

# Proiect Baze de Date

Tavă Andrei-Daniel 133

Magazin online de produse software.

## 1. Descriere și utilitate:

Proiectul modelează baza de date a unui magazin online de produse software(in principal jocuri video). Produsele sunt dezvoltate și publicate de studiouri pentru una sau mai multe platforme și pot face parte din diverse categorii. Utilizatorii pot cumpăra unul sau mai multe produse într-o tranzacție pentru ei sau cadou pentru alt utilizator. Produsele pot fi supuse reducerilor și pot primi recenzii de la utilizatori.

Baza de date stochează informații despre produse,utilizatori,reduceri,tranzacții și recenzii. Ea poate răspunde la diverse interogări simple,complexe și într-o anumită măsură istorice.

## 2. Reguli și restricții:

- un produs are un singur developer și un singur publisher.
- un produs poate face parte din mai multe categorii
- un produs este disponibil pe mai multe platforme, dar are aceleași attribute(precum preț) indiferent de platformă.
- un utilizator are acces la un produs cumpărat pe toate platformele pe care le deține.
- un utilizator are un singur email și un singur număr de telefon asociat.
- un utilizator poate cumpăra produse pentru alt utilizator(cadou).
- un produs nu poate fi returnat odată cumpărat(sau primit).
- două perioade de reduceri nu se pot suprapune în timp.

### 3-4-5. Entități, relații și attribute:

#### **Entități:**

**Product:** Joc video, conținut adițional sau alte produse software vândute. Cheia primară este una artificială și anume un id numeric.

**Attribute:**

- product\_id#**: cheia primară, numeric.

- product\_title: numele produsului, string, not null.

- release\_date: data de lansare(poate fi în viitor), dată calendaristică, poate fi null, default sysdate.

- base\_price: prețul de bază(fără reduceri) al produsului, numeric, not null, default 0.(Toate prețurile sunt în aceeași monedă)

**Category:** Categorie în care poate fi încadrat un produs. Cheia primară este un id numeric artificială(putea fi și numele, dar este greu de utilizat).

**Attribute:**

- category\_id#**: cheie primară, numeric.

- category\_name: denumirea categoriei, string, not null, unic.

Platform: Platforma(precum PC sau diverse console) pe care produsele pot fi disponibile. Cheia primară este un id numeric artificial.

Attribute:

- platform\_id#: cheie primară, numeric.
- platform\_name: numele platformei, string, not null, unic.
- release\_date: data de lansare(poate fi în viitor), dată calendaristică, poate fi null.

Franchise: Franciza din care poate face parte un produs. Cheia primară este un id numeric artificial.

Attribute:

- franchise\_id#: cheie primară, numeric.
- franchise\_name#: numele francizei, string, not null, unic.

Studio: Compania care dezvoltă/publică produse și deține francize/platforme. Cheia primară este un id numeric artificial.

Attribute:

- company\_id#: cheie primară, numeric.
- company\_name: numele companiei, string, not null, unic.

**Sale:** Perioada de reduceri, în timpul ei mai multe produse pot avea reduceri variate. Cheia primară este data de început, deoarece două astfel de perioade nu se pot suprapune.

**Attribute:**

- start\_date#**: data la care începe perioada, cheie primară, dată calendaristică
- end-date: data la care se încheie perioada, not null, unic, strict mai mare ca start\_date
- sale\_name: numele dat perioadei de reduceri, string, not null.

**Account:** Utilizator al serviciului. Cel care poate achiziționa produse și oferi recenzii. Cheia primară este un id numeric artificial (sunt mai multe chei candidat dar un id numeric facilitează mult operațiile).

**Attribute:**

- account\_id#**: cheie primară, numeric
- username: numele asociat contului, utilizat la login, string, not null, unique.
- display\_name: numele afișat pe profilul utilizatorului, string, not null.
- email: emailul asociat contului, string, not null, unique.
- phone: număr de telefon asociat contului, string, unique.
- join\_date: data la care utilizatorul și-a creat contul, date, not null.

-password\_hash: parola encriptată asociată contului: string, not null.

**Purchase:** O tranzacție efectuată de un utilizator, prin care cumpără pentru el sau pentru al utilizator unul sau mai multe produse. Entitate dependentă a Account. Cheia primară este compusă din id-ul utilizatorului care efectuează tranzacție și data la care a fost făcută.

**Atribute:**

-client\_id#: id-ul utilizatorului care efectuează tranzacția, cheie primară, cheie externă, numeric.

-purchase\_date#: data la care s-a efectuat tranzacția, cheie primară, dată calendaristică.

## **Relații:**

**Published\_by** (Product M(0), Studio 1): face legătura dintre un produs și studioul de către care a fost publicat.

**Developed\_by** (Product M(0), Studio 1): face legătura dintre un produs și studioul de către care a fost dezvoltat.

**Requires** (Product M(0), Product 1(0)): face legătura dintre un produs și produsul de care acesta

depinde (produsul nu poate fi folosit decât dacă utilizatorul deține și dependența sa).

Contains (Category M(1), Product M(1)): face legătura dintre un produs și categoriile care îl conțin.

Part\_of(Product M(0), Franchise 1(0)): face legătura dintre un produs și franciza din care acesta face parte.

Available\_on(Product M(0), Platform M(1)): face legătura dintre un produs și platformele pe care este disponibil.

On\_sale(Product M(1), Sale M(0)): face legătura dintre un produs și perioadele în cadrul cărora a fost redus.

Attribute:

-discount: reducerea(procentuală) la care este supus un produs, numeric, not null, primește valori din (0,1].

Review(Account M(0), Product M(0)): face legătura dintre un produs și utilizatorii care au oferit recenzii. Poate fi văzută și drept o entitate dependentă de Account și Product.

Attribute:

- review\_date: data la care a fost făcută recenzia, dată calendaristică, not null.

- rating: nota oferită de către utilizator pentru produs, numeric, not null, primește valori din [0,5].

Rights\_to(Studio 1(0), Franchise M(0)): face legătura dintre o franciză și studioul care deține drepturile asupra ei.

Parent\_of(Studio 1(0), Studio M(0)): face legătura dintre un studio și toate subsidiarele acestuia.

Owned\_by(Platform M(0), Studio 1(0)): face legătura dintre o platformă și studioul care o deține.

Uses(Account M(0), Platform M(1)): face legătura dintre un utilizator și platformele pe care acesta le utilizează.

Friends\_with(Account M(0), Account M(0)): face legătura dintre un utilizator și prietenii acestuia.

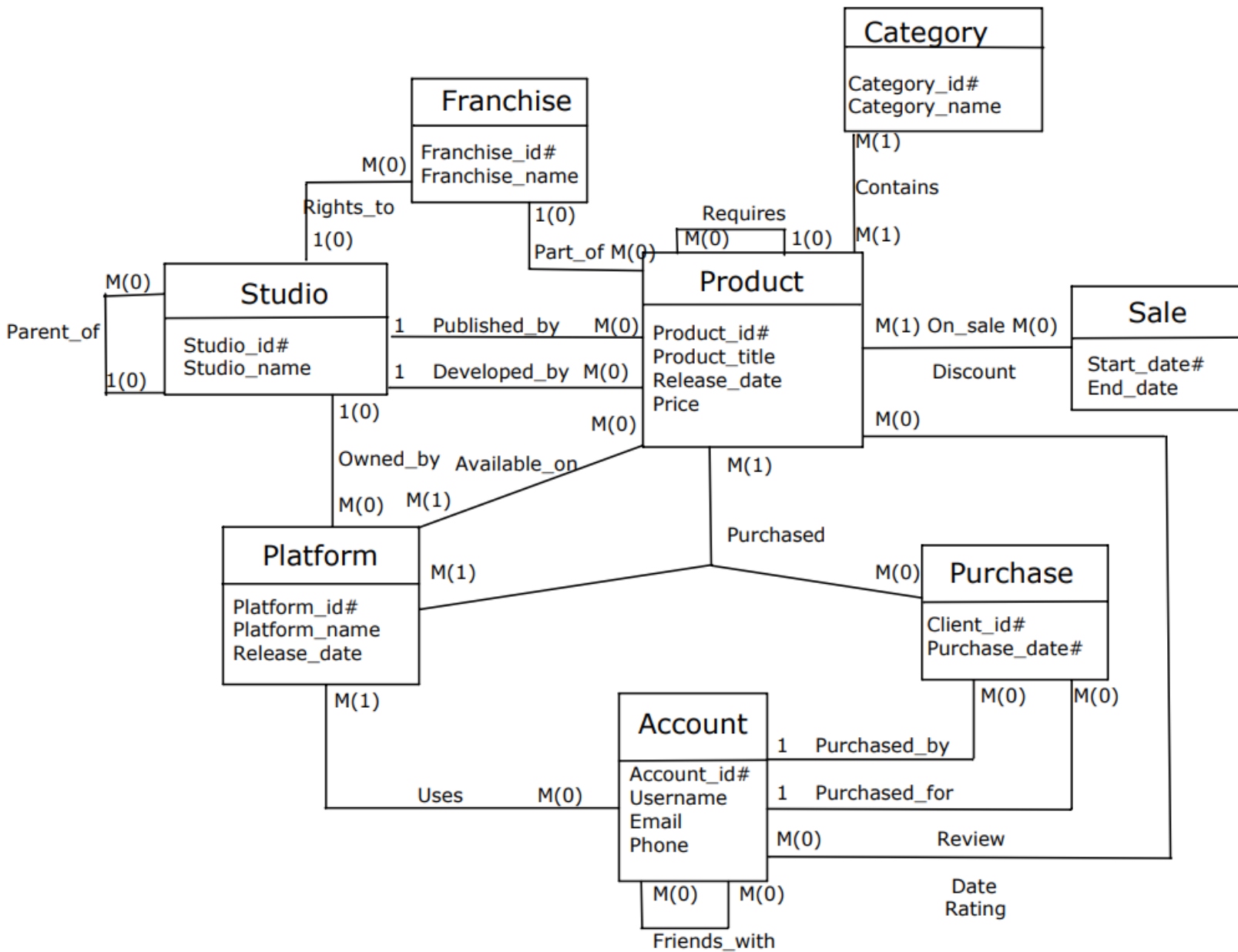


Purchased\_by(Purchase M(0), Account 1):  
face legătura dintre o tranzacție și utilizator care a efectuat-o.

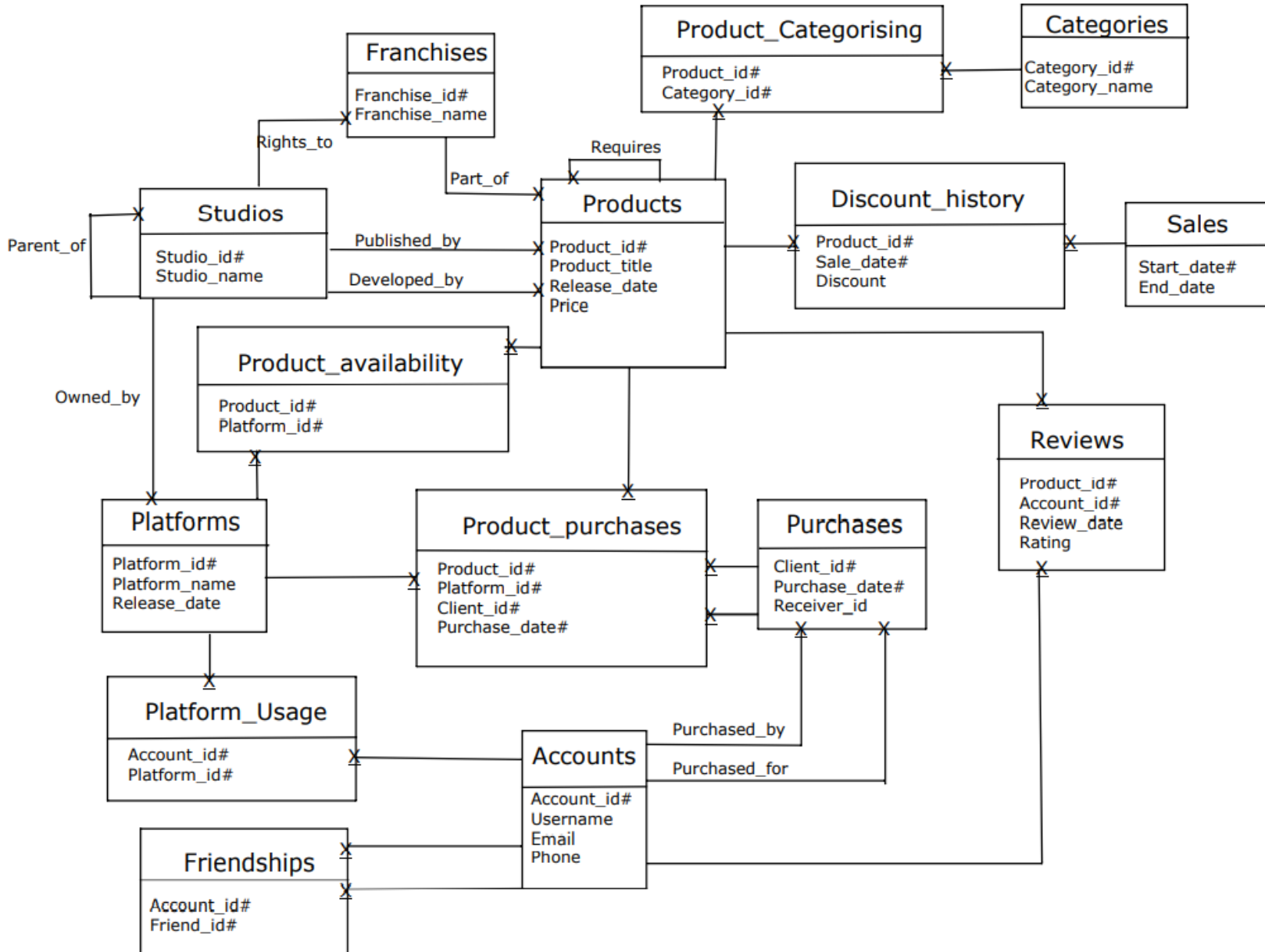
Purchased\_for(Purchase M(0), Account 1):  
face legătura dintre o tranzacție și utilizator care primește produsele.

Purchased(Product M(1), Platform M(1),  
Purchase M(0)): face legătura dintre o tranzacție,  
produsele cumpărate și platformele pentru care au  
fost cumpărate.

## 6.Diagrama E/R:



## 7. Diagrama Conceptuală:



## 8. Schemele Relationale:

Products(**product\_id#**, product\_title, release\_date, base\_price, publisher\_id, developer\_id, franchise\_id, dependency\_id)

Studios(**studio\_id#**, studio\_name, parent\_id)

Franchise(**franchise\_id#**, franchise\_name, holder\_id)

Categories(**category\_id#**, category\_name)

Platforms(**platform\_id#**, platform\_name, release\_date, owner\_id)

Sales(**start\_date#**, end\_date, sale\_name)

Accounts(**account\_id#**, username, display\_name, join\_date, email, phone, password\_hash)

Purchases(**client\_id#**, **purchase\_date#**, receiver\_id)

Reviews(**account\_id#**, **product\_id#**,  
review\_date, rating)

Product\_availability(**product\_id#**,  
**platform\_id#**)

Product\_categorising(**product\_id#**,  
**category\_id#**)

Discount\_history(**product\_id#**,  
**sale\_date#**, discount)

Platform\_usage(**platform\_id#**, **account\_id#**)

Friendships(**account\_id#**, **friend\_id#**)

Product\_purchases(**product\_id#**, **platform\_id**  
**#**, **client\_id#**, **purchase\_date#**)

## 9. Normalizarea FN1-FN3:

FN1: Diagrama este în FN1 deoarece nu există attribute multiple.

Situație nonFN1: Products are atributul multiplu Categories.

Products

Product	Categories
Isaac	Singleplayer
	Roguelike
TF2	Multiplayer
Terraria	Singleplayer
	Sandbox

Soluție: Descompunerea în Products și Product\_categorising(în diagrama, Category este cheie externă)

Products

Product
Isaac
TF2
Terraria

Product\_categorising

Product	Category
Isaac	Singleplayer
Isaac	Roguelike
TF2	Multiplayer
Terraria	Singleplayer
Terraria	Sandbox

FN2: Diagrama este în FN2 deoarece nu există dependențe parțiale.

Situație nonFN2: Price depinde funcțional doar de Product.

Product\_availability

Product	Platform	Price
Rocket	PC	9.99
Turnip	PC	20
Rocket	PS4	9.99
Spire	PS4	20

Soluție: Descompunerea în Product\_availability și Product\_prices (parte din tabela existentă Products din diagrama)

Product\_availability

Product	Platform
Rocket	PC
Turnip	PC
Rocket	PS4
Terraria	PS4

Product\_prices

Product	Price
Rocket	9.99
Turnip	20
Spire	20

FN3: Diagrama este în FN3 deoarece nu există dependențe tranzitive.

Situație nonFN3: Franchise\_owner depinde funcțional de Franchise, atribut non-prim.

### Products

Product	Franchise	Franchise_owner
SMB1	Mario	Nintendo
STH1	Sonic	Sega
SMB3	Mario	Nintendo
STH3&K	Sonic	Sega

Soluție: Descompunerea în Products și Franchises(tabelă existentă în diagramă)

### Products

Product	Franchise
SMB1	Mario
STH1	Sonic
SMB3	Mario
STH3&K	Sonic

### Franchises

Franchise	Franchise_Owner
Mario	Nintendo
Sonic	Sega



## 10-11. Secvențe, Creare și Inserare

```
CREATE SEQUENCE account_index START WITH 0 INCREMENT BY 1 MINVALUE 0
NOCACHE;
CREATE SEQUENCE product_index START WITH 0 INCREMENT BY 1 MINVALUE 0
NOCACHE;
CREATE SEQUENCE category_index START WITH 0 INCREMENT BY 1 MINVALUE 0
NOCACHE;
CREATE SEQUENCE platform_index START WITH 0 INCREMENT BY 1 MINVALUE 0
NOCACHE;
CREATE SEQUENCE studio_index START WITH 0 INCREMENT BY 1 MINVALUE 0
NOCACHE;
CREATE SEQUENCE franchise_index START WITH 0 INCREMENT BY 1 MINVALUE 0
NOCACHE;
/*****
*****/
CREATE TABLE categories (
    category_id    NUMBER(5) DEFAULT category_index.NEXTVAL PRIMARY KEY,
    category_name  VARCHAR2(32) NOT NULL UNIQUE);

CREATE TABLE studios (
    studio_id     NUMBER(5) DEFAULT studio_index.NEXTVAL PRIMARY KEY,
    studio_name   VARCHAR2(64) NOT NULL UNIQUE,
    parent_id    NUMBER(5) REFERENCES studios ( studio_id ) ON DELETE SET
NULL);

CREATE TABLE franchises (
    franchise_id   NUMBER(5) DEFAULT franchise_index.NEXTVAL PRIMARY KEY,
    franchise_name VARCHAR2(64) NOT NULL UNIQUE,
    holder_id     NUMBER(5) REFERENCES studios ( studio_id ) ON DELETE SET
NULL);

CREATE TABLE platforms (
    platform_id   NUMBER(5) DEFAULT platform_index.NEXTVAL PRIMARY KEY,
    platform_name VARCHAR2(32) NOT NULL UNIQUE,
    release_date  DATE DEFAULT sysdate,
    owner_id     NUMBER(5) REFERENCES studios ( studio_id ) ON DELETE SET
NULL);
```

```
CREATE TABLE sales (  
    start_date DATE DEFAULT sysdate PRIMARY KEY,  
    end_date DATE NOT NULL,  
    sale_name VARCHAR2(64) NOT NULL,  
    CONSTRAINT sales_valid CHECK ( end_date > start_date ));
```

```
CREATE TABLE accounts (  
    account_id NUMBER(5) DEFAULT account_index.NEXTVAL PRIMARY KEY,  
    username VARCHAR2(32) NOT NULL UNIQUE,  
    display_name VARCHAR2(64) NOT NULL,  
    join_date DATE DEFAULT sysdate NOT NULL,  
    email VARCHAR2(32) NOT NULL UNIQUE CHECK ( email LIKE '%@%' ),  
    phone VARCHAR2(15) UNIQUE,  
    password_hash CHAR(64) NOT NULL);
```

```
CREATE TABLE products (  
    product_id NUMBER(5) DEFAULT product_index.NEXTVAL PRIMARY KEY,  
    product_title VARCHAR2(64) NOT NULL,  
    release_date DATE DEFAULT sysdate,  
    base_price NUMBER(4, 2) DEFAULT 0 NOT NULL CHECK ( base_price >= 0 ),  
    developer_id NUMBER(5) NOT NULL REFERENCES studios ( studio_id ) ON  
DELETE CASCADE,  
    publisher_id NUMBER(5) NOT NULL REFERENCES studios ( studio_id ) ON  
DELETE CASCADE,  
    franchise_id NUMBER(5) REFERENCES franchises ( franchise_id ) ON DELETE  
SET NULL,  
    dependency_id NUMBER(5) REFERENCES products ( product_id ) ON DELETE  
CASCADE);
```

```
CREATE TABLE purchases (  
    client_id NUMBER(5) REFERENCES accounts ( account_id ) ON DELETE  
CASCADE,  
    purchase_date DATE DEFAULT sysdate,  
    receiver_id NUMBER(5) REFERENCES accounts ( account_id ) ON DELETE SET  
NULL,  
    CONSTRAINT pk_purchases PRIMARY KEY ( client_id, purchase_date ));
```

```
CREATE TABLE reviews (  
    account_id NUMBER(5) REFERENCES accounts ( account_id ) ON DELETE  
CASCADE,  
    product_id NUMBER(5) REFERENCES products ( product_id ) ON DELETE  
CASCADE,  
    review_date DATE DEFAULT sysdate NOT NULL,  
    rating NUMBER(3, 2) NOT NULL CHECK ( rating >= 0 AND rating <= 5 ),  
    CONSTRAINT pk_reviews PRIMARY KEY ( account_id, product_id ));
```

```
CREATE TABLE product_categorising (  
    product_id    NUMBER(5) REFERENCES products ( product_id ) ON DELETE  
    CASCADE,  
    category_id   NUMBER(5) REFERENCES categories ( category_id ) ON DELETE  
    CASCADE,  
    CONSTRAINT pk_prod_cat PRIMARY KEY ( product_id,category_id ));
```

```
CREATE TABLE product_availability (  
    product_id    NUMBER(5) REFERENCES products ( product_id ) ON DELETE  
    CASCADE,  
    platform_id   NUMBER(5) REFERENCES platforms ( platform_id ) ON DELETE  
    CASCADE,  
    CONSTRAINT pk_prod_plat PRIMARY KEY ( product_id,platform_id ));
```

```
CREATE TABLE discount_history (  
    product_id    NUMBER(5) REFERENCES products ( product_id ) ON DELETE  
    CASCADE,  
    sale_date     DATE REFERENCES sales ( start_date ) ON DELETE CASCADE,  
    discount       NUMBER(2, 2) NOT NULL CHECK ( discount > 0 AND discount <= 1  
) ,  
    CONSTRAINT pk_discount PRIMARY KEY ( product_id,sale_date ));
```

```
CREATE TABLE platform_usage (  
    account_id    NUMBER(5) REFERENCES accounts ( account_id ) ON DELETE  
    CASCADE,  
    platform_id   NUMBER(5) REFERENCES platforms ( platform_id ) ON DELETE  
    CASCADE,  
    CONSTRAINT pk_plat_usg PRIMARY KEY ( platform_id,account_id ));
```

```
CREATE TABLE friendships (  
    account_id    NUMBER(5) REFERENCES accounts ( account_id ) ON DELETE  
    CASCADE,  
    friend_id     NUMBER(5) REFERENCES accounts ( account_id ) ON DELETE  
    CASCADE,  
    CONSTRAINT non_self_friend CHECK ( account_id != friend_id ),  
    CONSTRAINT pk_friends PRIMARY KEY ( account_id,friend_id ));
```

```
CREATE TABLE product_purchases (  
    product_id    NUMBER(5) REFERENCES products ( product_id ) ON DELETE  
    CASCADE,  
    platform_id    NUMBER(5) REFERENCES platforms (platform_id) ON DELETE  
    CASCADE,  
    client_id      NUMBER(5),  
    purchase_date  DATE,
```

```
CONSTRAINT fk_prod_purch FOREIGN KEY ( client_id,purchase_date )
REFERENCES purchases ( client_id,purchase_date ) ON DELETE CASCADE,
CONSTRAINT pk_prod_purch PRIMARY KEY (
product_id,platform_id,client_id,purchase_date ));
/*****/
```

```
INSERT INTO accounts
(username,display_name,email,phone,password_hash,join_date)
VALUES ('widderr','wicz','tavaandrei@gmail.com','+40759145680',
'7138f2e1e38c8b5b9e06d4822e083560d4ce717b8c45f571b6768d852193f0d7',
TO_DATE('07/06/2015, 7:27:27 PM', 'MM/DD/YYYY, HH12:MI:SS AM'));
```

```
INSERT INTO accounts
(username,display_name,email,phone,password_hash,join_date)
VALUES
('berkesmcheru','bigboiberke','berkemusellim@hotmail.com','+40757049004',
'3c97be15cc5259a68287081c4b41d7ef0cfea261edc9dcbca2b2357a737c34ca',
TO_DATE('05/07/2020, 5:26:26 PM', 'MM/DD/YYYY, HH12:MI:SS AM'));
```

```
INSERT INTO accounts
(username,display_name,email,phone,password_hash,join_date)
VALUES ('Qmpz','Diaconu','weAre@palmier.com','(void*(0))',
'af5f269ddf697cd26239e7f7e6853e1d3e8fdcd213b9f0ffe825f7725582643f',
TO_DATE('05/07/2020, 5:30:05 PM', 'MM/DD/YYYY, HH12:MI:SS AM'));
```

```
INSERT INTO accounts
(username,display_name,email,phone,password_hash,join_date)
VALUES ('JohnXina','JohnCena','youcantseeme@fbi.mail.us',NULL,
'c83f0be82792393aa49eaae8115931279c0d45259577acea04e50d3b4b7b0344',
TO_DATE('05/07/2020, 5:39:54 PM', 'MM/DD/YYYY, HH12:MI:SS AM'));
```

```
INSERT INTO accounts
(username,display_name,email,phone,password_hash,join_date)
VALUES ('freesciofficial','Sizzle','sizzlefrostindeed@gmail.com',NULL,
'f0cc9b7bf0cb92e5bca6d191a7a4350f17f3ea0d28e0a5e143b347ea560a4434',
TO_DATE('05/09/2015, 5:30:11 PM', 'MM/DD/YYYY, HH12:MI:SS AM'));
```

```
INSERT INTO accounts
(username,display_name,email,phone,password_hash,join_date)
VALUES ('popescu_d017','Decebal
Popescu','decebalpopescu2013@yahoo.ro','074000000',
'05047861e93fb4b8ce12534d7b4eb21020595c45dcf2693bfe906da1b4b20fc5',
TO_DATE('07/10/2021, 2:15:11 PM', 'MM/DD/YYYY, HH12:MI:SS AM'));
```

```
*****/
INSERT INTO categories ( category_name ) VALUES ( 'Roguelike' );
```

```

INSERT INTO categories ( category_name ) VALUES ( 'First Person Shooter' );
INSERT INTO categories ( category_name ) VALUES ( 'Multiplayer' );
INSERT INTO categories ( category_name ) VALUES ( 'Singleplayer' );
INSERT INTO categories ( category_name ) VALUES ( 'Sandbox' );
INSERT INTO categories ( category_name ) VALUES ( '2D' );
INSERT INTO categories ( category_name ) VALUES ( '3D' );
/*****/
INSERT INTO studios (studio_name,parent_id)
VALUES ('Valve',NULL);

INSERT INTO studios (studio_name,parent_id)
VALUES ('Sony Interactive Entertainment',NULL);

INSERT INTO studios (studio_name,parent_id)
VALUES ('Microsoft',NULL);

INSERT INTO studios (studio_name,parent_id)
VALUES ('Nintendo',NULL);

INSERT INTO studios (studio_name,parent_id)
VALUES ('Mojang',2);

INSERT INTO studios (studio_name,parent_id)
VALUES ('Nicalis, Inc',NULL);
/*****/
INSERT INTO platforms (platform_name,owner_id,release_date)
VALUES ('Personal Computer',NULL,NULL);

INSERT INTO platforms (platform_name,owner_id,release_date)
VALUES ('PlayStation 3',1,TO_DATE('23 MAR 2007', 'DD MON YYYY'));

INSERT INTO platforms (platform_name,owner_id,release_date)
VALUES ('PlayStation 4',1,TO_DATE('29 NOV 2013', 'DD MON YYYY'));

INSERT INTO platforms (platform_name,owner_id,release_date)
VALUES ('PlayStation 5',1,TO_DATE('19 NOV 2020', 'DD MON YYYY'));

INSERT INTO platforms (platform_name,owner_id,release_date)
VALUES ('Xbox 360',2,TO_DATE('02 DEC 2005', 'DD MON YYYY'));

INSERT INTO platforms (platform_name,owner_id,release_date)
VALUES ('Xbox One',2,TO_DATE('22 NOV 2013', 'DD MON YYYY'));

INSERT INTO platforms (platform_name,owner_id,release_date)
VALUES ('Xbox Series X',2,TO_DATE('10 NOV 2020', 'DD MON YYYY'));

```

```
INSERT INTO platforms (platform_name,owner_id,release_date)
VALUES ('Nintendo Switch',3,TO_DATE('03 MAR 2017', 'DD MON YYYY'));
/*****/
```

```
INSERT INTO franchises (franchise_name,holder_id)
VALUES ('Team Fortress',0);
```

```
INSERT INTO franchises (franchise_name,holder_id)
VALUES ('Binding of Isaac',5);
```

```
INSERT INTO franchises (franchise_name,holder_id)
VALUES ('Pokemon',3);
```

```
INSERT INTO franchises (franchise_name,holder_id)
VALUES ('Alice in Wonderland',NULL);
```

```
INSERT INTO franchises (franchise_name,holder_id)
VALUES ('Age of Empires',2);
```

```
/*****/
INSERT INTO products
(product_title,release_date,base_price,publisher_id,developer_id,franchise_id,depe
ndency_id)
VALUES ('Team Fortress 2',TO_DATE('10 OCT 2007', 'DD MON
YYYY'),0,0,0,0,NULL);
```

```
INSERT INTO products
(product_title,release_date,base_price,publisher_id,developer_id,franchise_id,depe
ndency_id)
VALUES ('Minecraft',TO_DATE('17 MAY 2009', 'DD MON
YYYY'),23.95,4,4,NULL,NULL);
```

```
INSERT INTO products
(product_title,release_date,base_price,publisher_id,developer_id,franchise_id,depe
ndency_id)
VALUES ('The Binding of Isaac: Rebirth',TO_DATE('04 NOV 2014', 'DD MON
YYYY'),14.99,5,5,1,NULL);
```

```
INSERT INTO products
(product_title,release_date,base_price,publisher_id,developer_id,franchise_id,depe
ndency_id)
VALUES ('The Binding of Isaac: Afterbirth',TO_DATE('30 OCT 2015', 'DD MON
YYYY'),10.99,5,5,1,2);
```

```
INSERT INTO products
(product_title,release_date,base_price,publisher_id,developer_id,franchise_id,dependency_id)
VALUES ('The Binding of Isaac: Afterbirth+',TO_DATE('03 JAN 2017', 'DD MON
YYYY'),9.99,5,5,1,3);
```

```
INSERT INTO products
(product_title,release_date,base_price,publisher_id,developer_id,franchise_id,dependency_id)
VALUES ('The Binding of Isaac: Repentance',TO_DATE('31 MAR 2021', 'DD MON
YYYY'),14.59,5,5,1,4);
```

```
INSERT INTO products
(product_title,release_date,base_price,publisher_id,developer_id,franchise_id,dependency_id)
VALUES ('Pokemon Sword and Shield',TO_DATE('15 NOV 2021', 'DD MON
YYYY'),30,3,3,2,NULL);
```

```
/*****
```

```
INSERT INTO product_categorising VALUES (0,1);
INSERT INTO product_categorising VALUES (0,2);
INSERT INTO product_categorising VALUES (0,6);
INSERT INTO product_categorising VALUES (1,2);
INSERT INTO product_categorising VALUES (1,3);
INSERT INTO product_categorising VALUES (1,4);
INSERT INTO product_categorising VALUES (1,6);
INSERT INTO product_categorising VALUES (2,0);
INSERT INTO product_categorising VALUES (2,3);
INSERT INTO product_categorising VALUES (2,5);
INSERT INTO product_categorising VALUES (3,0);
INSERT INTO product_categorising VALUES (3,3);
INSERT INTO product_categorising VALUES (3,5);
INSERT INTO product_categorising VALUES (4,0);
INSERT INTO product_categorising VALUES (4,3);
INSERT INTO product_categorising VALUES (4,5);
INSERT INTO product_categorising VALUES (5,0);
INSERT INTO product_categorising VALUES (5,3);
INSERT INTO product_categorising VALUES (5,5);
INSERT INTO product_categorising VALUES (6,3);
INSERT INTO product_categorising VALUES (6,6);
```

```
*****/
```

```
INSERT INTO product_availability VALUES (0,0);
INSERT INTO product_availability VALUES (0,1);
INSERT INTO product_availability VALUES (0,4);
INSERT INTO product_availability VALUES (1,0);
INSERT INTO product_availability VALUES (1,1);
```

```

INSERT INTO product_availability VALUES (1,2);
INSERT INTO product_availability VALUES (1,3);
INSERT INTO product_availability VALUES (1,4);
INSERT INTO product_availability VALUES (1,5);
INSERT INTO product_availability VALUES (1,6);
INSERT INTO product_availability VALUES (1,7);
INSERT INTO product_availability VALUES (2,0);
INSERT INTO product_availability VALUES (2,2);
INSERT INTO product_availability VALUES (2,5);
INSERT INTO product_availability VALUES (3,0);
INSERT INTO product_availability VALUES (3,2);
INSERT INTO product_availability VALUES (3,5);
INSERT INTO product_availability VALUES (4,0);
INSERT INTO product_availability VALUES (4,2);
INSERT INTO product_availability VALUES (4,5);
INSERT INTO product_availability VALUES (5,0);
INSERT INTO product_availability VALUES (5,2);
INSERT INTO product_availability VALUES (5,5);
INSERT INTO product_availability VALUES (6,7);
/*****
*****/
INSERT INTO sales VALUES (TO_DATE('01 AUG 2018', 'DD MON
YYYY'),TO_DATE('02 AUG 2018', 'DD MON YYYY'),'Lightning Sale');
INSERT INTO sales VALUES (TO_DATE('28 DEC 2019', 'DD MON
YYYY'),TO_DATE('07 JAN 2020', 'DD MON YYYY'),'Winter Sale 2020');
INSERT INTO sales VALUES (TO_DATE('31 MAR 2021', 'DD MON
YYYY'),TO_DATE('07 APR 2021', 'DD MON YYYY'),'Roguelike Sale 2021');
INSERT INTO sales VALUES (TO_DATE('15 NOV 2021', 'DD MON
YYYY'),TO_DATE('20 NOV 2021', 'DD MON YYYY'),'Nintendo Handheld Sale');
INSERT INTO sales VALUES (TO_DATE('19 JUN 2022', 'DD MON YYYY'),TO_DATE('30
JUN 2022', 'DD MON YYYY'),'Summer Sale 2022');
/*****
*****/
INSERT INTO discount_history VALUES (1,TO_DATE('01 AUG 2018', 'DD MON
YYYY'),0.3);
INSERT INTO discount_history VALUES (1,TO_DATE('28 DEC 2019', 'DD MON
YYYY'),0.15);
INSERT INTO discount_history VALUES (2,TO_DATE('28 DEC 2019', 'DD MON
YYYY'),0.2);
INSERT INTO discount_history VALUES (3,TO_DATE('28 DEC 2019', 'DD MON
YYYY'),0.25);
INSERT INTO discount_history VALUES (4,TO_DATE('28 DEC 2019', 'DD MON
YYYY'),0.1);
INSERT INTO discount_history VALUES (2,TO_DATE('31 MAR 2021', 'DD MON
YYYY'),0.5);

```



```

INSERT INTO discount_history VALUES (3,TO_DATE('31 MAR 2021', 'DD MON
YYYY'),0.5);
INSERT INTO discount_history VALUES (4,TO_DATE('31 MAR 2021', 'DD MON
YYYY'),0.5);
INSERT INTO discount_history VALUES (5,TO_DATE('31 MAR 2021', 'DD MON
YYYY'),0.5);
INSERT INTO discount_history VALUES (6,TO_DATE('15 NOV 2021', 'DD MON
YYYY'),0.5);
INSERT INTO discount_history VALUES (1,TO_DATE('19 JUN 2022', 'DD MON
YYYY'),0.1);
INSERT INTO discount_history VALUES (2,TO_DATE('19 JUN 2022', 'DD MON
YYYY'),0.2);
/*****
*****/
INSERT INTO reviews VALUES (0,0,TO_DATE('08/12/2015', 'DD/MM/YYYY'),4.7);
INSERT INTO reviews VALUES (0,1,TO_DATE('07/07/2018', 'DD/MM/YYYY'),5);
INSERT INTO reviews VALUES (0,5,TO_DATE('15/09/2021', 'DD/MM/YYYY'),4);
INSERT INTO reviews VALUES (1,1,TO_DATE('08/07/2018', 'DD/MM/YYYY'),5);
INSERT INTO reviews VALUES (1,6,TO_DATE('17/11/2021', 'DD/MM/YYYY'),3);
INSERT INTO reviews VALUES (3,1,TO_DATE('10/07/2018', 'DD/MM/YYYY'),5);
INSERT INTO reviews VALUES (4,1,TO_DATE('11/07/2018', 'DD/MM/YYYY'),5);
INSERT INTO reviews VALUES (5,1,TO_DATE('12/07/2018', 'DD/MM/YYYY'),5);
INSERT INTO reviews VALUES (5,2,TO_DATE('21/03/2022', 'DD/MM/YYYY'),1.13);
/*****
*****/
INSERT INTO purchases (client_id,receiver_id,purchase_date)
VALUES (0,0,TO_DATE('08/07/2015', 'DD/MM/YYYY'));

INSERT INTO purchases (client_id,receiver_id,purchase_date)
VALUES (0,0,TO_DATE('02/04/2021', 'DD/MM/YYYY'));

INSERT INTO purchases (client_id,receiver_id,purchase_date)
VALUES (1,1,TO_DATE('01/08/2018', 'DD/MM/YYYY'));

INSERT INTO purchases (client_id,receiver_id,purchase_date)
VALUES (1,1,TO_DATE('16/11/2021', 'DD/MM/YYYY'));

INSERT INTO purchases (client_id,receiver_id,purchase_date)
VALUES (3,3,TO_DATE('03/08/2018', 'DD/MM/YYYY'));

INSERT INTO purchases (client_id,receiver_id,purchase_date)
VALUES (4,4,TO_DATE('07/07/2015', 'DD/MM/YYYY'));

INSERT INTO purchases (client_id,receiver_id,purchase_date)
VALUES (5,0,TO_DATE('19/11/2017', 'DD/MM/YYYY'));
/*****
*****/

```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (0,TO_DATE('08/07/2015', 'DD/MM/YYYY'),0,0);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (0,TO_DATE('02/04/2021', 'DD/MM/YYYY'),2,0);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (0,TO_DATE('02/04/2021', 'DD/MM/YYYY'),2,2);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (0,TO_DATE('02/04/2021', 'DD/MM/YYYY'),3,0);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (0,TO_DATE('02/04/2021', 'DD/MM/YYYY'),3,2);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (0,TO_DATE('02/04/2021', 'DD/MM/YYYY'),4,0);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (0,TO_DATE('02/04/2021', 'DD/MM/YYYY'),4,2);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (0,TO_DATE('02/04/2021', 'DD/MM/YYYY'),5,0);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (0,TO_DATE('02/04/2021', 'DD/MM/YYYY'),5,2);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (1,TO_DATE('01/08/2018', 'DD/MM/YYYY'),1,0);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (1,TO_DATE('01/08/2018', 'DD/MM/YYYY'),1,7);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (1,TO_DATE('16/11/2021', 'DD/MM/YYYY'),6,7);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (3,TO_DATE('03/08/2018', 'DD/MM/YYYY'),1,0);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (4,TO_DATE('07/07/2015', 'DD/MM/YYYY'),0,0);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (5,TO_DATE('19/11/2017', 'DD/MM/YYYY'),1,0);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (5,TO_DATE('19/11/2017', 'DD/MM/YYYY'),1,2);
```

```
INSERT INTO product_purchases (client_id,purchase_date,product_id,platform_id)
VALUES (5,TO_DATE('19/11/2017', 'DD/MM/YYYY'),1,3);
```

```
/******
```

```
INSERT INTO platform_usage VALUES (0,0);
```

```
INSERT INTO platform_usage VALUES (0,2);
```

```
INSERT INTO platform_usage VALUES (0,3);
```

```
INSERT INTO platform_usage VALUES (1,0);
```

```
INSERT INTO platform_usage VALUES (1,7);
```

```
INSERT INTO platform_usage VALUES (2,0);
```

```
INSERT INTO platform_usage VALUES (3,0);
```

```
INSERT INTO platform_usage VALUES (4,0);
```

```
INSERT INTO platform_usage VALUES (5,0);
```

```
INSERT INTO platform_usage VALUES (5,2);
```

```
INSERT INTO platform_usage VALUES (5,3);
```

```
/******
```

```
INSERT INTO friendships VALUES (0,1);
```

```
INSERT INTO friendships VALUES (0,2);
```

```
INSERT INTO friendships VALUES (0,3);
```

```
INSERT INTO friendships VALUES (0,4);
```

```
INSERT INTO friendships VALUES (0,5);
```

```
INSERT INTO friendships VALUES (1,2);
```

```
INSERT INTO friendships VALUES (1,3);
```

```
INSERT INTO friendships VALUES (2,5);
```

```
/******
```

```
COMMIT;
```

```
15 | SELECT * FROM products;
```

Script Output x Query Result x								
All Rows Fetched: 7 in 0.002 seconds								
	PRODUCT_ID	PRODUCT_TITLE	RELEASE_DATE	BASE_PRICE	DEVELOPER_ID	PUBLISHER_ID	FRANCHISE_ID	DEPENDENCY_ID
1	0	Team Fortress 2	10-OCT-07	0	0	0	0	(null)
2	1	Minecraft	17-MAY-09	23.95	4	4	(null)	(null)
3	2	The Binding of Isaac: Rebirth	04-NOV-14	14.99	5	5	1	(null)
4	3	The Binding of Isaac: Afterbirth	30-OCT-15	10.99	5	5	1	2
5	4	The Binding of Isaac: Afterbirth+	03-JAN-17	9.99	5	5	1	3
6	5	The Binding of Isaac: Repentance	31-MAR-21	14.59	5	5	1	4
7	6	Pokemon Sword and Shield	15-NOV-21	30	3	3	2	(null)

15

SELECT \* FROM accounts;

Script Output x

Query Result x

All Rows Fetched: 6 in 0.002 seconds





	ACCOUNT_ID	USERNAME	DISPLAY_NAME	JOIN_DATE	EMAIL	PHONE	PASSWORD_HASH
1	0	widderr	widz	06-JUL-15	tavaandrei@gmail.com	+40759145680	7138f2e1e38c8b5b9e06d4822e08356
2	1	berkesmcheru	bigboiberke	07-MAY-20	berkemusellim@hotmail.com	+40757049004	3c97be15cc5259a68287081c4b41d7e
3	2	Qmpz	Diaconu	07-MAY-20	weAre@palmier.com	(void* (0))	af5f269ddf697cd26239e7f7e6853e1
4	3	JohnXina	JohnCena	07-MAY-20	youcantseeme@fbi.mail.us	(null)	c83f0be82792393aa49eaae81159312
5	4	freesciofficial	Sizzle	09-MAY-15	sizzlefrostindeed@gmail.com	(null)	f0cc9b7bf0cb92e5bca6d191a7a4350
6	5	popescu d017	Decebal Popescu	10-JUL-21	decebalpopescu2013@yahoo.ro	074000000	05047861e93fb4b8ce12534d7b4eb21

15

SELECT \* FROM platforms;

Script Output x

Query Result x

    SQL

All Rows Fetched: 8 in 0.002 seconds

	PLATFORM_ID	PLATFORM_NAME	RELEASE_DATE	OWNER_ID
1	0	Personal Computer	(null)	(null)
2	1	PlayStation 3	23-MAR-07	1
3	2	PlayStation 4	29-NOV-13	1
4	3	PlayStation 5	19-NOV-20	1
5	4	Xbox 360	02-DEC-05	2
6	5	Xbox One	22-NOV-13	2
7	6	Xbox Series X	10-NOV-20	2
8	7	Nintendo Switch	03-MAR-17	3

15

SELECT \* FROM studios;

Script Output x

Query Result x

 All Rows Fetched: 6 in 0.002 seconds

	STUD...	STUDIO_NAME	PARENT_ID
1	0	Valve	(null)
2	1	Sony Interactive Entertainment	(null)
3	2	Microsoft	(null)
4	3	Nintendo	(null)
5	4	Mojang	2
6	5	Nicalis, Inc	(null)

```
15 SELECT * FROM sales;
```

Script Output x

Query Result x

All Rows Fetched: 5 in 0.003 seconds

	START_DATE	END_DATE	SALE_NAME
1	01-AUG-18	02-AUG-18	Lightning Sale
2	28-DEC-19	07-JAN-20	Winter Sale 2020
3	31-MAR-21	07-APR-21	Roquelike Sale 2021
4	15-NOV-21	20-NOV-21	Nintendo Handheld Sale
5	19-JUN-22	30-JUN-22	Summer Sale 2022

```
15 SELECT * FROM reviews;
```

Query Result x

All Rows Fetched: 9 in 0.037 seconds

	ACCOUNT_ID	PRODUCT_ID	REVIEW_DATE	RATING
1	0	0	08-DEC-15	4.7
2	0	1	07-JUL-18	5
3	0	5	15-SEP-21	4
4	1	1	08-JUL-18	5
5	1	6	17-NOV-21	3
6	3	1	10-JUL-18	5
7	4	1	11-JUL-18	5
8	5	1	12-JUL-18	5
9	5	2	21-MAR-22	1.13

## 12. Cereri SQL:

1) Să se determine numele produsului, numele platformei, data la care a fost achiziționat și prețul de bază pentru toate produsele deținute de utilizatorul care și-a creat contul în Iulie 2015, prima zi de luni.

```
SELECT
    product_title,
    platform_name,
    purchase_date date_added,
    CASE base_price          --case afisare
        WHEN 0 THEN 'Free'
        ELSE to_char(base_price)
    END normal_price
FROM
    products
    JOIN product_purchases USING ( product_id )      --join tabele relevante
    JOIN platforms USING ( platform_id )
    JOIN purchases USING ( client_id,
                           purchase_date )
WHERE
    receiver_id = (SELECT account_id                --determinare utilizatori
                   FROM accounts
                   WHERE trunc(join_date) = next_day(TO_DATE('01/JUL/2015',
'DD/MON/YYYY'), 'Monday'));
```

```
2  /*Sa se afiseze numele produsului, numele platformei,
3  data la care a fost achizitionat si pretul de baza
4  pentru toate produsele detinute de utilizatorul care
5  si-a creat contul in Iulie 2015, prima zi de luni. */
6  SELECT
7      product_title,
8      platform_name,
9      purchase_date date_added,
10     CASE base_price          --case afisare
11         WHEN 0 THEN 'Free'
12         ELSE to_char(base_price)
13         END normal_price
14 FROM
15     products
16     JOIN product_purchases USING ( product_id )      --join tabele relevante
17     JOIN platforms USING ( platform_id )
18     JOIN purchases USING ( client_id,
19                             purchase_date )
20 WHERE
21     receiver_id = (SELECT account_id                  --determinare utilizatori
22                     FROM accounts
23                     WHERE trunc(join_date) = next_day(TO_DATE('01/JUL/2015', 'DD/MON/YYYY'), 'Monday'));
```

Script Output x		Query Result x		All Rows Fetched: 12 in 0.01 seconds	
	PRODUCT_TITLE	PLATFORM_NAME	DATE_ADDED	NORMAL_PRICE	
1	Team Fortress 2	Personal Computer	08-JUL-15	Free	
2	Minecraft	Personal Computer	19-NOV-17	23.95	
3	Minecraft	PlayStation 4	19-NOV-17	23.95	
4	Minecraft	PlayStation 5	19-NOV-17	23.95	
5	The Binding of Isaac: Rebirth	Personal Computer	02-APR-21	14.99	
6	The Binding of Isaac: Rebirth	PlayStation 4	02-APR-21	14.99	
7	The Binding of Isaac: Afterbirth	Personal Computer	02-APR-21	10.99	
8	The Binding of Isaac: Afterbirth	PlayStation 4	02-APR-21	10.99	
9	The Binding of Isaac: Afterbirth+	Personal Computer	02-APR-21	9.99	

2) Pentru toate produsele cu rating mediu  $\geq 3$  să se afișeze titlul, ratingul și numărul de unități vândute și să se facă ordonarea după această valoare și după rating (Produsele fără rating vor primi din oficiu 0).

```
SELECT
    product_title,
    nvl(AVG(rating), 0) rating,    --determinare rating
    units_sold
FROM
    products
    LEFT OUTER JOIN reviews USING ( product_id )
    LEFT OUTER JOIN (SELECT
        product_id,
        COUNT(*) units_sold
        FROM(SELECT
            pp.product_id,
            pp.client_id,          --determinarea unitatilor per produs
            pp.purchase_date
            FROM product_purchases pp
            GROUP BY
                pp.product_id,
                pp.client_id,
                pp.purchase_date)
        GROUP BY
            product_id) USING ( product_id )
GROUP BY
    product_title,
    units_sold
HAVING
    nvl(AVG(rating), 0) >= 3      --filtrarea pentru rating
ORDER BY
    units_sold DESC,
    rating DESC;                --ordonarea
```



```

26 /*Pentru toate produsele cu rating mediu >=3 sa se afiseze titlul, ratingul si
27 numarul de unitati vandute si sa se faca ordonarea dupa aceasta valoare si dupa rating
28 Produsele fara rating vor primi din oficiu 0*/
29
30 SELECT
31     product_title,
32     nvl(AVG(rating), 0) rating,      --determinare rating
33     units_sold
34 FROM
35     products
36     LEFT OUTER JOIN reviews USING ( product_id )
37     LEFT OUTER JOIN (SELECT
38         product_id,
39         COUNT(*) units_sold
40     FROM (SELECT
41         pp.product_id,
42         pp.client_id,          --determinarea unitatilor per produs
43         pp.purchase_date
44     FROM product_purchases pp
45     GROUP BY
46         pp.product_id,
47         pp.client_id,

```

Script Output x Query Result x

SQL All Rows Fetched: 4 in 0.014 seconds

	PRODUCT_TITLE	RATING	UNITS_SOLD
1	Minecraft	5	3
2	Team Fortress 2	4.7	2
3	The Binding of Isaac: Repentance	4	1
4	Pokemon Sword and Shield	3	1

3) Sa se afișeze username și totalul ultimei tranzacții efectuate de utilizatorii cu username scris doar cu litere mici sau din mai puțin de 10 caractere. Datele vor fi ordonate după total, descrescător.

```

WITH
last_tranz AS (      --determinarea ultimelor tranzactii pentru fiecare user
    SELECT
        MAX(purchase_date) purchase_date,
        client_id
    FROM purchases
    WHERE
        client_id IN (SELECT account_id
                        FROM accounts
                        WHERE
                            username = lower(username)
                            OR length(username) < 10)
    GROUP BY
        client_id),
last_purchased AS (
    SELECT *
    FROM product_purchases --determinarea produselor din acele tranzactii
    WHERE
        ( client_id,purchase_date ) IN (SELECT
                                            client_id,
                                            purchase_date
                                        FROM last_tranz))

SELECT
    username,
    SUM(actual_price) total --determinarea totalului
FROM
    (SELECT DISTINCT p.product_title,
        username,
        p.base_price,
        base_price * ( 1 - nvl(d.discount, 0) ) actual_price --calculul pretului
    FROM
        discount_history d
        JOIN sales s          ON ( d.sale_date = s.start_date )
        RIGHT OUTER JOIN last_purchased l ON ( l.product_id =
d.product_id
                                AND l.purchase_date >= s.start_date
                                AND l.purchase_date <= s.end_date )
        JOIN products p      ON ( l.product_id = p.product_id )
        JOIN accounts        ON ( account_id = client_id ))

```

```
GROUP BY
    username
ORDER BY
    total DESC;
```

```
60 /*Sa se afiseze username si totalul ultimei tranzactii efectuate
61 de utilizatoii cu username scris doar cu litere mici sau din mai putin de 10 caractere
62 Datele vor fi ordonate dupa total, descrescator*/
63
64 WITH
65 last_tranz AS (          --determinarea ultimelor tranzactii pentru fiecae user
66     SELECT
67         MAX(purchase_date) purchase_date,
68         client_id
69     FROM purchases
70     WHERE
71         client_id IN (SELECT account_id
72                        FROM accounts
73                        WHERE
74                            username = lower(username)
75                            OR length(username) < 10)
76     GROUP BY
77         client_id),
78 last_purchased AS (
79     SELECT *
80     FROM product_purchases --determinarea produselor cumparate in acele tranzactii
```

Script Output x Query Result x  
All Rows Fetched: 5 in 0.038 seconds

	USERNAME	TOTAL
1	widderr	25.28
2	JohnXina	23.95
3	popescu d017	23.95
4	berkesmcheru	15
5	freesciofficial	0

4) Să se afișeze toate produsele care au fost cumpărate la o zi după ce o reducere din care au făcut parte s-a terminat, împreună cu utilizatorul care le-a cumparat și data cumpararii.

```
SELECT
    a.username,
    product_title,
    purchase_date
FROM
    accounts a                                --joinul relevant
    JOIN purchases      ON ( a.account_id = client_id )
    JOIN product_purchases USING ( client_id,purchase_date )
    JOIN products USING ( product_id )
WHERE
    purchase_date - 1 IN (SELECT end_date --determinarea datelor de
incheiere pentru sales
                           FROM
                               discount_history d
                               JOIN sales ON ( sale_date = start_date )
                           WHERE
                               d.product_id = product_id);
```

```
106
107 /*Sa se afiseze toate produsele care au fost cumparate la o zi
108 dupa ce o reducere din care acestea au facut parte s-a terminat,
109 impreuna cu utilizatorul care le-a cumparat si data cumpararii*/
110
111 SELECT
112     a.username,
113     product_title,
114     purchase_date
115 FROM
116     accounts a                                --joinul relevant
117     JOIN purchases      ON ( a.account_id = client_id )
118     JOIN product_purchases USING ( client_id,purchase_date )
119     JOIN products USING ( product_id )
120 WHERE
121     purchase_date - 1 IN (SELECT end_date --determinarea datelor de incheiere pentru sales
122                             FROM
123                                 discount_history d
124                                 JOIN sales ON ( sale_date = start_date )
125                             WHERE
126                                 d.product_id = product_id);
```

Script Output x Query Result x

All Rows Fetched: 1 in 0.007 seconds

	USERNAME	PRODUCT_TITLE	PURCHASE_DATE
1	JohnXina	Minecraft	03-AUG-18

5) Să se afișeze acronimul platformei (PC,XB360,XB1,XBSX,PS3-5,NS) și studioul de care este deținută pentru toate platformele care au cel puțin un produs dezvoltat de deținătorul platformei sau un subsidiar al acesteia.

[illegible]

```

131
132 /*Sa se afiseze acronimul platformei(PC,XB360,XB1,XBSX,PS3-5,NS) si
133 compania de care este detinuta pentru toate platformele care au cel putin
134 un produs dezvoltat de detinatorul platformei sau un subsidiar
135 al acesteia*/
136
137 SELECT
138     decode(platform_id, 0, 'PC', 1, 'PS3',2, 'PS4', --decode pentru acronim
139                 3, 'PS5', 4,'XB360', 5, 'XB1',
140                 6, 'XBSX',7, 'NS', 'ALTELE') platform,
141     nvl(studio_name, 'Nobody') owner
142 FROM
143     platforms pout
144     LEFT OUTER JOIN studios ON ( owner_id = studio_id )
145 WHERE EXISTS (SELECT 1 --subcerere ce determina existenta unui produs
146               FROM
147                   products
148                   JOIN product_availability USING ( product_id )
149               WHERE
150                   pout.platform_id = platform_id
151               AND developer_id IN (SELECT studio_id --determinarea subsidiarilor

```

Script Output x Query Result x

All Rows Fetched: 4 in 0.018 seconds

	PLATFORM	OWNER
1	XB360	Microsoft
2	XB1	Microsoft
3	XBSX	Microsoft
4	NS	Nintendo

## 13. Actualizări și Suprimări:

1) Să se șteargă toate recenziile efectuate de utilizatori care nu dețin produsul respectiv.

```
DELETE FROM reviews
WHERE
    product_id NOT IN ( SELECT
                        DISTINCT product_id
                        FROM
                            product_purchases --lista produselor recenzorului
                        JOIN purchases USING ( client_id,purchase_date )
                        WHERE
                            receiver_id = account_id);
```

```
160 /*Sa se stearga toate recenziile efectuate de utilizatori care nu detin produsul respectiv*/
161
162 DELETE FROM reviews
163 WHERE
164     product_id NOT IN ( SELECT
165                         DISTINCT product_id
166                         FROM
167                             product_purchases --lista produselor recenzorului
168                         JOIN purchases USING ( client_id,purchase_date )
169                         WHERE
170                             receiver_id = account_id);
171
```

Script Output x Query Result x  
Task completed in 0.016 seconds

3 rows deleted.

2) Să se modifice display\_name adăugându-se '\_darnic' pentru toți utilizatorii care nu dețin niciun produs dar au dăruit produse altora.

```
UPDATE accounts
SET display_name = display_name || '_darnic'
WHERE
    account_id NOT IN (SELECT receiver_id --utilizatorii care nu au cumparat
                        FROM purchases)
    AND account_id IN (SELECT client_id --utilizatorii care nu detin
                       FROM purchases);
```

```
174 /*Sa se modifice display_name adaugandu-se '_darnic' pentru toti
175 utilizatorii care nu detin niciun produs dar au daruit produse altora*/
176
177 UPDATE accounts
178 SET display_name = display_name || '_darnic'
179 WHERE
180     account_id NOT IN (SELECT receiver_id --utilizatorii care nu au cumparat
181                        FROM purchases)
182     AND account_id IN (SELECT client_id --utilizatorii care nu detin
183                        FROM purchases);
```

Script Output x Query Result x  
Task completed in 0.022 seconds

1 row updated.



3) Să se seteze deținătorul la null pentru toate francizele care nu au avut un produs de 5 ani.

```
UPDATE franchises
SET holder_id = NULL
WHERE
    franchise_id NOT IN (SELECT franchise_id
                        FROM products    --determinarea francizelor fara produse
recente
                        GROUP BY franchise_id
                        HAVING MAX(release_date) >= add_months(sysdate, - 12
* 5));
```

```
188 /*Sa se seteze detinatorul la null pentru toate francizele
189 care nu au produse sau nu au avut un produs de 5 ani*/
190
191 UPDATE franchises
192 SET holder_id = NULL
193 WHERE
194     franchise_id NOT IN (SELECT franchise_id
195                         FROM products    --determinarea francizelor fara produse recente
196                         GROUP BY franchise_id
197                         HAVING MAX(release_date) >= add_months(sysdate, - 12 * 5));
198
```

Script Output x Query Result x  
Task completed in 0.019 seconds

3 rows updated.

## 16. Outer și Division:

**Outer Join:** Pentru fiecare franciză să se afișeze deținătorul drepturilor, numărul de produse care aparțin francizei și titlul celei mai recente platforme pe care sunt disponibile produse ale sale.

```
SELECT
    franchise_name,
    nvl(studio_name, 'Public Domain') holder,
    titles,
    nvl(platform_name, 'No Titles') latest_platform
FROM
    (SELECT
        franchise_name,
        studio_name,
        COUNT(DISTINCT product_id) titles, --se numara produsele si se
determina data platformei recente
        MAX(p.release_date) max_release
    FROM
        franchises
        LEFT OUTER JOIN studios ON ( holder_id = studio_id )
        LEFT OUTER JOIN products USING ( franchise_id )
        LEFT OUTER JOIN product_availability USING ( product_id )
        LEFT OUTER JOIN platforms p USING ( platform_id )
    GROUP BY
        franchise_name,
        studio_name)
    LEFT OUTER JOIN platforms ON ( max_release = release_date ); --join
pentru denumirea platformei
```

```

203  /*Pentru fiecare franciza sa se afiseze detinatorul drepturilor,
204  numarul de produse care apartin francizei si titlul celei mai recente platforme
205  pe care sunt disponibile produse ale sale*/
206
207  SELECT
208      franchise_name,
209      nvl(studio_name, 'Public Domain') holder,
210      titles,
211      nvl(platform_name, 'No Titles') latest_platform
212  FROM
213      (SELECT
214          franchise_name,
215          studio_name,
216          COUNT(DISTINCT product_id) titles, --se numara produsele si se determina data platformei recente
217          MAX(p.release_date) max_release
218      FROM
219          franchises
220          LEFT OUTER JOIN studios ON ( holder_id = studio_id )
221          LEFT OUTER JOIN products USING ( franchise_id )
222          LEFT OUTER JOIN product_availability USING ( product_id )

```

Script Output x

Query Result x

All Rows Fetched: 5 in 0.008 seconds

	FRANCHISE_NAME	HOLDER	TITLES	LATEST_PLATFORM
1	Team Fortress	Public Domain	1	PlayStation 3
2	Binding of Isaac	Nicalis, Inc	4	PlayStation 4
3	Pokemon	Nintendo	1	Nintendo Switch
4	Alice in Wonderland	Public Domain	0	No Titles
5	Age of Empires	Public Domain	0	No Titles

## Division:

1) Pentru fiecare utilizator să se afișeze produsele care le are în comun cu toți prietenii săi.

```

SELECT
    username,
    product_title
FROM
    products p
    JOIN product_purchases o ON ( p.product_id = o.product_id )
    JOIN purchases          USING ( client_id,purchase_date )
    JOIN accounts           ON ( account_id = receiver_id )
WHERE
    NOT EXISTS (SELECT friend_id
                FROM friendships
                WHERE account_id = receiver_id
                UNION --determinarea prietenilor
                SELECT account_id
                FROM friendships
                WHERE friend_id = receiver_id

```

MINUS

SELECT receiver\_id

FROM

product\_purchases i --determinarea utilizatorilor care detin

produsul

JOIN purchases USING ( client\_id,purchase\_date )

WHERE

i.product\_id = o.product\_id);

```
230 /*Pentru fiecare utilizator sa se afiseze produsele care le are in comun cu toti prietenii sai*/
231
232 SELECT
233     username,
234     product_title
235 FROM
236     products p
237     JOIN product_purchases o ON ( p.product_id = o.product_id )
238     JOIN purchases          USING ( client_id,purchase_date )
239     JOIN accounts           ON ( account_id = receiver_id )
240 WHERE
241     NOT EXISTS (SELECT friend_id
242                 FROM friendships
243                 WHERE account_id = receiver_id
244                 UNION --determinarea prietenilor
245                     SELECT account_id
246                     FROM friendships
247                     WHERE friend_id = receiver_id
248                 MINUS
249                     SELECT receiver_id
250                     FROM
```

Script Output x Query Result x  
All Rows Fetched: 2 in 0.01 seconds

	USERNAME	PRODUCT_TITLE
1	JohnXina	Minecraft
2	freesciofficial	Team Fortress 2

2) Să se afișeze produsele care fac parte simultan din categoriile 'Singleplayer' și '3D'.

```
SELECT product_title
FROM products p
WHERE
    NOT EXISTS (SELECT category_id
                FROM categories --categoriile respective
                WHERE
                    initcap(category_name) = 'Singleplayer'
                    OR upper(category_name) = '3D'
                MINUS
                SELECT category_id --categoriile produsului
                FROM product_categorising pc
                WHERE pc.product_id = p.product_id);
```

```
256 /*Sa se afiseze produsele care fac parte simultan din categoriile 'Singleplayer' si '3D'*/
257
258 SELECT product_title
259 FROM products p
260 WHERE
261     NOT EXISTS (SELECT category_id
262                 FROM categories --categoriile respective
263                 WHERE
264                     initcap(category_name) = 'Singleplayer'
265                     OR upper(category_name) = '3D'
266                 MINUS
267                 SELECT category_id --categoriile produsului
268                 FROM product_categorising pc
269                 WHERE pc.product_id = p.product_id);
```

Script Output x Query Result x	
All Rows Fetched: 2 in 0.004 seconds	
PRODUCT_TITLE	
1 Minecraft	
2 Pokemon Sword and Shield	

## 17. Optimizare Cerere:

Să se afișeze toate produsele reduse în anul 2019 sau care sunt deținute de utilizatori ce folosesc platforma 'PlayStation 4' și au făcut cel puțin un review.

### Inițială

Algebră:

$$\text{Result} = \sigma_{\text{product\_title}}(\pi_{\text{sale\_date}=2019}(\text{Products} \bowtie \text{Discount\_history}) \cup \sigma_{\text{product\_title}}((\text{Products} \bowtie \text{Product\_purchases} \bowtie \text{Purchases} \bowtie (\sigma_{\text{account\_id}}(\pi_{\text{platform\_name}='PlayStation4'}(\text{Platforms} \bowtie \text{Platform\_usage} \bowtie \text{Accounts}))) \cap \sigma_{\text{account\_id}}(\text{Reviews} \bowtie \text{Accounts})))$$

### SQL:

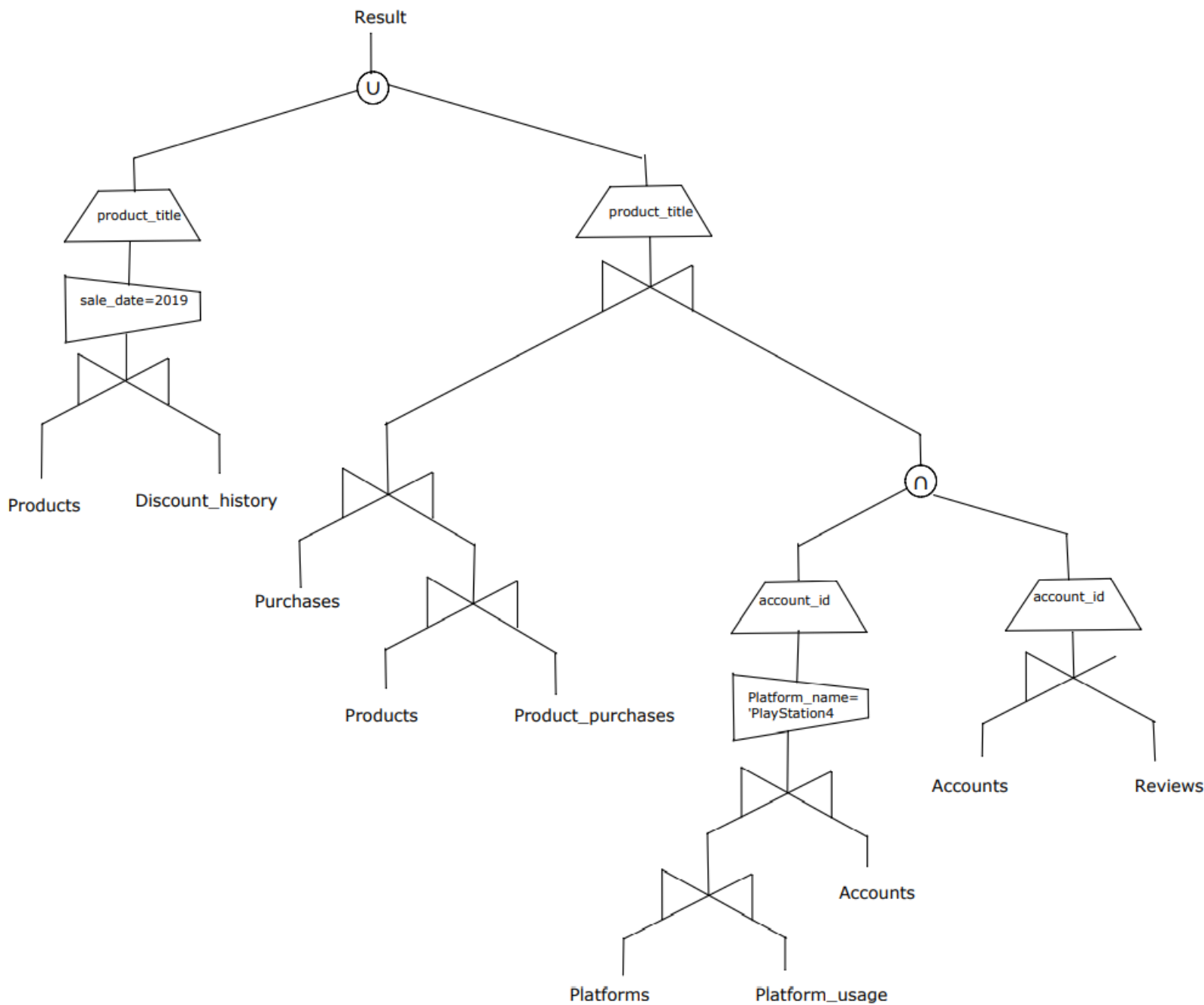
```
SELECT product_title
FROM
  products
  JOIN discount_history USING ( product_id )
WHERE
  to_char(sale_date, 'YYYY') = '2019'
UNION
SELECT
  product_title
FROM
  products
  JOIN product_purchases USING ( product_id )
  JOIN purchases USING ( client_id,purchase_date )
  JOIN ( SELECT account_id
        FROM
          platforms
          JOIN platform_usage USING ( platform_id )
          JOIN accounts USING ( account_id )
        WHERE
```

```

        lower(platform_name) = 'playstation 4'
    INTERSECT
        SELECT account_id
    FROM
        reviews
    JOIN accounts USING ( account_id ))
ON ( receiver_id = account_id );

```

Arbore:



# Optimizată

Algebră:

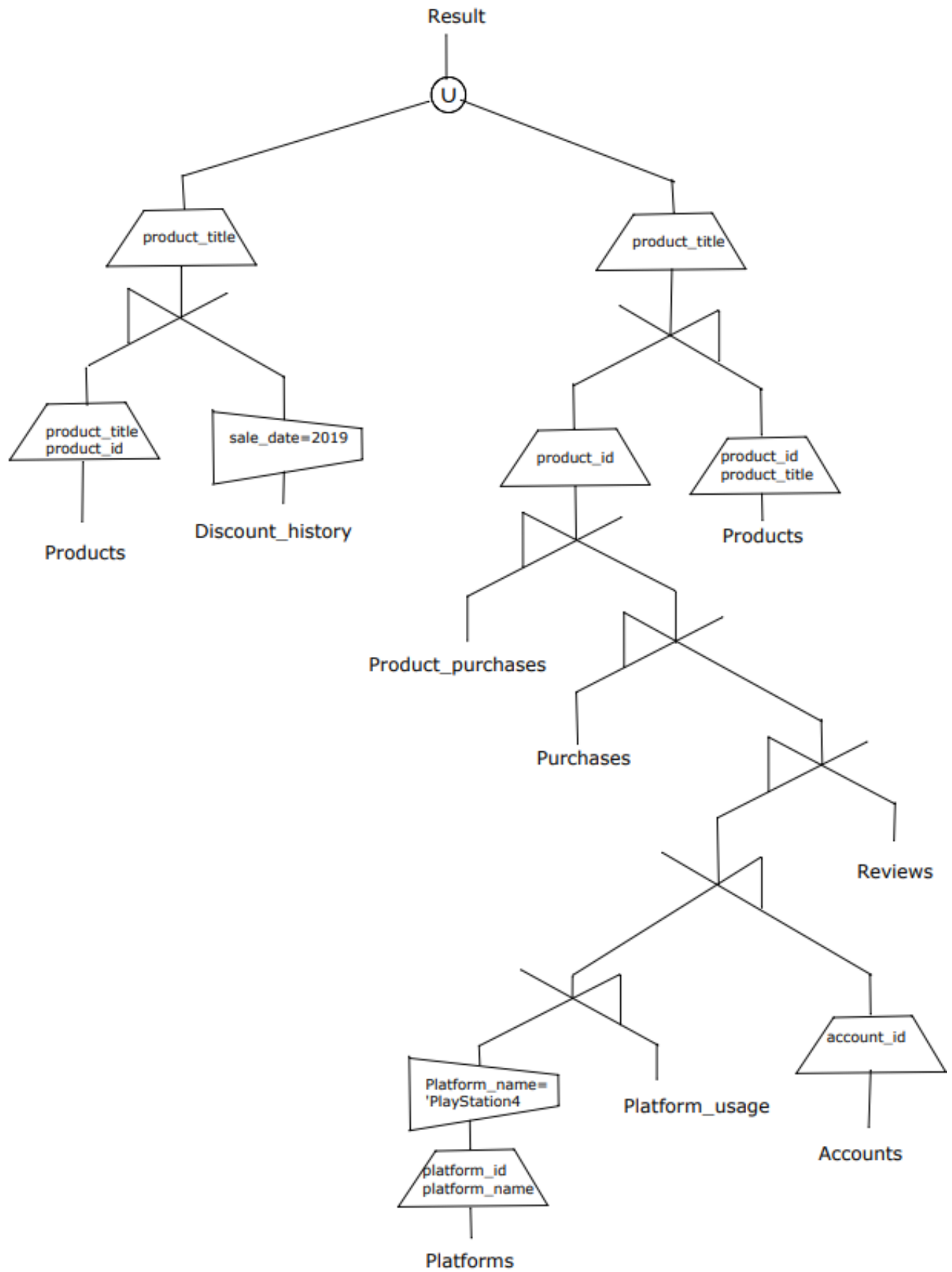
$$\text{Result} = \sigma_{\text{product\_title}}(\sigma_{\text{product\_title,product\_id}}(\text{Products}) \bowtie \pi_{\text{sale\_date}=2019}(\text{Discount\_history}) \cup \\ \sigma_{\text{product\_title}}(\sigma_{\text{product\_id}}(\text{Products}) \bowtie \sigma_{\text{product\_title,product\_id}}(\text{Product\_purchases} \bowtie ( \\ \text{Purchases} \bowtie ((\sigma_{\text{account\_id}}(\text{Accounts}) \bowtie (\text{Platform\_usage} \bowtie \\ \pi_{\text{platform\_name}='PlayStation4'}(\sigma_{\text{platform\_id,platform\_name}}(\text{Platforms})))) \bowtie \text{Reviews}))))))$$

SQL:

```
SELECT product_title
FROM
    products
    JOIN (SELECT product_id
          FROM discount_history
          WHERE to_char(sale_date, 'YYYY') = '2019')
    USING(product_id)
UNION
SELECT product_title
FROM(SELECT account_id
     FROM ( SELECT
             platform_id,
             platform_name
           FROM
             platforms
           WHERE
             lower(platform_name) = 'playstation 4')
     JOIN platform_usage USING ( platform_id )
     JOIN accounts USING ( account_id )
     JOIN reviews USING ( account_id ))
JOIN purchases ON ( receiver_id = account_id)
JOIN product_purchases USING ( client_id,purchase_date )
JOIN products USING ( product_id );
```



Arbore:



## 18a. Normalizarea BCNF-FN5:

BCNF: Diagrama este în BCNF deoarece toți determinanții sunt chei candidat sau includ o cheie candidat (nu există chei candidat care se suprapun)

Situație nonBCNF: Sale\_end depinde funcțional de Sale\_start, iar Sale\_start nu este cheie candidat și nu include o cheie candidat.

### Discount\_history

Product	Sale_start	Sale_end
Minecraft	01-AUG-2017	03-AUG-2017
Terraria	01-AUG-2017	03-AUG-2017
Minecraft	19-NOV-2019	29-NOV-2019
Celeste	19-NOV-2019	29-NOV-2019

Soluție: Descompunerea în Discount\_history și Sales (tabelă existentă în diagrama)

### Discount\_history

Product	Sale_start
Minecraft	01-AUG-2017
Terraria	01-AUG-2017
Minecraft	19-NOV-2019
Celeste	19-NOV-2019

### Sales

Start_date	End_date
01-AUG-2017	03-AUG-2017
19-NOV-2019	29-NOV-2019

FN4: Diagrama este în FN4 deoarece nu există dependențe multiple.

Situație nonFN4: fiecare Product determină o mulțime de Platform, iar un Account lasă o recenzie indiferent de platformă

### Reviews

Product	Platform	Account
TF2	PC	uzzy
TF2	PS3	uzzy
TF2	XB360	uzzer
TF2	PC	acker
TF2	PS3	acker
TF2	XB360	acker
PokeS&S	NS	acker
PokeS&S	NS	tillot

Soluție: Descompunerea în Product\_availability(tabelă existentă în diagrama) și Product\_reviews(parte din tabela existentă Reviews din diagrama)

### Product\_availability

Product	Platform
TF2	PC
TF2	PS3
TF2	XB360
PokeS&S	NS

### Product\_reviews

Product	Account
TF2	uzzy
TF2	acker
PokeS&S	acker
PokeS&S	tillot

FN5: Diagrama nu este în FN5 deoarece tabela Product\_purchases nu este în FN5:

Relația poate fi descompusă deoarece un Account deține un Product pentru toate Platform pe care le folosește dacă Product este disponibil pentru acel Platform.(Tabelă simplificată)

Product\_purchases

Product	Platform	Account
Rocket	PC	uzzy
Rocket	PS4	uzzy
Rocket	XB1	tillot
Hollow	NS	uzzy
Hollow	NS	tillot

Soluție: Descompunerea în Product\_availability(tabelă existentă în diagramă), Platform\_usage(tabelă existentă în diagramă) Și Product\_purchases.

Product\_availability

Product	Platform
Rocket	PC
Rocket	PS4
Rocket	XB1
Hollow	NS

Platform\_Usage

Account	Platform
uzzy	PC
uzzy	PS4
uzzy	NS
tillot	XB1
tillot	NS

Product\_purchases

Account	Product
uzzy	Rocket
uzzy	Hollow
tillot	Rocket
tillot	Hollow

## 18b. Denormalizare:

Denormalizarea se va efectua în principal pe tabelele Product\_purchases și Purchases, deoarece sunt cele mai utilizate în cereri.

Reținerea prețului unui produs la data respectivă în tabela Product\_purchases îmbunătățește enorm cererile corespunzătoare, deși se pierde FN2. De asemenea se poate reține totalul unei tranzacții în Purchases pentru a eficientiza și mai mult cereri legate de sumele tranzacțiilor. Aceste denormalizări nu aduc probleme la modificări de date deoarece prețurile și sumele din trecut nu se pot schimba.