

Gestiunea unui Magazin de Muzică

Facultatea de Matematica si Informatica

Baze de Date

Grupa: 141

Rachieru Gheorghe Gabriel

April 28, 2025

Contents

1	Descrierea modelului real	2
1.1	Utilitatea sistemului de baza de date	2
2	Prezentarea constrângerilor	2
3	Descrierea entităților	3
4	Descrierea relațiilor	3
4.1	Relatiile de tip One-To-Many si Many-To-Many	3
4.2	Relatia de tip 3	3
5	Descrierea atributelor	4
6	Realizarea diagramei entitate-relație	10
7	Realizarea diagramei conceptuale	11
8	Scheme relaționale	11
9	Realizarea normalizarii FN1-FN3	12
9.1	Forma normala 1 - FN1	12
9.2	Forma normala 2 - FN2	12
9.3	Forma normala 3 - FN3	13
10	Crearea tabelelor in SQL si inserarea de date	14
10.1	RECORD LABEL	14
10.2	PRODUCT	14
10.3	ALBUM	15
10.4	MERCHANDISE	15
10.5	GENRE	15
10.6	ARTIST	16
10.7	PROVIDER	16
10.8	MEDIA FORMAT	17
10.9	FUNCTION	17
10.10	EMPLOYEE	17
10.11	CUSTOMER	18
10.12	STOCK	19
10.13	PURCHASE	19
10.14	SONG	20

1 Descrierea modelului real

Pe zi ce trece, toate afacerile, activitățile și acțiunile iau amploare. Același fenomen se poate observa și în ceea ce privește domeniul muzical. Produsele muzicale devin tot mai diverse și sunt prezentate în atât de multe forme, iar numărul de producători este într-o creștere continuă și, tocmai de aceea, crearea unei baze de date este necesară pentru reținerea acestor informații. Acest proiect a fost creat cu scopul de a gestiona toate datele necesare organizării activității unui magazin de muzică, numit "Rachi Records".

Totodată, am ales această temă deoarece vreau să îmi aprofundez cunoștințele în domeniul bazelor de date, cu precădere în sistemul de gestiune al bazelor de date Oracle. O bază de date bine structurată pentru "Rachi Records" ar permite gestionarea eficientă a stocului de muzică, urmărirea vânzărilor, analiza preferințelor clienților și automatizarea proceselor operaționale. Astfel, magazinul poate oferi o experiență mai bună clienților și poate crește eficiența afacerii în ansamblu.

1.1 Utilitatea sistemului de baza de date

Proiectul "Gestiunea unui magazin de muzică" este o bază de date care monitorizează vânzarea de produse din stocul unui magazin, acestea fiind achiziționate de la diferiți furnizori. Scopul principal al acesteia este de a furniza informații precise și actualizate.

Cu ajutorul bazei de date, sunt înregistrate următoarele informații: detalii despre clienți, comenzile plasate de aceștia, produsele achiziționate de clienți, locația fizică a produselor în magazin, detalii despre angajații fiecărei locații și informații despre furnizorii de produse. Astfel, această bază de date asigură o gestionare eficientă a activităților magazinului de muzică și contribuie la îmbunătățirea experienței clienților.

2 Prezentarea constrângerilor

- Comenzile pot fi procesate între orele 08.00-22.00 de luni până sâmbătă.
- Clienții primesc confirmarea comenzii telefonic sau prin mail.
- Orice client care comandă de mai mult de 150 de lei nu plătește costul livrării.
- Un produs poate fi distribuit de maxim 5 furnizori.
- Prețului de bază al unui produs îi este adăugat "adaos" în funcție de formula media.
- O casa de discuri face mai multe albume sau niciunul.
- Un album are mai multe genuri muzicale si cel puțin unul.
- Un produs, fie el Album sau Merchandise, este asociat mai multor melodii si cel puțin una.
- Mai multi artisti pot inregistra o singura melodie.
- Un album este disponibil in mai multe formate media si cel puțin unul.
- Un produs apare in mai multe stocuri sau in niciunul.
- Un furnizor trimite mai multe stocuri, sau niciunul.
- Un Angajat are o singura functie si cel puțin una.
- Un Angajat se ocupa de o comanda sau de niciuna.
- Un Cumparator a efectuat mai multe comenzi, si cel puțin una.
- O achiziție are un singur produs, cantitatea acestuia, clientul si angajatul care se ocupa de comanda.
- Un angajat se ocupa de mai multe comenzi sau de niciuna.
- Un client are mai multe achizitii, sau niciuna, fiind inscris in baza de date fara a cumpara ceva anterior.
- Un produs poate aparea in mai multe achizitii, sau in niciuna.

3 Descrierea entităților

- Pentru fiecare casa de discuri se cunosc id-ul casei, numele casei si adresa
- Pentru fiecare produs de tip Merchandise se cunoaste id-ul, tipul, numele, si optional marimea.
- Pentru fiecare album se cunosc id-ul albumului, denumirea albumului, data aparitiei, lungimea și pretul de baza.
- Pentru fiecare furnizor se cunosc id-ul furnizorului si denumirea furnizorului
- Pentru fiecare format media se cunosc id-ul formatului și denumirea si adaosul.
- Pentru fiecare artist se cunoaste id-ul artistului, numărul, numele artistului, prenumele artistului si grupul muzical din care face parte.
- Pentru fiecare client se cunosc id-ul clientului, numele și prenumele, numărul de telefon, adresa si localitatea.
- Pentru fiecare angajat se cunosc id-ul angajatului, numele si prenumele, numărul de telefon, adresa, data angajarii si salariu.
- Pentru fiecare angajat se cunosc id-ul angajatului, numele si prenumele, numărul de telefon, adresa, data angajarii si salariu.
- Pentru fiecare functie se cunoaște id-ul functiei, denumirea, salariul minim si maxim.
- Pentru fiecare gen muzical se cunosc id-ul genului si numele.

4 Descrierea relațiilor

4.1 Relatiile de tip One-To-Many si Many-To-Many

Relatie	Cardinalitate	Observatii
Produce	Recording Label-Album: One to Many	-
Exista in	Album-Media Format: One to Many	-
Are/Inregistreaza	Product-Artist: Many to Many	-
Distribuie	Provider-Product: Many to Many	-
Apartine	Product-Genre: One to Many	-
Are o	Employee-Function: One to Many	-

4.2 Relatia de tip 3

Relatia de tip 3 dintre entitatile **PRODUCT**, **CUSTOMER** si **EMPLOYEE** definește procesul prin care un client cumpără produse, iar tranzacția este gestionată de un angajat. Fiecare achiziție este asociată cu un singur angajat, care o procesează, dar un angajat poate gestiona mai multe achiziții. De asemenea, un client poate face mai multe achiziții sau poate exista în sistem fără a fi cumpărat nimic. Această relație ajută la urmărirea istoricului de cumpărături și la gestionarea vânzărilor.

5 Descrierea atributelor

1. ALBUM

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori implicite	Observatii
id_product	NUMBER(3,0)	PK, FK	101, 102, 103	-	Identificator unic
album_name	VARCHAR2(100)	-	GNX, SOS, Swimming	-	Numele albumului ca text
length	CHAR(5)	-	45:30, 38:22, 72:15	-	Format MM:SS
id_record_label	NUMBER(3,0)	FK	1, 2, 3	-	Legat de casa de discuri

2. MERCHANDISE

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori Im-plicite	Observatii
id_product	NUMBER(3,0)	PK, FK	201, 202, 203	-	Identificator unic
merch_type	VARCHAR2(100)	-	T-shirt, Hoodie, Poster, Cap	-	Tipul produsului
merch_size	VARCHAR2(4)	-	S, M, L, XL, XXL	NULL	Obligatoriu doar pentru îmbrăcăminte

3. PRODUCT

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori Im-plicite	Observatii
id_product	NUMBER(3,0)	PK	101, 201, 301	-	Cheia primară a produsului
price	NUMBER(6,2)	-	99.99, 149.99, 24.99	-	Preț de bază în RON
release_date	DATE	-	2024-02-05, 2023-11-12, 2022-08-30	SYSDATE	Data lansării produsului

4. GENRE

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori Im-plicite	Observatii
id_genre	NUMBER(3,0)	PK	1, 2, 3, 4	-	Identificator unic
genre_name	VARCHAR2(100)	-	Rock, Pop, Jazz, Metal, Hip-Hop	-	Numele genului muzical
id_album	NUMBER(3,0)	FK	101, 102, 103	-	Legat de albumul asociat

5. ARTIST

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori implicite	Observatii
id_artist	NUMBER(3,0)	PK	401, 402, 403	-	Identificator unic
first_name	VARCHAR2(100)	-	John, Maria, Alex, Laura	-	Prenumele artistului
last_name	VARCHAR2(100)	-	Doe, Smith, Johnson, Popescu	-	Numele de familie
band	VARCHAR2(100)	-	The Rockers, Metallica, Arctic Monkeys	NULL	Poate fi NULL pentru artiști solo

6. MEDIA FORMAT

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori implicite	Observatii
id_media_format	NUMBER(3,0)	PK	1, 2, 3, 4	-	Identificator unic
media_format_name	VARCHAR2(100)	-	Vinyl, CD, Digital, Cassette	-	Numele formatului media
id_product	NUMBER(3,0)	FK	101, 102, 103	-	Legat de produsul asociat
procent_added	NUMBER(3,0)	-	15, 5, 0, 10	0	Procentul adăugat la preț

7. PROVIDER

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori implicite	Observatii
id_provider	NUMBER(3,0)	PK	1, 2, 3	-	Identificator unic
provider_name	VARCHAR2(100)	-	Music Warehouse, Vinyl Plus, Global Distribution	-	Numele furnizorului
phone_number	CHAR(10)	-	0712345678, 0723456789, 0734567890	-	Format românesc
adresa	VARCHAR2(100)	-	Str. Muzicii 10, Bd. Distribuției 25, Alea Furnizorilor 5	-	Adresa furnizorului

8. FUNCTION

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori implicite	Observatii
id_function	NUMBER(3,0)	PK	1, 2, 3, 4	-	Identificator unic
function_name	VARCHAR2(100)	-	Manager, Sales Associate, Cashier, Inventory Specialist	-	Numele funcției
min_salary	NUMBER(6,2)	-	2500.00, 3000.00, 4000.00	-	Salariul minim în RON
max_salary	NUMBER(6,2)	-	5000.00, 7000.00, 10000.00	-	Salariul maxim în RON

9. EMPLOYEE

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori implicite	Observatii
id_employee	NUMBER(3,0)	PK	1, 2, 3, 4	-	Identificator unic
id_function	NUMBER(3,0)	FK	1, 2, 3, 4	-	Legat de funcția angajatului
last_name	VARCHAR2(100)	-	Ionescu, Popescu, Popa, Smith	-	Numele de familie
first_name	VARCHAR2(100)	-	Ion, Maria, Ana, John	-	Prenumele angajatului
min_salary	NUMBER(6,2)	-	2500.00, 3000.00, 4000.00	-	Salariul minim pentru funcție
max_salary	NUMBER(6,2)	-	5000.00, 7000.00, 10000.00	-	Salariul maxim pentru funcție
phone	CHAR(11)	-	0712345678, 0723456789, 0734567890	-	Format românesc
hire_date	DATE	-	2020-05-15, 2022-03-10, 2023-08-22	SYSDATE	Data angajării
salary	NUMBER(6,2)	-	3500.00, 4200.00, 6000.00	-	Salariul curent

10. CUSTOMER

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori implicite	Observatii
id_customer	NUMBER(3,0)	PK	1, 2, 3, 4	-	Identificator unic
last_name	VARCHAR2(100)	-	Georgescu, Vasilescu, Brown, Johnson	-	Numele de familie
first_name	VARCHAR2(100)	-	Andrei, Elena, Robert, Sarah	-	Prenumele clientului
adresa	VARCHAR2(300)	-	Str. Primăverii 10, Bd. Tineretului 25, Main Street 42	-	Adresa completă
city	VARCHAR2(100)	-	București, Cluj-Napoca, Iași, London, New York	-	Orașul clientului

11. RECORD LABEL

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori implicite	Observatii
id_record_label	NUMBER(3,0)	PK	1, 2, 3, 4	-	Identificator unic
label_name	VARCHAR2(100)	-	Universal Music, Sony Music, Warner Records	-	Numele casei de discuri
email	VARCHAR2(100)	-	contact@universal.com, info@sony.com, support@warner.com	-	Email de contact

12. PURCHASE

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori implicite	Observatii
id_purchase	NUMBER(3,0)	PK	1001, 1002, 1003	-	Identificator unic
id_customer	NUMBER(3,0)	FK	1, 2, 3, 4	-	Legat de clientul care cumpără
id_employee	NUMBER(3,0)	FK	1, 2, 3, 4	-	Legat de angajatul care procesează
id_product	NUMBER(3,0)	FK	101, 201, 301	-	Legat de produsul cumpărat
purchase_date	DATE	-	2024-03-15, 2024-02-28, 2024-01-10	SYSDATE	Data achiziției
payment_method	VARCHAR2(100)	-	Card, Cash, PayPal, Transfer bancar	'Card'	Metoda de plată
status	VARCHAR2(100)	-	Pending, Completed, Cancelled, Delivered	'Pending'	Statusul comenzii
quantity	NUMBER(3,0)	-	1, 2, 3, 5, 10	1	Cantitatea cumpărată

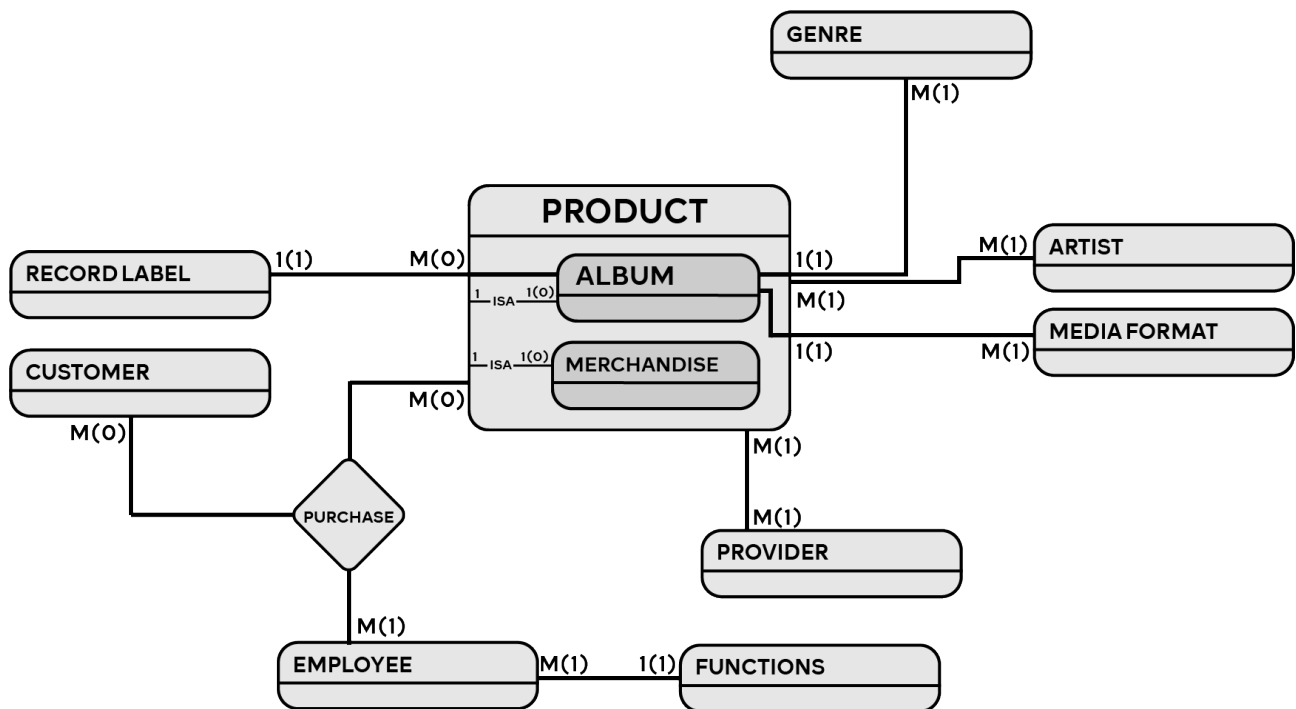
13. STOCK

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori implicite	Observatii
id_stock	NUMBER(3,0)	PK	501, 502, 503	-	Identificator unic
id_product	NUMBER(3,0)	FK	101, 201, 301	-	Legat de produsul în stoc
id_provider	NUMBER(3,0)	FK	1, 2, 3	-	Legat de furnizorul produsului
quantity	NUMBER(3,0)	-	10, 25, 50, 100	0	Cantitatea disponibilă

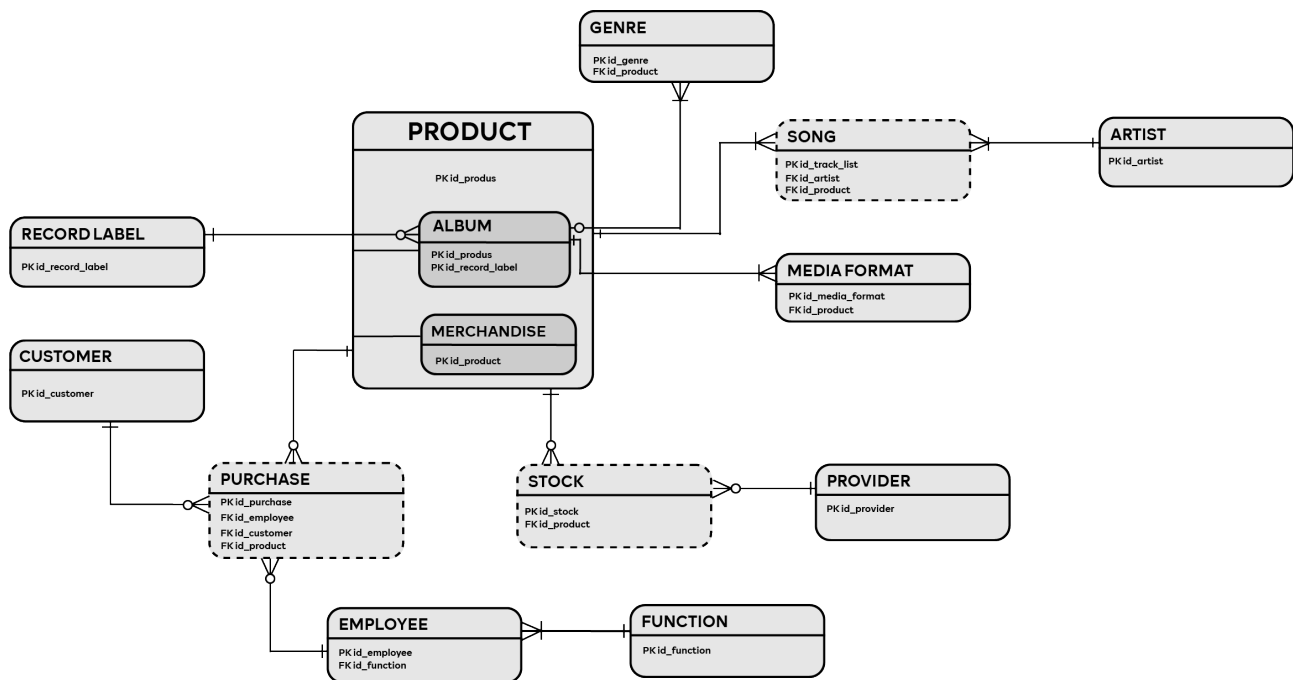
14. SONG

Atribut	Tip de Date	Constrangeri	Valori Posibile	Valori im-plicite	Obersvatii
id_song	NUMBER(3,0)	PK	601, 602, 603	-	Identificator unic
id_artist	NUMBER(3,0)	FK	401, 402, 403	-	Legat de artistul melodiei
id_product	NUMBER(3,0)	FK	101, 102, 103	-	Legat de produsul (albumul)
title	VARCHAR2(100)	-	Summer Vibes, Rock Forever, Soul of Mine	-	Titlul melodiei
length	VARCHAR2(100)	-	3:45, 4:20, 2:55, 6:10	-	Durata melodiei (MM:SS)

6 Realizarea diagramei entitate-relație



7 Realizarea diagramei conceptuale



8 Scheme relaționale

Schemele relaționale corespunzătoare diagramei conceptuale sunt următoarele:

ALBUM(id_product#, length, id_record_label)

MERCHANDISE(id_product#, merch_type, merch_size)

PRODUCT(id_product#, price, release_date)

GENRE(id_genre#, genre_name, id_album)

ARTIST(id_artist#, first_name, last_name, band)

MEDIA_FORMAT(id_media_format#, media_format_name, id_product, procent_added)

PROVIDER(id_provider#, provider_name, phone_number, adress)

FUNCTION(id_function#, function_name, min_salary, max_salary)

EMPLOYEE(id_employee#, id_function, last_name, first_name, phone, hire_date, salary)

CUSTOMER(id_customer#, last_name, first_name, adress, city)

RECORD LABEL(id_record_label#, label_name, email)

PURCHASE(id_purchase#, id_customer, id_employee, id_product, purchase_date, payment_method, status, quantity)

STOCK(id_stock#, id_product, id_provider, quantity)

SONG(id_song#, id_artist, id_product, title, length)

9 Realizarea normalizării FN1-FN3

9.1 Forma normala 1 - FN1

Forma normală 1 (FN1) impune ca toate valorile stocate într-un câmp al unei baze de date să fie atomice, adică să conțină o singură valoare, nu liste sau grupuri de date. Totodată, fiecare rând din tabel trebuie să poată fi identificat în mod unic, prin intermediul unei chei primare.

În modelul implementat, cerințele acestei forme normale sunt respectate: nu există câmpuri care conțin mai multe valori, iar fiecare înregistrare este identificabilă în mod clar cu ajutorul unei chei primare.

Pentru a ilustra această etapă de normalizare, se poate analiza următorul exemplu:

ARTIST	SONG
Artist1	SG1, SG2, SG3
Artist2	SG2, SG4
Artist3	SG3

Table 1: Exemplu non-FN1.

ARTIST	SONG
Artist1	SG1
Artist1	SG2
Artist1	SG3
Artist2	SG2
Artist2	SG4
Artist3	SG3

Table 2: Exemplu FN1.

9.2 Forma normala 2 - FN2

O relație se află în a doua formă normală (FN2) doar dacă este deja în forma normală 1 (FN1) și dacă fiecare atribut care nu face parte din cheia primară depinde de întreaga cheie, nu doar de o parte a acesteia.

Mai exact, FN2 presupune ca toate atributele non-cheie dintr-o tabelă să fie dependente funcțional de cheia primară în întregime, nu parțial.

În cazul modelului implementat, sunt îndeplinite condițiile formei normale a doua: relațiile sunt în FN1, iar toate atributele care nu fac parte din cheile primare sunt dependente în mod complet de acestea.

Pentru a ilustra procesul de normalizare, vom analiza exemplul următor:

id_artist#	first_name	id_product#	price
A1	Kendrick	P1	6.99
A2	West	P1	6.99
A1	Kendrick	P2	15.99
A3	Miller	P3	10.49

Table 3: Exemplu non-FN2.

Observăm că avem următoarele dependente:

{id_artist#} - first_name. id_artist determină funcțional numele.

{id_product#} - price

id_artist#	id_product#	price
A1	P1	6.99
A2	P1	6.99
A1	P2	15.99
A3	P3	10.49

Table 4: Exemplu FN2.

id_artist#	first_name
A1	Kendrick
A2	West
A3	Miller

Table 5: Exemplu FN2.

9.3 Forma normala 3 - FN3

Asemănător, o relație se află în forma normală 3 (FN3) doar dacă este deja în forma normală 2 (FN2) și fiecare atribut care nu face parte din cheia primară este dependent direct de cheia primară, fără să existe dependențe tranzitive.

Cu alte cuvinte, FN3 cere ca orice atribut non-cheie să depindă doar de cheia primară, de întreaga cheie și exclusiv de aceasta, fără intermediari prin alte atribute.

În cadrul modelului implementat, se respectă toate cerințele formei normale a treia: toate atributele care nu sunt chei sunt dependente direct de cheia primară.

Pentru a ilustra această normalizare, vom analiza exemplul următor:

id_artist#	id_product#	price
A1	P1	6.99
A2	P1	6.99
A1	P2	15.99
A3	P3	10.49

Table 6: Exemplu non-FN3.

Pentru a aduce relația R în forma normală 3 (FN3), se elimină dependențele funcționale tranzitive, astfel încât toate atributele non-cheie să depindă direct și exclusiv de cheia primară.

R1(id_product#, price)

R2(id_artist#, id_product#)

id_product#	price
P1	6.99
P2	15.99
P3	10.49

Table 7: Exemplu FN3.

id_artist#	id_product#
A1	P1
A2	P1
A1	P2
A3	P3

Table 8: Exemplu FN3.

10 Crearea tabelelor in SQL si inserarea de date

10.1 RECORD LABEL

```
CREATE TABLE RECORD_LABEL (  
    id_record_label NUMBER(3,0) CONSTRAINT pk_record_label PRIMARY KEY,  
    label_name VARCHAR2(100),  
    email VARCHAR2(100)  
);  
  
CREATE SEQUENCE SEQ_RECORD_LABEL  
    INCREMENT BY 1  
    START WITH 1  
    MAXVALUE 1000  
    NOCYCLE;  
  
INSERT INTO RECORD_LABEL VALUES (SEQ_RECORD_LABEL.NEXTVAL, 'Universal Music',  
'contact@universalmusic.com');  
INSERT INTO RECORD_LABEL VALUES (SEQ_RECORD_LABEL.NEXTVAL, 'Sony Music',  
'info@sonymusic.com');  
INSERT INTO RECORD_LABEL VALUES (SEQ_RECORD_LABEL.NEXTVAL, 'Republic Records',  
'office@republicrecords.com');  
INSERT INTO RECORD_LABEL VALUES (SEQ_RECORD_LABEL.NEXTVAL, 'Atlantic Records',  
'info@atlanticrecords.com');  
INSERT INTO RECORD_LABEL VALUES (SEQ_RECORD_LABEL.NEXTVAL, 'Electrecord',  
'contact@electrecord.ro');  
  
COMMIT;
```

10.2 PRODUCT

```
CREATE TABLE PRODUCT (  
    id_product NUMBER(3,0) CONSTRAINT pk_product PRIMARY KEY,  
    price NUMBER(6,2),  
    release_date DATE DEFAULT SYSDATE  
);  
  
CREATE SEQUENCE SEQ_PRODUCT  
    INCREMENT BY 1  
    START WITH 1  
    MAXVALUE 1000  
    NOCYCLE;  
  
INSERT INTO PRODUCT VALUES (SEQ_PRODUCT.NEXTVAL, 49.99, TO_DATE('15/03/2022', 'DD/MM/YYYY'));  
INSERT INTO PRODUCT VALUES (SEQ_PRODUCT.NEXTVAL, 49.99, TO_DATE('15/03/2022', 'DD/MM/YYYY'));  
INSERT INTO PRODUCT VALUES (SEQ_PRODUCT.NEXTVAL, 39.99, TO_DATE('22/05/2022', 'DD/MM/YYYY'));  
INSERT INTO PRODUCT VALUES (SEQ_PRODUCT.NEXTVAL, 29.99, TO_DATE('10/07/2022', 'DD/MM/YYYY'));  
INSERT INTO PRODUCT VALUES (SEQ_PRODUCT.NEXTVAL, 19.99, TO_DATE('30/09/2022', 'DD/MM/YYYY'));  
INSERT INTO PRODUCT VALUES (SEQ_PRODUCT.NEXTVAL, 59.99, TO_DATE('12/12/2022', 'DD/MM/YYYY'));  
INSERT INTO PRODUCT VALUES (SEQ_PRODUCT.NEXTVAL, 34.99, TO_DATE('25/01/2023', 'DD/MM/YYYY'));  
INSERT INTO PRODUCT VALUES (SEQ_PRODUCT.NEXTVAL, 44.99, TO_DATE('18/03/2023', 'DD/MM/YYYY'));  
INSERT INTO PRODUCT VALUES (SEQ_PRODUCT.NEXTVAL, 24.99, TO_DATE('05/05/2023', 'DD/MM/YYYY'));  
INSERT INTO PRODUCT VALUES (SEQ_PRODUCT.NEXTVAL, 64.99, TO_DATE('28/06/2023', 'DD/MM/YYYY'));  
INSERT INTO PRODUCT VALUES (SEQ_PRODUCT.NEXTVAL, 54.99, TO_DATE('15/08/2023', 'DD/MM/YYYY'));  
  
COMMIT;
```

10.3 ALBUM

```
CREATE TABLE ALBUM (  
    id_product NUMBER(3,0) CONSTRAINT pk_album PRIMARY KEY,  
    album_name VARCHAR2(100), -- Adăugat numele albumului conform documentației  
    length CHAR(5), -- Modificat tipul de date conform documentației  
    id_record_label NUMBER(3,0),  
    CONSTRAINT fk_album_product FOREIGN KEY (id_product) REFERENCES PRODUCT(id_product),  
    CONSTRAINT fk_album_label FOREIGN KEY (id_record_label) REFERENCES RECORD_LABEL(id_record_label)  
);  
  
INSERT INTO ALBUM VALUES (1, 'Swimming', '49:07', 1);  
INSERT INTO ALBUM VALUES (2, 'GNX', '42:11', 2);  
INSERT INTO ALBUM VALUES (3, 'Short n Sweet', '42:19', 3);  
INSERT INTO ALBUM VALUES (4, 'HUMBLE', '43:00', 4);  
INSERT INTO ALBUM VALUES (5, 'folklore', '45:30', 5);  
INSERT INTO ALBUM VALUES (6, 'the album', '30:45', 5);  
  
COMMIT;
```

10.4 MERCHANDISE

```
CREATE TABLE MERCHANDISE (  
    id_product NUMBER(3,0) CONSTRAINT pk_merchandise PRIMARY KEY,  
    merch_type VARCHAR2(100),  
    merch_size VARCHAR2(4),  
    CONSTRAINT fk_merchandise_product FOREIGN KEY (id_product) REFERENCES PRODUCT(id_product)  
);  
  
INSERT INTO MERCHANDISE VALUES (6, 'T-Shirt', 'M');  
INSERT INTO MERCHANDISE VALUES (7, 'Poster', NULL);  
INSERT INTO MERCHANDISE VALUES (8, 'Cap', 'S');  
INSERT INTO MERCHANDISE VALUES (9, 'Mug', NULL);  
INSERT INTO MERCHANDISE VALUES (10, 'Hoodie', 'XL');  
  
COMMIT;
```

10.5 GENRE

```
CREATE TABLE GENRE (  
    id_genre NUMBER(3,0) CONSTRAINT pk_genre PRIMARY KEY,  
    genre_name VARCHAR2(100),  
    id_album NUMBER(3,0),  
    CONSTRAINT fk_genre_album FOREIGN KEY (id_album) REFERENCES ALBUM(id_product)  
);  
  
CREATE SEQUENCE SEQ_GENRE  
    INCREMENT BY 1  
    START WITH 1  
    MAXVALUE 1000  
    NOCYCLE;  
  
INSERT INTO GENRE VALUES (SEQ_GENRE.NEXTVAL, 'RNB', 1);  
INSERT INTO GENRE VALUES (SEQ_GENRE.NEXTVAL, 'Rap', 2);  
INSERT INTO GENRE VALUES (SEQ_GENRE.NEXTVAL, 'Pop', 3);  
INSERT INTO GENRE VALUES (SEQ_GENRE.NEXTVAL, 'Rap-RNB', 4);  
INSERT INTO GENRE VALUES (SEQ_GENRE.NEXTVAL, 'Folk', 5);  
  
COMMIT;
```


10.6 ARTIST

```
CREATE TABLE ARTIST (  
    id_artist NUMBER(3,0) CONSTRAINT pk_artist PRIMARY KEY,  
    first_name VARCHAR2(100),  
    last_name VARCHAR2(100),  
    band VARCHAR2(100)  
);  
  
CREATE SEQUENCE SEQ_ARTIST  
    INCREMENT BY 1  
    START WITH 1  
    MAXVALUE 1000  
    NOCYCLE;  
  
INSERT INTO ARTIST VALUES (1, 'Mac', 'Miller', NULL);  
INSERT INTO ARTIST VALUES (2, 'Kendrick', 'Lamar', NULL);  
INSERT INTO ARTIST VALUES (3, 'Sabrina', 'Carpenter', NULL);  
INSERT INTO ARTIST VALUES (4, 'Taylor', 'Swift', NULL);  
INSERT INTO ARTIST VALUES (5, 'boygenius', 'boygenius', 'band');  
INSERT INTO ARTIST VALUES (6, 'Rihanna', NULL, NULL);  
INSERT INTO ARTIST VALUES (7, 'Frank', 'Ocean', NULL);  
INSERT INTO ARTIST VALUES (8, 'Billie', 'Eilish', NULL);  
INSERT INTO ARTIST VALUES (9, 'The Weeknd', NULL, NULL);  
INSERT INTO ARTIST VALUES (10, 'SZA', NULL, NULL);  
  
COMMIT;
```

10.7 PROVIDER

```
CREATE TABLE PROVIDER (  
    id_provider NUMBER(3,0) CONSTRAINT pk_provider PRIMARY KEY,  
    provider_name VARCHAR2(100),  
    phone_number CHAR(11),  
    address VARCHAR2(100)  
);  
  
CREATE SEQUENCE SEQ_PROVIDER  
    INCREMENT BY 1  
    START WITH 1  
    MAXVALUE 1000  
    NOCYCLE;  
  
INSERT INTO PROVIDER VALUES (SEQ_PROVIDER.NEXTVAL, 'Music Distribution SRL', '07456123890',  
    'Str. Muzicii nr. 15, București');  
INSERT INTO PROVIDER VALUES (SEQ_PROVIDER.NEXTVAL, 'Vinyl Factory', '07345678912',  
    'Calea Victoriei nr. 78, București');  
INSERT INTO PROVIDER VALUES (SEQ_PROVIDER.NEXTVAL, 'Merch Production', '07234567891',  
    'Bd. Unirii nr. 45, București');  
INSERT INTO PROVIDER VALUES (SEQ_PROVIDER.NEXTVAL, 'Global Logistics', '07123456789',  
    'Str. Industriilor nr. 12, Cluj-Napoca');  
INSERT INTO PROVIDER VALUES (SEQ_PROVIDER.NEXTVAL, 'Record Distributor', '07987654321',  
    'Str. Republicii nr. 36, Timișoara');  
INSERT INTO PROVIDER VALUES (SEQ_PROVIDER.NEXTVAL, 'Audio Tech SRL', '07765432109',  
    'Str. Aviatiei nr. 24, București');  
INSERT INTO PROVIDER VALUES (SEQ_PROVIDER.NEXTVAL, 'Music Warehouse', '07654321098',  
    'Calea Floreasca nr. 55, București');  
  
COMMIT;
```

10.8 MEDIA FORMAT

```
CREATE TABLE MEDIA_FORMAT (  
    id_media_format NUMBER(3,0) CONSTRAINT pk_media_format PRIMARY KEY,  
    media_format_name VARCHAR2(100),  
    id_product NUMBER(3,0),  
    procent_added NUMBER(3,0) DEFAULT 0,  
    CONSTRAINT fk_media_format_product FOREIGN KEY (id_product) REFERENCES PRODUCT(id_product)  
);
```

```
CREATE SEQUENCE SEQ_MEDIA_FORMAT  
    INCREMENT BY 1  
    START WITH 1  
    MAXVALUE 1000  
    NOCYCLE;
```

```
INSERT INTO MEDIA_FORMAT VALUES (SEQ_MEDIA_FORMAT.NEXTVAL, 'CD', 1, 10);  
INSERT INTO MEDIA_FORMAT VALUES (SEQ_MEDIA_FORMAT.NEXTVAL, 'Vinyl 12"', 2, 20);  
INSERT INTO MEDIA_FORMAT VALUES (SEQ_MEDIA_FORMAT.NEXTVAL, 'Digital', 3, 0);  
INSERT INTO MEDIA_FORMAT VALUES (SEQ_MEDIA_FORMAT.NEXTVAL, 'Cassette', 4, 5);  
INSERT INTO MEDIA_FORMAT VALUES (SEQ_MEDIA_FORMAT.NEXTVAL, 'Vinyl 7"', 15, 25);
```

```
COMMIT;
```

10.9 FUNCTION

```
CREATE TABLE FUNCTION (  
    id_function NUMBER(3,0) CONSTRAINT pk_function PRIMARY KEY,  
    function_name VARCHAR2(100),  
    min_salary NUMBER(6,2),  
    max_salary NUMBER(6,2)  
);
```

```
CREATE SEQUENCE SEQ_FUNCTION  
    INCREMENT BY 1  
    START WITH 1  
    MAXVALUE 1000  
    NOCYCLE;
```

```
INSERT INTO FUNCTION VALUES (SEQ_FUNCTION.NEXTVAL, 'Manager', 5000, 8000);  
INSERT INTO FUNCTION VALUES (SEQ_FUNCTION.NEXTVAL, 'Sales Representative', 3000, 5000);  
INSERT INTO FUNCTION VALUES (SEQ_FUNCTION.NEXTVAL, 'Cashier', 2500, 3500);  
INSERT INTO FUNCTION VALUES (SEQ_FUNCTION.NEXTVAL, 'Marketing Specialist', 4000, 6000);  
INSERT INTO FUNCTION VALUES (SEQ_FUNCTION.NEXTVAL, 'Store Assistant', 2000, 3000);  
INSERT INTO FUNCTION VALUES (SEQ_FUNCTION.NEXTVAL, 'IT Support', 3500, 5500);  
INSERT INTO FUNCTION VALUES (SEQ_FUNCTION.NEXTVAL, 'Inventory Manager', 4500, 6500);
```

```
COMMIT;
```

10.10 EMPLOYEE

```
CREATE TABLE EMPLOYEE (  
    id_employee NUMBER(3,0) CONSTRAINT pk_employee PRIMARY KEY,  
    id_function NUMBER(3,0),  
    last_name VARCHAR2(100),  
    first_name VARCHAR2(100),  
    phone CHAR(11),  
    hire_date DATE DEFAULT SYSDATE,  
    salary NUMBER(6,2),
```

```
CONSTRAINT fk_employee_function FOREIGN KEY (id_function) REFERENCES FUNCTION(id_function)
);

CREATE SEQUENCE SEQ_EMPLOYEE
  INCREMENT BY 1
  START WITH 1
  MAXVALUE 1000
  NOCYCLE;

INSERT INTO EMPLOYEE VALUES (SEQ_EMPLOYEE.NEXTVAL, 1, 'Popescu', 'Maria', '07123456789',
TO_DATE('15/01/2020', 'DD/MM/YYYY'), 6500);
INSERT INTO EMPLOYEE VALUES (SEQ_EMPLOYEE.NEXTVAL, 2, 'Ionescu', 'Andrei', '07234567890',
TO_DATE('10/03/2020', 'DD/MM/YYYY'), 4000);
INSERT INTO EMPLOYEE VALUES (SEQ_EMPLOYEE.NEXTVAL, 3, 'Popa', 'Elena', '07345678901',
TO_DATE('22/05/2021', 'DD/MM/YYYY'), 3000);
INSERT INTO EMPLOYEE VALUES (SEQ_EMPLOYEE.NEXTVAL, 4, 'Georgescu', 'Mihai', '07456789012',
TO_DATE('17/08/2021', 'DD/MM/YYYY'), 5000);
INSERT INTO EMPLOYEE VALUES (SEQ_EMPLOYEE.NEXTVAL, 5, 'Stanciu', 'Cristina', '07567890123',
TO_DATE('05/12/2021', 'DD/MM/YYYY'), 2500);
INSERT INTO EMPLOYEE VALUES (SEQ_EMPLOYEE.NEXTVAL, 6, 'Dumitrache', 'Alexandru', '07678901234',
TO_DATE('10/02/2022', 'DD/MM/YYYY'), 4500);
INSERT INTO EMPLOYEE VALUES (SEQ_EMPLOYEE.NEXTVAL, 7, 'Marinescu', 'Diana', '07789012345',
TO_DATE('18/04/2022', 'DD/MM/YYYY'), 5500);

COMMIT;
```

10.11 CUSTOMER

```
CREATE TABLE CUSTOMER (
  id_customer NUMBER(3,0) CONSTRAINT pk_customer PRIMARY KEY,
  last_name VARCHAR2(100),
  first_name VARCHAR2(100),
  address VARCHAR2(300),
  city VARCHAR2(100)
);

CREATE SEQUENCE SEQ_CUSTOMER
  INCREMENT BY 1
  START WITH 1
  MAXVALUE 1000
  NOCYCLE;

INSERT INTO CUSTOMER VALUES (SEQ_CUSTOMER.NEXTVAL, 'Dumitrescu', 'Alexandru',
'Str. Libertății nr. 10, București', 'București');
INSERT INTO CUSTOMER VALUES (SEQ_CUSTOMER.NEXTVAL, 'Radu', 'Ioana',
'Str. Unirii nr. 25, Cluj-Napoca', 'Cluj-Napoca');
INSERT INTO CUSTOMER VALUES (SEQ_CUSTOMER.NEXTVAL, 'Stoica', 'Gabriel',
'Bd. Independenței nr. 15, Iași', 'Iași');
INSERT INTO CUSTOMER VALUES (SEQ_CUSTOMER.NEXTVAL, 'Munteanu', 'Diana',
'Str. Mihai Viteazul nr. 8, Timișoara', 'Timișoara');
INSERT INTO CUSTOMER VALUES (SEQ_CUSTOMER.NEXTVAL, 'Dinu', 'Bogdan',
'Str. Primăverii nr. 12, Brașov', 'Brașov');
INSERT INTO CUSTOMER VALUES (SEQ_CUSTOMER.NEXTVAL, 'Vasile', 'Andreea',
'Str. Florilor nr. 5, Constanța', 'Constanța');
INSERT INTO CUSTOMER VALUES (SEQ_CUSTOMER.NEXTVAL, 'Preda', 'Robert',
'Str. Tudor Vladimirescu nr. 18, Craiova', 'Craiova');
INSERT INTO CUSTOMER VALUES (SEQ_CUSTOMER.NEXTVAL, 'Manole', 'Cristina',
'Str. Mioritei nr. 7, Sibiu', 'Sibiu');
```

```
COMMIT;
```

10.12 STOCK

```
CREATE TABLE STOCK (  
    id_stock NUMBER(3,0) CONSTRAINT pk_stock PRIMARY KEY,  
    id_product NUMBER(3,0),  
    id_provider NUMBER(3,0),  
    quantity NUMBER(3,0) DEFAULT 0,  
    CONSTRAINT fk_stock_product FOREIGN KEY (id_product) REFERENCES PRODUCT(id_product),  
    CONSTRAINT fk_stock_provider FOREIGN KEY (id_provider) REFERENCES PROVIDER(id_provider)  
);
```

```
CREATE SEQUENCE SEQ_STOCK  
    INCREMENT BY 1  
    START WITH 1  
    MAXVALUE 1000  
    NOCYCLE;
```

```
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 1, 1, 50);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 2, 2, 30);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 3, 1, 100);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 4, 2, 20);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 5, 1, 40);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 6, 3, 75);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 7, 3, 60);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 8, 3, 90);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 9, 3, 45);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 10, 3, 30);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 11, 1, 25);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 12, 2, 15);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 13, 1, 35);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 14, 3, 50);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 15, 2, 40);  
INSERT INTO STOCK VALUES (SEQ_STOCK.NEXTVAL, 16, 1, 30);
```

```
COMMIT;
```

10.13 PURCHASE

```
CREATE TABLE PURCHASE (  
    id_purchase NUMBER(3,0) CONSTRAINT pk_purchase PRIMARY KEY,  
    id_customer NUMBER(3,0),  
    id_employee NUMBER(3,0),  
    id_product NUMBER(3,0),  
    purchase_date DATE DEFAULT SYSDATE,  
    payment_method VARCHAR2(100) DEFAULT 'Card',  
    status VARCHAR2(100) DEFAULT 'Pending',  
    quantity NUMBER(3,0) DEFAULT 1,  
    CONSTRAINT fk_purchase_customer FOREIGN KEY (id_customer) REFERENCES CUSTOMER(id_customer),  
    CONSTRAINT fk_purchase_employee FOREIGN KEY (id_employee) REFERENCES EMPLOYEE(id_employee),  
    CONSTRAINT fk_purchase_product FOREIGN KEY (id_product) REFERENCES PRODUCT(id_product)  
);
```

```
CREATE SEQUENCE SEQ_PURCHASE  
    INCREMENT BY 1  
    START WITH 1  
    MAXVALUE 1000
```

NOCYCLE;

```
INSERT INTO PURCHASE VALUES (SEQ_PURCHASE.NEXTVAL, 1, 2, 1, TO_DATE('20/04/2023', 'DD/MM/YYYY'),
'Card', 'Completed', 1);
INSERT INTO PURCHASE VALUES (SEQ_PURCHASE.NEXTVAL, 2, 2, 3, TO_DATE('25/04/2023', 'DD/MM/YYYY'),
'Cash', 'Completed', 1);
INSERT INTO PURCHASE VALUES (SEQ_PURCHASE.NEXTVAL, 3, 3, 6, TO_DATE('30/04/2023', 'DD/MM/YYYY'),
'Card', 'Completed', 2);
INSERT INTO PURCHASE VALUES (SEQ_PURCHASE.NEXTVAL, 4, 2, 2, TO_DATE('05/05/2023', 'DD/MM/YYYY'),
'Card', 'Pending', 1);
INSERT INTO PURCHASE VALUES (SEQ_PURCHASE.NEXTVAL, 5, 3, 8, TO_DATE('10/05/2023', 'DD/MM/YYYY'),
'Cash', 'Pending', 3);
INSERT INTO PURCHASE VALUES (SEQ_PURCHASE.NEXTVAL, 1, 2, 5, TO_DATE('15/05/2023', 'DD/MM/YYYY'),
'Card', 'Completed', 1);
INSERT INTO PURCHASE VALUES (SEQ_PURCHASE.NEXTVAL, 2, 3, 10, TO_DATE('20/05/2023', 'DD/MM/YYYY'),
'Cash', 'Cancelled', 1);
INSERT INTO PURCHASE VALUES (SEQ_PURCHASE.NEXTVAL, 3, 2, 4, TO_DATE('25/05/2023', 'DD/MM/YYYY'),
'Card', 'Completed', 1);
INSERT INTO PURCHASE VALUES (SEQ_PURCHASE.NEXTVAL, 6, 4, 12, TO_DATE('02/06/2023', 'DD/MM/YYYY'),
'Card', 'Completed', 1);
INSERT INTO PURCHASE VALUES (SEQ_PURCHASE.NEXTVAL, 7, 5, 13, TO_DATE('10/06/2023', 'DD/MM/YYYY'),
'Cash', 'Completed', 2);
INSERT INTO PURCHASE VALUES (SEQ_PURCHASE.NEXTVAL, 8, 2, 15, TO_DATE('18/06/2023', 'DD/MM/YYYY'),
'Card', 'Pending', 1);
INSERT INTO PURCHASE VALUES (SEQ_PURCHASE.NEXTVAL, 4, 3, 11, TO_DATE('25/06/2023', 'DD/MM/YYYY'),
'Cash', 'Completed', 1);
```

COMMIT;

10.14 SONG

```
CREATE TABLE SONG (
    id_song NUMBER(3,0) CONSTRAINT pk_song PRIMARY KEY,
    id_artist NUMBER(3,0),
    id_product NUMBER(3,0),
    title VARCHAR2(100),
    length VARCHAR2(100),
    CONSTRAINT fk_song_artist FOREIGN KEY (id_artist) REFERENCES ARTIST(id_artist),
    CONSTRAINT fk_song_product FOREIGN KEY (id_product) REFERENCES PRODUCT(id_product)
);
```

```
CREATE SEQUENCE SEQ_SONG
    INCREMENT BY 1
    START WITH 1
    MAXVALUE 1000
    NOCYCLE;
```

```
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 1, 1, 'Self Care', '5:01');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 1, 1, 'Come back to earth', '3:39');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 2, 2, 'luther', '4:15');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 2, 2, 'heart pt. 6', '5:10');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 3, 3, 'Taste', '4:54');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 3, 3, 'Please Please Please', '5:57');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 2, 4, 'HUMBLE', '6:22');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 2, 2, 'DNA', '7:06');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 4, 11, 'the one', '3:45');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 4, 5, 'the lakes', '4:20');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 7, 12, 'Nikes', '5:14');
```

```
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 7, 12, 'Ivy', '4:09');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 2, 13, 'LOYALTY', '3:47');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 2, 13, 'LOVE', '3:33');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 9, 15, 'Blinding Lights', '3:20');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 9, 15, 'Save Your Tears', '3:35');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 8, 16, 'Happier Than Ever', '4:58');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 8, 16, 'NDA', '3:15');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 10, 3, 'Kill Bill', '2:33');
INSERT INTO SONG VALUES (SEQ_SONG.NEXTVAL, 10, 3, 'Snooze', '3:21');

COMMIT;
```