner@gmail.com','0776469824');

INSERT INTO PRODUCTION\_COMPANY(prod\_id, company\_name,company\_mail,company\_phone)

VALUES (602, 'Universal','universal@gmail.com','0798233224');

INSERT INTO PRODUCTION\_COMPANY(prod\_id, company\_name,company\_mail,company\_phone)

VALUES (603, 'Lion','lion@gmail.com','0709123664');

INSERT INTO PRODUCTION\_COMPANY(prod\_id, company\_name,company\_mail,company\_phone)

VALUES (604, 'SM Entertainment','sm@gmail.com','0798237214');

INSERT INTO PRODUCTION\_COMPANY(prod\_id, company\_name,company\_mail,company\_phone)

VALUES (605, 'TM Production','tm@gmail.com','0797627264');

INSERT INTO PRODUCTION\_COMPANY(prod\_id, company\_name,company\_mail,company\_phone)

VALUES (606, 'Tips Industries','tips@gmail.com','0798467337');

INSERT INTO PRODUCTION\_COMPANY(prod\_id, company\_name,company\_mail,company\_phone)

VALUES (607, 'Creative Impulse','creative@gmail.com','0745465524');

INSERT INTO PRODUCTION\_COMPANY(prod\_id, company\_name,company\_mail,company\_phone)

VALUES (608, 'Cash Money','cashm@gmail.com','0799467325');

INSERT INTO PRODUCTION\_COMPANY(prod\_id, company\_name,company\_mail,company\_phone)

VALUES (609, 'Lang Van','langvan@gmail.com','0799437904');

7. --ARTIST (artist\_id# , stage\_name, artist\_last\_name , artist\_first\_name, nationality , prod\_id as FK)

INSERT INTO ARTIST(artist\_id, stage\_name,artist\_last\_name,artist\_first\_name,nationality,prod\_id)

VALUES (800, 'Nina Simone','Waymon','Eunice Kathleen','American',600);

INSERT INTO ARTIST(artist\_id, stage\_name,artist\_last\_name,artist\_first\_name,nationality,prod\_id)

VALUES (801, 'AC/DC',null,null,'Australian',601);

INSERT INTO ARTIST(artist\_id, stage\_name,artist\_last\_name,artist\_first\_name,nationality,prod\_id)

VALUES (802, 'Arctic Monkeys',null,null,'British',602);

INSERT INTO ARTIST(artist\_id, stage\_name,artist\_last\_name,artist\_first\_name,nationality,prod\_id)

VALUES (803, 'Queen',null,null,'British',603);

INSERT INTO ARTIST(artist\_id, stage\_name,artist\_last\_name,artist\_first\_name,nationality,prod\_id)

VALUES (804, 'Pink Floyd',null,null,'British',604);

INSERT INTO ARTIST(artist\_id, stage\_name,artist\_last\_name,artist\_first\_name,nationality,prod\_id)

VALUES (805, 'Hozier',null,null,'Irish',605);

INSERT INTO ARTIST(artist\_id, stage\_name,artist\_last\_name,artist\_first\_name,nationality,prod\_id)

VALUES (806, 'RATB',null,null,'Romanian',606);

INSERT INTO ARTIST(artist\_id, stage\_name,artist\_last\_name,artist\_first\_name,nationality,prod\_id)

VALUES (807, 'The Strokes',null,null,'British',607);

INSERT INTO ARTIST(artist\_id, stage\_name,artist\_last\_name,artist\_first\_name,nationality,prod\_id)

VALUES (808, 'Glass Animals',null,null,'British',608);

INSERT INTO ARTIST(artist\_id, stage\_name,artist\_last\_name,artist\_first\_name,nationality,prod\_id)

VALUES (809, 'Modest Mouse',null,null,'American',609);

INSERT INTO ARTIST(artist\_id, stage\_name,artist\_last\_name,artist\_first\_name,nationality,prod\_id)

VALUES (810, 'Lou Reed','Reed','Lewis','American',600);

8. Contributes

insert into contributes values(10,200,100);

insert into contributes values(11,201,200);

insert into contributes values(101,202,300);

insert into contributes values(15,203,50);

insert into contributes values(20,204,100);

insert into contributes values(21,205,500);

insert into contributes values(25,206,101);

insert into contributes values(27,207,102);

insert into contributes values(29,208,110);

insert into contributes values(35,209,1000);

9. Contains

insert into contains values(406,200);

insert into contains values(400,201);

insert into contains values(403,202);

insert into contains values(404,203);

insert into contains values(405,204);

insert into contains values(408,205);

insert into contains values(401,206);

insert into contains values(407,207);

insert into contains values(402,208);

insert into contains values(409,209);

10. HAS

insert into has values(401,300);

insert into has values(402,301);

insert into has values(404,302);

insert into has values(405,303);

insert into has values(406,304);

insert into has values(403,305);

insert into has values(400,306);

insert into has values(407,307);

insert into has values(408,308);

insert into has values(409,309);

11. Sung by

insert into sung\_by values(400,802);

insert into sung\_by values(401,807);

insert into sung\_by values(407,807);

insert into sung\_by values(402,806);

insert into sung\_by values(403,801);

insert into sung\_by values(404,804);

insert into sung\_by values(405,803);

insert into sung\_by values(406,803);

insert into sung\_by values(408,805);

insert into sung\_by values(409,800);

12.

insert into belongs\_to values(400,502);

insert into belongs\_to values(401,503);

insert into belongs\_to values(402,502);

insert into belongs\_to values(403,500);

insert into belongs\_to values(403,508);

insert into belongs\_to values(404,500);

insert into belongs\_to values(405,504);

insert into belongs\_to values(405,507);

insert into belongs\_to values(405,500);

insert into belongs\_to values(406,500);

insert into belongs\_to values(407,502);

insert into belongs\_to values(408,509);

insert into belongs\_to values(409,504);

insert into belongs\_to values(409,509);

insert into belongs\_to values(409,505);

CERINTE

**11.join pe >4 tabele si ordonari**

--1.Sa se afiseze artistul, casa de productie, melodiile si genul acestora, sortate descrescator dupa data lansarii melodiei

SELECT a.stage\_name, p.company\_name,s.song\_name, s.release\_date,g.genre\_name

from artist a

join production\_company p on a.prod\_id = p.prod\_id

join sung\_by su on a.artist\_id = su.artist\_id

join song s on su.song\_id = s.song\_id

join belongs\_to b on s.song\_id = b.song\_id

join genre g on b.genre\_id = g.genre\_id

order by 4 desc;

**--CASE**

--2.Sa se afiseze pt toate playlisturile 'ASCULTATE DES' daca au nr of streams

--peste 500, 'ASCULTATE MEDIU' pt 100<nr of streams<500 si

--'ASCULTATE PUTIN' in rest.

SELECT CASE WHEN c.nr\_streams < 100 THEN 'ASCULTAT PUTIN'

WHEN c.nr\_streams <500 THEN 'ASCULTAT MEDIU'

ELSE 'ASCULTAT DES'

END as nr\_ascultari, c.playlist\_id,p.playlist\_name

FROM contributes c

JOIN playlist p on c.playlist\_id = p.playlist\_id;

**--DECODE—**

--3--Sa se afiseze pentru fiecare playlist emotia pe care sunt bazate numele, daca exista.In caz contrar se va afisa 'no emotion detected'.

SELECT DECODE(playlist\_name,'Happysongs','Happy','Sadsongs','Sad','no emotion detected')

FROM playlist

**--NVL--**

--4--Sa se afiseze mesajul 'band' in cazul in care artistul este trupa, in caz contrar sa se afiseze prenumle artistului.

SELECT NVL(artist\_first\_name,'band')

from artist;

**--NVL2--**

--4--Sa se afiseze mesajul 'solo' in cazul in care artistul este solo, in caz contrar sa se afiseze mesajul 'band'.

SELECT NVL2(artist\_first\_name,'solo','band')

from artist;

**--FUNCTII PE SIRURI DE CARACTERE--**

--5--Sa se afiseze informatii despre melodiile ale carui gen muzical este rock.

select \*

from song s

join belongs\_to b on s.song\_id = b.song\_id

join genre g on b.genre\_id = g.genre\_id

where lower(genre\_name) = 'rock'

**+FILTRARE LA NIVEL DE LINII**

--6--Sa se afiseze playlisturile care contin melodii jazz si melodiile corespunzatoare.

select \*

from playlist p

join contains h on p.playlist\_id = h.playlist\_id

join song s on h.song\_id = s.song\_id

join belongs\_to b on s.song\_id=b.song\_id

join genre g on b.genre\_id = g.genre\_id

where upper(genre\_name) = 'JAZZ'

--functii pe date calendaristice--

--7--Sa se afiseze data primei zile de luni dupa datele de lansare din 2020

--NEXT\_DAY(‘02-MAR-2016’, ‘Monday’) = ’07-MAR-2007’

SELECT NEXT\_DAY(s.release\_date, 'Monday')

from song s

where extract (year from s.release\_date) =2020;

--8--Sa se afiseze ultima zi calendaristica a lunilor in care a fost lansat un cantec din 1979

SELECT LAST\_DAY(s.release\_date)

from song s

where extract (year from s.release\_date) =1979;

**--WITH --**

--9--Sa se afiseze informatii despre melodiile care au fost released in acelasi an in care s a inregistrat un user.

WITH inregistrari AS (SELECT DISTINCT TO\_CHAR(user\_register\_date,'YYYY') AS data1

FROM User\_SMR)

SELECT s.song\_name,s.release\_date

FROM inregistrari i, song s

WHERE TO\_CHAR(s.release\_date,'YYYY') IN i.data1;

**--GRUPARI DE DATE.....--**

--Sa se afiseze toate id-urile artistilor care au cel putin 2 melodii si nr melodiilor

select artist\_id, count(song\_id)

from sung\_by

group by artist\_id

having count(song\_id)>=2;

**--SUBCERERI CORELATE--**

--Afisati numele artistilor care au prima litera a numelui de scena egala

--cu cea a usernameului utilizatorilor care au contribuit la un playlist

--cu id ul mai mic de 205.

select a1.stage\_name

from artist a1

where cast(upper(a1.stage\_name) as char(1)) in (

select cast(upper(u.user\_name) as char(1))

from user\_smr u

join contributes c on u.user\_id=c.user\_id

join playlist p on c.playlist\_id=p.playlist\_id

join contains co on p.playlist\_id = co.playlist\_id

join song s on co.song\_id=s.song\_id

join sung\_by sb on s.song\_id = sb.song\_id

join artist a2 on sb.artist\_id = a2.artist\_id

where p.playlist\_id < 205);

**--SUBCERERI NECORELATE—**

--Sa se afiseze username ul utilzatorilor al caror nume are aceeasi lungime cu cea a numelui de scena al unui artist care a scos o melodie dupa 1980.

SELECT U.USER\_NAME

from user\_smr u

where length(u.user\_name) in

(select length(a.stage\_name)

from artist a

join sung\_by s on s.artist\_id = a.artist\_id

join song so on s.song\_id = so.song\_id

where to\_char(so.release\_date,'YYYY')>1980);

**--12--**

Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.

-- 12.1. -- UPDATE Fiecare utilizator reasculta playlistul pe care l-a ascultat cel mai des.

UPDATE Contributes c

SET nr\_streams = nr\_streams + 1

WHERE playlist\_id IN (SELECT playlist\_id

FROM Contributes c1

WHERE c.user\_id = c1. user\_id and

nr\_streams = (SELECT max(nr\_streams)

FROM Contributes c2

WHERE c1.user\_id = c2.user\_id));

12.2--Sa se schimbe paypalul userilor care lungimea paypalului mai mare decat cea mai lunga parola din tabel.

update user\_smr

set user\_PaypalAccount = user\_name

where length(user\_PaypalAccount) >

(SELECT max(length(user\_password))

from User\_SMR);

select \*

from user\_SMR;

rollback;

--12.3--Sa se schimbe parola userilor care este mai scurta decat lungimea maixma a unui user\_name din tabel in user\_name-ul asociat.

update user\_smr

set user\_password = user\_name

where length(user\_password) <

(SELECT max(length(user\_name))

from User\_SMR);

select \*

from user\_SMR;

--rollback;