**2.Relation Schemas**

**Relational Model:**

Publisher(publisher\_id, publisher\_name, country, city)

**Functional Dependencies:**

publisher\_id 🡪 publisher\_name, country, city

publisher\_name 🡪 publisher\_id, country, city

**Candidate Keys:**

{(publisher\_id), (publisher\_name)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Publisher (

publisher\_id **INT** **NOT NULL AUTO\_INCREMENT**,

publisher\_name**VARCHAR(45) NOT NULL**,

country **VARCHAR(45) NOT NULL**,

city **VARCHAR(45) NOT NULL**,

**PRIMARY KEY**(publisher\_id),

**UNIQUE**(publisher\_name) )

**Album**

**Relational Model:**

Album(album\_id, album\_name, picture, album\_type, published\_date, publisher\_id)

Foreign Key: publisher\_id to Publisher.

**Functional Dependencies:**

album\_id 🡪 album\_name, picture, album\_type, published\_date, publisher\_id

**Candidate Keys:**

{(album\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Album (

album\_id **INT** **NOT NULL AUTO\_INCREMENT**,

album\_name **VARCHAR(45) NOT NULL**,

picture\_path **VARCHAR(1024)**,

album\_type **ENUM** (‘single’, ‘album’),

published\_date **DATE NOT NULL**,

publisher\_id **INT** **NOT NULL,**

**PRIMARY KEY**(album\_id),

**FOREIGN KEY**(publisher\_id) **REFERENCES** Publisher(publisher\_id) )

**Artist**

**Relational Model:**

Album(artist\_id, artist\_name, description, picture)

**Functional Dependencies:**

artist\_id 🡪 artist\_name, description, picture

**Candidate Keys:**

{(artist\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Artist (

artist\_id **INT** **NOT NULL AUTO\_INCREMENT**,

artist\_name **VARCHAR(45) NOT NULL**,

description **VARCHAR(2048)**,

picture\_path **VARCHAR(1024)**,

**PRIMARY KEY**(artist\_id) )

**Track**

**Relational Model:**

Album(track\_id, track\_name, recording\_type, duration , danceability, acousticness, instrumentalness, speechness, balance, loudness, language, price, date\_of\_addition, album\_id)

Foreign Key: album\_id to Album.

**Functional Dependencies:**

track\_id 🡪 track\_name, recording\_type, duration , danceability, acousticness, instrumentalness, speechness, balance, loudness, language, price, date\_of\_addition, album\_id

**Candidate Keys:**

{(track\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Track (

track\_id **INT** **NOT NULL AUTO\_INCREMENT**,

track\_name **VARCHAR(45) NOT NULL**,

recording\_type **ENUM** (‘live, ‘studio’),

duration **TIME NOT NULL**,

danceability **DOUBLE**(64,63),

acousticness **DOUBLE**(64,63)**,**

instrumentalness **DOUBLE**(64,63),

speechness **DOUBLE**(64,63),

balance **DOUBLE**(64,63),

loudness **DOUBLE**(64,63),

language **VARCHAR**(45),

price **DOUBLE** (64, 32) **NOT NULL**,

date\_of\_addition **DATE NOT NULL**,

album\_id **INT NOT NULL**,

**PRIMARY KEY**(track\_id),

**FOREIGN KEY**(album\_id) **REFERENCES** Album(album\_id) )

**Artist**

**Relational Model:**

Album(artist\_id, artist\_name, description, picture)

**Functional Dependencies:**

artist\_id 🡪 artist\_name, description, picture

**Candidate Keys:**

{(artist\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Artist (

artist\_id **INT** **NOT NULL AUTO\_INCREMENT**,

artist\_name **VARCHAR(45) NOT NULL**,

description **VARCHAR(2048)**,

picture\_path **VARCHAR(1024)**,

**PRIMARY KEY**(artist\_id) )

**Album\_Belongs\_To\_Artist**

**Relational Model:**

Album\_Belongs\_To\_Artist (artist\_id, album\_id)

Foreign Key: artist\_id to Artist

album\_id to Album

**Functional Dependencies:**

(artist\_id, album\_id) 🡪 (artist\_id, album\_id) (Trivial)

**Candidate Keys:**

{(artist\_id, album\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Album\_Belongs\_To\_Artist (

artist\_id **INT** **NOT NULL**,

album\_id **INT NOT NULL**,

**PRIMARY KEY**(artist\_id, album\_id),

**FOREIGN KEY**(artist\_id) **REFERENCES** Artist,

**FOREIGN** **KEY**(album\_id) **REFERENCES** Album)

**Track\_Belongs\_To\_Artist**

**Relational Model:**

Track \_Belongs\_To\_Artist (track\_id, album\_id)

Foreign Key: track\_id to Artist

album\_id to Album

**Functional Dependencies:**

(track\_id, album\_id) 🡪 (track\_id, album\_id) (Trivial)

**Candidate Keys:**

{(track\_id, album\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Track\_Belongs\_To\_Artist (

track\_id **INT** **NOT NULL**,

album\_id **INT NOT NULL**,

**PRIMARY KEY**(track \_id, album\_id),

**FOREIGN KEY**(track \_id) **REFERENCES** Track,

**FOREIGN** **KEY**(album\_id) **REFERENCES** Album)

**Playlist**

**Relational Model:**

Playlist(playlist\_id, playlist\_name, description, picture, creator\_id)

**Functional Dependencies:**

playlist\_id 🡪 playlist\_name, description, picture, creator\_id

**Candidate Keys:**

{(person\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Playlist (

playlist\_id **INT** **NOT NULL AUTO\_INCREMENT**,

playlist\_name **VARCHAR(45) NOT NULL**,

description **VARCHAR(2048)**,

picture\_path **VARCHAR(1024)**,

creator\_id **INT NOT NULL**,

**PRIMARY KEY**(playlist\_id),

**FOREIGN** **KEY**(creator\_id) **REFERENCES** User(user\_id))

**Added**

**Relational Model:**

Playlist(playlist\_id, track\_id, date)

**Functional Dependencies:**

(playlist\_id, track\_id, date)🡪 (playlist\_id, track\_id, date) (Trivial)

**Candidate Keys:**

{(playlist\_id, track\_id, date)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Added (

playlist\_id **INT** **NOT NULL**,

track\_id **INT** **NOT NULL**,

date **TIMESTAMP NOT NULL,**

**PRIMARY KEY**(playlist\_id, track\_id, date),

**FOREIGN** **KEY**(track\_id) **REFERENCES** Track,

**FOREIGN** **KEY**(playlist\_id) **REFERENCES** Playlist)

**Person**

**Relational Model:**

Person(person\_id, username, fullname, password , email)

**Functional Dependencies:**

person\_id 🡪 username, fullname, password , email

username 🡪 person\_id, fullname, password, email

email 🡪 person\_id, username, fullname, password

**Candidate Keys:**

{(person\_id), (username), (email)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Person (

person\_id **INT** **NOT NULL AUTO\_INCREMENT**,

username **VARCHAR(45) NOT NULL**,

fullname **VARCHAR(45)**,

password **VARCHAR(45)** **NOT NULL**,

email **VARCHAR(128) NOT NULL**,

**PRIMARY KEY**(person\_id),

**UNIQUE**(username),

**UNIQUE**(email) )

**User (extends Person)**

**Relational Model:**

User(user\_id, country, language, picture , date\_of\_registration, birthday, gender, budget)

Foreign Key: user\_id to Person(person\_id).

**Functional Dependencies:**

user\_id 🡪 country, language, picture , date\_of\_registration, birthday, gender, budget

**Candidate Keys:**

{(user\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** User (

user \_id **INT** **NOT NULL**,

country **VARCHAR(45)**,

language **VARCHAR(45)**,

picture\_path **VARCHAR(1024)**,

date\_of\_registration **DATE**,

birthday **DATE**,

gender **ENUM**(‘M’, ‘F’, ‘NB’),

budget **DOUBLE**(64,32),

**PRIMARY KEY**(user \_id),

**FOREIGN KEY**(user\_id) **REFERENCES** Person(person\_id) )

**Admin (extends Person)**

**Relational Model:**

Admin(admin\_id)

Foreign Key: admin\_id to Person(person\_id).

**Functional Dependencies:**

admin\_id 🡪 admin\_id (Trivial)

**Candidate Keys:**

{( admin \_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Admin (

admin \_id **INT** **NOT NULL**,

**PRIMARY KEY**(admin \_id),

**FOREIGN KEY**(admin \_id) **REFERENCES** Person(person\_id) )

**Buys**

**Relational Model:**

Buys(user\_id, track\_id)

Foreign Key: user \_id to User

track\_id to Track.

**Functional Dependencies:**

(user\_id, track\_id)🡪 (user\_id, track\_id) (Trivial)

**Candidate Keys:**

{( user\_id, track\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Buys (

user \_id **INT** **NOT NULL**,

track\_id **INT NOT NULL**,

**PRIMARY KEY**(user\_id, track\_id) ,

**FOREIGN KEY**(user \_id) **REFERENCES** User,

**FOREIGN KEY**(track \_id) **REFERENCES** Track)

**Gift**

**Relational Model:**

Gift(giver\_id, receiver\_id, track\_id)

Foreign Key: giver \_id to User(user\_id).

receiver \_id to User(user\_id).

track\_id to Track.

**Functional Dependencies:**

(giver\_id,receiver\_id, track\_id)🡪 (giver\_id, receiver\_id, track\_id) (Trivial)

**Candidate Keys:**

{( giver\_id,receiver\_id, track\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Gift (

giver \_id **INT** **NOT NULL**,

receiver \_id **INT** **NOT NULL**,

track\_id **INT NOT NULL**,

**PRIMARY KEY**(giver\_id, receiver\_id, track\_id) ,

**FOREIGN KEY**(giver \_id) **REFERENCES** User(user\_id),

**FOREIGN KEY**(receiver \_id) **REFERENCES** User(user\_id),

**FOREIGN KEY**(track \_id) **REFERENCES** Track)

**Listens**

**Relational Model:**

Listens(user\_id, track\_id)

Foreign Key: user \_id to User

track\_id to Track.

**Functional Dependencies:**

(user\_id, track\_id)🡪 (user\_id, track\_id) (Trivial)

**Candidate Keys:**

{( user\_id, track\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Listens (

user \_id **INT** **NOT NULL**,

track\_id **INT NOT NULL**,

date **TIMESTAMP NOT NULL**,

**PRIMARY KEY**(user\_id, track\_id) ,

**FOREIGN KEY**(user \_id) **REFERENCES** User,

**FOREIGN KEY**(track \_id) **REFERENCES** Track)

**Bans**

**Relational Model:**

Bans(user\_id, admin\_id)

Foreign Key: user \_id to User.

admin\_id to Admin.

**Functional Dependencies:**

user\_id🡪 admin\_id

**Candidate Keys:**

{( user\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Bans (

user \_id **INT** **NOT NULL**,

admin\_id **INT NOT NULL**,

**PRIMARY KEY**(user\_id) ,

**FOREIGN KEY**(user \_id) **REFERENCES** User,

**FOREIGN KEY**(admin \_id) **REFERENCES** Admin)

**Follows**

**Relational Model:**

Follows(user\_id, playlist\_id)

Foreign Key: user \_id to User.

playlist\_id to Playlist.

**Functional Dependencies:**

(user\_id, playlist\_id)🡪 (user\_id, playlist\_id) (Trivial)

**Candidate Keys:**

{( user\_id, playlist\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Follows (

user \_id **INT** **NOT NULL**,

playlist\_id **INT NOT NULL**,

**PRIMARY KEY**(user\_id, playlist\_id) ,

**FOREIGN KEY**(user \_id) **REFERENCES** User,

**FOREIGN KEY**(playlist \_id) **REFERENCES** Playlist)

**Rates**

**Relational Model:**

Rates(user\_id, playlist\_id, rate)

Foreign Key: user \_id to User.

playlist\_id to Playlist.

**Functional Dependencies:**

(user\_id, playlist\_id)🡪 rate

**Candidate Keys:**

{( user\_id, playlist\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Rates (

user \_id **INT** **NOT NULL**,

playlist\_id **INT NOT NULL**,

rate **INT NOT NULL,**

**PRIMARY KEY**(user\_id, playlist\_id) ,

**FOREIGN KEY**(user \_id) **REFERENCES** User,

**FOREIGN KEY**(playlist \_id) **REFERENCES** Playlist)

**Comments**

**Relational Model:**

Comments(user\_id, playlist\_id, date, comment)

Foreign Key: user \_id to User.

playlist\_id to Playlist.

**Functional Dependencies:**

(user\_id, playlist\_id, date)🡪 comment

**Candidate Keys:**

{( user\_id, playlist\_id, date)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Comments (

user \_id **INT** **NOT NULL**,

playlist\_id **INT NOT NULL**,

comment **VARHAR(2048) NOT NULL,**

date **TIMESTAMP NOT NULL,**

**PRIMARY KEY**(user\_id, playlist\_id, date) ,

**FOREIGN KEY**(user \_id) **REFERENCES** User,

**FOREIGN KEY**(playlist \_id) **REFERENCES** Playlist)

**Collaborates**

**Relational Model:**

Collaborates(user\_id, playlist\_id)

Foreign Key: user \_id to User.

playlist\_id to Playlist.

**Functional Dependencies:**

(user\_id, playlist\_id)🡪 (user\_id, playlist\_id) (Trivial)

**Candidate Keys:**

{( user\_id, playlist\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Follows (

user \_id **INT** **NOT NULL**,

playlist\_id **INT NOT NULL**,

**PRIMARY KEY**(user\_id, playlist\_id) ,

**FOREIGN KEY**(user \_id) **REFERENCES** User,

**FOREIGN KEY**(playlist \_id) **REFERENCES** Playlist)

**Friendship**

**Relational Model:**

Collaborates(user1\_id, user2\_id)

Foreign Key: user1 \_id to User(user\_id).

user2 \_id to User(user\_id).

**Functional Dependencies:**

(user1\_id, user2\_id)🡪 (user1\_id, user2\_id) (Trivial)

**Candidate Keys:**

{(user1\_id, user2\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Friendship (

user1 \_id **INT** **NOT NULL**,

user2\_id **INT NOT NULL**,

**PRIMARY KEY**(user1\_id, user2\_id) ,

**FOREIGN KEY**(user1 \_id) **REFERENCES** User(user\_id),

**FOREIGN KEY**(user2 \_id) **REFERENCES** User(user\_id))

**Blocks**

**Relational Model:**

Blocks(blocker\_id, blocked\_id)

Foreign Key: blocker \_id to User(user\_id).

blocked \_id to User(user\_id).

**Functional Dependencies:**

(blocker \_id, blocked\_id)🡪 (blocker \_id, blocked\_id) (Trivial)

**Candidate Keys:**

{( blocker \_id, blocked\_id)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Blocks (

blocker \_id **INT** **NOT NULL**,

blocked\_id **INT NOT NULL**,

**PRIMARY KEY**(blocker\_id, blocked\_id) ,

**FOREIGN KEY**(blocker \_id) **REFERENCES** User(user\_id),

**FOREIGN KEY**(blocked \_id) **REFERENCES** User(user\_id))

**Sends\_Message**

**Relational Model:**

Sends\_Message(sender\_id, receiver\_id, date, message)

Foreign Key: sender \_id to User(user\_id).

receiver \_id to User(user\_id).

**Functional Dependencies:**

(sender \_id, receiver\_id, date)🡪 message

**Candidate Keys:**

{( sender \_id, receiver\_id, date)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Sends\_Message (

sender \_id **INT** **NOT NULL**,

receiver\_id **INT NOT NULL**,

date **TIMESTAMP NOT NULL**,

message **VARCHAR(2048) NOT NULL**,

**PRIMARY KEY**(sender\_id, receiver\_id, date) ,

**FOREIGN KEY**(sender \_id) **REFERENCES** User(user\_id),

**FOREIGN KEY**(receiver \_id) **REFERENCES** User(user\_id))

**Posts**

**Relational Model:**

Posts(writer\_id, receiver\_id, date, post)

Foreign Key: writer \_id to User(user\_id).

receiver \_id to User(user\_id).

**Functional Dependencies:**

(writer \_id, receiver\_id, date)🡪 post

**Candidate Keys:**

{( writer \_id, receiver\_id, date)}

**Normal Form:**

BCNF

**Table Definition:**

**CREATE TABLE IF NOT EXISTS** Posts (

writer \_id **INT** **NOT NULL**,

receiver\_id **INT NOT NULL**,

date **TIMESTAMP NOT NULL**,

post **VARCHAR(2048) NOT NULL**,

**PRIMARY KEY**(writer\_id, receiver\_id, date) ,

**FOREIGN KEY**(writer \_id) **REFERENCES** User(user\_id),

**FOREIGN KEY**(receiver \_id) **REFERENCES** User(user\_id))