Group 3_8

Play Archive

DDL Script

Group Leader :

Jay Sanghani (202101185)

Group Members :

Dhyey Ladani (202101182)

Jay Sanghani (202101185)

Utsav Maru (202101195)

Krish Rupapara (202101198)

User

```
CREATE TABLE Users (
        user_id int,
        user_name varchar(20) not null,
        email varchar(50) not null,
        date_of_birth date not null,
        password text not null,
        status bool default true,
        date_created date not null,
        wallet_coin float default 0,
        country varchar(30),
        primary key(user_id)
);

CREATE TABLE Transaction_details (
        transaction_id int,
        source_id int,
        recipient_id int not null,
        amount float not null,
        transaction_date date not null,
        payment_method varchar(20) not null,
        primary key(transaction_id),
        foreign key(source_id) references users(user_id)
        on update cascade on delete cascade
);
```

APP

```
CREATE TABLE App(
        app_id int PRIMARY KEY,
        app_name VARCHAR(50) NOT NULL,
        description TEXT NOT NULL,
        app_size FLOAT NOT NULL,
        price FLOAT DEFAULT 0,
        rating FLOAT DEFAULT 0,
        release_date DATE DEFAULT CURRENT_DATE,
        is_active BOOL DEFAULT TRUE,
        age_restrictions INT DEFAULT 0,
        app_version VARCHAR(10) NOT NULL,
        downloads INT DEFAULT 0,
        add_contains BOOL DEFAULT FALSE,
        in_app_purchases BOOL DEFAULT FALSE,
        min_android_ver VARCHAR(10) NOT NULL,
        dev_id INT NOT NULL,
        FOREIGN KEY (dev_id) REFERENCES Developer(dev_id)
        ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Purchased_Apps(
        app_id INT,
        user_id INT,
        installed_date DATE DEFAULT CURRENT_DATE,
        given_rating FLOAT DEFAULT 0,
```

```
    update_available bool DEFAULT FALSE,
    app_version VARCHAR(10),

    PRIMARY KEY(app_id, user_id),
    FOREIGN KEY (app_id) REFERENCES APP(app_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (user_id) REFERENCES Users(user_id)
    ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Developer (
    dev_id int,
    dev_name varchar(30) not null,
    dev_email varchar(50) not null,
    dev_website varchar(50),
    dev_country varchar(50),
    dev_joindate date default current_date,
    dev_revenue float default 0,

    primary key(dev_id)
);

CREATE TABLE Permissions (
    app_id int,
    permissions varchar(20),

    primary key(app_id, permissions),
    foreign key (app_id) references app(app_id)
    on update cascade on delete cascade
```

```
);



CREATE TABLE App_category (
        app_id int,
        app_category varchar(40),

        primary key(app_id, app_category),

        foreign key(app_id) references app(app_id)
        on update cascade on delete cascade

);


CREATE TABLE App_wishlist (
        app_id int,
        user_id int,

        primary key(app_id, user_id),

        foreign key (app_id) references app(app_id)
        on update cascade on delete cascade,

        foreign key (user_id) references users(user_id)
        on update cascade on delete cascade
);
```

Movie

```
create table Movie (
        movie_id int ,
        movie_name varchar(50) not null,
        IMDB int not null,
        description text ,
        movie_length int not null,
        downloads int default 0,
        price float default 0,
        age_restrictions int default 0,
        release_date date not null,
        is_active bool default false,
        max_quality varchar(5),
        studio_id int not null,
        rating float default 0,

        primary key(movie_id),

        foreign key (studio_id) references studio(studio_id)
        on update cascade on delete cascade
);
```

```
CREATE TABLE Movie_wishlist (
        movie_id int,
        user_id int,

        primary key(movie_id, user_id),

        foreign key (movie_id) references movie(movie_id)
        on update cascade on delete cascade,

        foreign key (user_id) references users(user_id)
        on update cascade on delete cascade
);


CREATE TABLE Movie_languages (
        movie_id int,
        movie_languages varchar(50),

        primary key(movie_id,movie_languages),

        foreign key(movie_id) references Movie(movie_id)
        on update cascade on delete cascade
);

CREATE TABLE Movie_cast (
        movie_id int,
        movie_cast varchar(50),

        primary key(movie_id,movie_cast),
```

```
    foreign key(movie_id) references Movie(movie_id)
    on update cascade on delete cascade
);

CREATE TABLE Movie_directors (
    movie_id int,
    movie_director varchar(50),

    primary key(movie_id,movie_director),

    foreign key(movie_id) references Movie(movie_id)
    on update cascade on delete cascade
);

CREATE TABLE Movie_writers (
    movie_id int,
    movie_writer varchar(50),

    primary key(movie_id,movie_writer),

    foreign key(movie_id) references Movie(movie_id)
    on update cascade on delete cascade
);

CREATE TABLE Purchased_Movies(
    movie_id INT,
    user_id INT,
    purchased_date DATE DEFAULT CURRENT_DATE,
```

```
        given_rating FLOAT DEfAULT 0,

        PRIMARY KEY(movie_id, user_id),
        FOREIGN KEY (movie_id) REFERENCES movie(movie_id)
        ON UPDATE CASCADE ON DELETE CASCADE,
        FOREIGN KEY (user_id) REFERENCES Users(user_id)
        ON UPDATE CASCADE ON DELETE CASCADE
);



create table Studio (
        studio_id int PRIMARY KEY,
        studio_name varchar(50) not null,
        studio_email varchar(50) not null,
        studio_country varchar(50),
        studio_join_date DATE DEFAULT CURRENT_DATE,
        studio_revenue float default 0
);

create table Movie_category(
        movie_id int ,
        movie_category Varchar(20),

        primary key(movie_id,movie_category),
        foreign key(movie_id) references Movie(movie_id)
        on update cascade on delete cascade
);
```

Book

```
create table Ebook(

        book_id int,
        book_name varchar(50) not null,
        book_size float not null,
        pages int not null,
        downloads int default 0,
        rating float default 0,
        price float default 0,
        is_audiobook bool default false,
        publish_date date not null,
        description text not null,
        age_restrictions int default 0,
        publisher_id int not null,

        primary key(book_id),
        foreign key(publisher_id) references Publisher(publisher_id)
        on update cascade on delete cascade
);

create table Written_by(

        book_id int,
```

```
    author_id int,

    primary key(book_id, author_id),

    foreign key(book_id) references Ebook(book_id)
    on update cascade on delete cascade,
    foreign key(author_id) references author(author_id)
    on update cascade on delete cascade

);


create table Author(

    author_id int,
    author_name varchar(50) not null,
    author_email varchar(50) not null,
    author_country varchar (50),
    author_join_date date default current_date,

    primary key(author_id)

);

create table Publisher(

    publisher_id int,
    publisher_name varchar(50) not null,
    publisher_email varchar(50) not null,
```

```
	publisher_country varchar (50),
	publisher_revenue float default 0,

	primary key(publisher_id)
);

create table Book_languages(

	book_id int,
	book_languages varchar(50),

	primary key(book_id,book_languages),
	foreign key(book_id) references Ebook(book_id)
	on update cascade on delete cascade
);

create table Book_category(

	book_id int,
	book_category varchar(50),

	primary key(book_id,book_category),
	foreign key(book_id) references Ebook(book_id)
	on update cascade on delete cascade
);

create table Book_wishlist(

	user_id int,
```

```
    book_id int,

    primary key(book_id,user_id),

    foreign key(book_id) references Ebook(book_id)
    on update cascade on delete cascade,

    foreign key(user_id) references Users(user_id)
    on update cascade on delete cascade
);

create table Purchased_books
(
    user_id int,
    book_id int,
    installed_date date default current_date,
    given_rating float default 0,

    foreign key(book_id) references Ebook(book_id)
    on update cascade on delete cascade,

    foreign key(user_id) references Users(user_id)
    on update cascade on delete cascade
);
```