Apostu Alexandru-Mihai
Grupa 212

Gaming Industry Company
- Proiect BD -

1) Descrierea modelului real,a utilității acestuiași a regulilor de funcționare.

Dorim sa creem o baza de date pentru o companie mica din industria IT, specializata pe ramura de gaming. Aceasta companie creeaza jocuri video la 0, le lanseaza sub numele unui publisher si le vinde pe propriul site, in format digital.

Angajatii sunt de 4 tipuri: developeri, testeri, artisti si analisti, fiecare avand rolul sau vital in dezvoltarea companiei.

Clientii pot fi de doua tipuri: persoane fizice sau organizatii (companii), niciunul din ei fiind limitati la cumpararea tipurilor de produse in vreun mod.

Produsele sunt fie jocuri video, de un anumit gen, pentru o anumita categorie de varsta, fie software pentru tinerii pasionati de streaming, cat si pentru developerii care au nevoie de unelte pentru crearea unor modele asemanatoare, dar individuale.

Comenzile pot contine 1 sau mai multe produse in ele, fiecare comanda avand un unic client si o plata efectuata.

Persoanele care le cumpara sunt inregistrate, fiecare avand un username unic, astfel incat fiecare persoana detine o “librarie” de jocuri pe care le va avea permanent in posesie.

Proiectele neterminate, atat jocurile cat si aplicatiile software se retin si ele, cat si angajatii care lucreaza la acestea.

2) *Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.*

- Un angajat trebuie sa fie obligatoriu developer, tester, artist sau analist.
- Fiecare client trebuie sa fie ori persoana, ori companie.
- Fiecare joc trebuie sa aiba un UNIC publisher. Nu poate exista un joc lansat fara publisher.
- Fiecare persoana detine O SINGURA librarie de jocuri video.
- La fiecare proiect trebuie sa lucreze MINIM un developer si MINIM un tester.
- La fiecare joc neterminat trebuie sa lucreze MINIM un artist.
- Fiecare comanda trebuie sa contina MACAR un produs.
- Fiecare comanda trebuie sa aiba neaparat plata efectuata (un payment_id)
- Fiecare software trebuie sa fie ori Streaming Software, ori Dev_Utilities.

3) *Descrierea entităților, incluzând precizarea cheii primare.*

EMPLOYEES :

- PK: employee_id
- Atribute: Numele, Prenumele, Salariul si Data Angajarii.

➔ Developer :

- PK: developer_id
- FK: employee_id
- Atribute: Pozitia curenta (intern, junior, senior) in companie.

➔ Artist :

- PK: artist_id
- FK: employee_id
- Atribute: Ratingul de pe piata al artistului.

➔ Tester :

- PK: tester_id
- FK: employee_id
- Atribute: Experienta ca tester si rata de eroare (sansa ca programul / jocul sa aiba un bug dupa ce iese pe piata si a fost evaluat de respectivul angajat ca fiind OK)

➔ Analyst :

- PK: analyst_id
- FK: employee_id
- Atribute: Rata de success a tuturor modelelor de bussiness propuse de angajat, cat si procentul cu care au crescut veniturile companiei datorita ideilor respectivului angajat.

CLIENTS :

- PK: client_id
- Atribute: Numarul de telefon al clientului si email-ul de contact.

➔ Organisation :

- PK: organisation_id
- FK: client_id
- Atribute: Numele companiei, CUI-ul acesteia, si Valoarea Neta.

➔ Person :

- PK: person_id
- FK: client_id, library_id
- Atribute: Numele si Prenumele individului, Varsta,Username-ul care detine libraria de jocuri / software si fidelitatea (numarul de ani de cand e client).

PRODUCT :

- PK: product_id
- Atribute: Numele produsului, Pretul de vanzare si Platforma pentru care este disponibil

➔ Game :

- PK: game_id
- FK: product_id, publisher_id
- Atribute: genul (categoria) jocului, data lansarii acestuia, tipul (singleplayer / multiplayer), ratingul de varsta, cat si o variabila de tip bool care sa indice daca poate fi jucat in VR sau nu.

➔ Software :

- PK: software_id
- FK: product_id
- Atribute: Numarul Versiunii programului, care indica cea mai noua versiunea lansata a acestuia.

❖ Streaming_Software :

- PK: stream_soft_id
- FK: software_id
- Atribute: Platforma de streaming pentru care este disponibil programul (Twitch, Youtube etc)

❖ Dev_Utilities :

- PK : dev_utils_id
- FK : software_id
- Atribute: Tipul de licenta (personala, profesionala, premium etc)

PUBLISHER :

- PK: publisher_id
- Atribute: Numele firmei sub care se publica jocul, Comisionul pe care aceasta il primeste, cat si un Cont bancar pentru efectuarea tranzactiilor.

ORDERS :

- PK: order_id
- FK: product_id, client_id, payment_id
- Atribute: Data la care s-a efectuat comanda, ip-ul clientului care a cumparat si discountul aplicat, daca acesta exista.

PAYMENT :

- PK: payment_id
- Atribute: Modalitatea de efectuarii a platii (card / paypal / swift / vouchere etc), numarul si data facturii, cat si mentiuni cu privire la aceasta.

PROJECT :

- PK: project_id
- FK: developer_id, tester_id
- Atribute: Numele proiectului la care se lucreaza, fie joc, fie aplicatie, si rata de completare a acesteia (%).

➔ Unfinished_Game :

- PK: unfinished_game_id
- FK: project _id, artist_id
- Atribute: O variabila care sa spuna daca jocul este lansat in early access sau nu si o data de lansare estimata pentru piata.

➔ Unfinished_Soft :

- PK: unfinished_soft_id
- FK: project _id, analyst_id
- Atribute: O variabila de tip care sa spuna in ce faza de development este aplicatia.

➔ Library :

- PK: library_id
- Atribute: Numele librariei.

4) *Descrierea relațiilor, incluzând precizarea cardinalității acestora.*

EMPLOYEES -> DEVELOPER 1(1) : 1(0) [Relatie “IS A”]

EMPLOYEES -> ANALYST 1(1) : 1(0) [Relatie “IS A”]

EMPLOYEES -> ARTIST 1(1) : 1(0) [Relatie “IS A”]

EMPLOYEES -> TESTER 1(1) : 1(0) [Relatie “IS A”]
> Fiecare angajat apartine unei singure categorii de functii din cele 4.

CLIENTS -> PERSON 1(1) : 1(0) [Relatie “IS A”]

CLIENTS -> ORGANISATION 1(1) : 1(0) [Relatie “IS A”]
> Fiecare client e fie persoana fizica, fie companie.

PRODUCT -> SOFTWARE 1(1) : 1(0) [Relatie “IS A”]

PRODUCT -> GAMES 1(1) : 1(0) [Relatie “IS A”]
> Fiecare produs e joc sau software.

SOFTWARE -> STREAMING_SOFTWARE 1(1) : 1(0) [Relatie “IS A”]

SOFTWARE -> DEV_UTILS 1(1) : 1(0) [Relatie “IS A”]
> Fiecare software e pentru developeri sau pentru streameri.

PROJECT -> UNFINISHED_GAMES 1(1) : 1(0) [Relatie “IS A”]

PROJECT -> UNFINISHED_SOFT 1(1) : 1(0) [Relatie “IS A”]
> Fiecare proiect in lucru e joc sau aplicatie.

PUBLISHER -> GAMES 1(1) : M(0)
> Un publisher poate lansa mai multe jocuri, iar un joc are obligatoriu un publisher.

LIBRARY -> GAMES M(0) : M(0)
> Un librarie poate contine niciunul, unul sau mai multe jocuri, iar un joc se poate afla in mai multe librarii, sau in niciuna.

PERSON -> LIBRARY 1(1) : 1(1)
> O librarie apartine unei unice persoane si viceversa.

CLIENTS -> ORDER 1(1) : M(0)
> Un client poate avea mai multe comenzi.

PRODUCT -> ORDER M(1) : M(0)
> O comanda poate contine unul (minim 1) sau mai multe produse.

PAYMENT -> ORDER 1(1) : 1(1)
> O comanda are obligatoriu plata efectuata (una singura).

DEVELOPER -> PROJECT M(1) : M(0)
> La un proiect pot lucra mai multi developeri, dar minim unul.

TESTER -> PROJECT M(1) : M(0)
> La un proiect pot lucra mai multi testeri, dar minim unul.

ARTIST -> UNFINISHED_GAMES M(1) : M(0)
> La un joc pot lucra mai multi artisti, dar minim unul.

ANALYST -> UNFINISHED_SOFT 1(1) : M(0)
> La un soft pot lucra mai multi analisti, dar minim unul.

5) *Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.*

* EMPLOYEES

    - employee_id int NOT NULL (Primary Key)
    - first_name varchar(255) NOT NULL
    - last_name varchar(255) NOT NULL
    - salary int NOT NULL
    - hire_date date NOT NULL

* DEVELOPER

    - employee_id int NOT NULL UNIQUE (Foreign Key)
    - developer_id int NOT NULL (Primary Key)
    - position varchar(255) NOT NULL

* ARTIST

    - employee_id int NOT NULL UNIQUE (Foreign Key)
    - artist_id int NOT NULL (Primary Key)
    - rating int NOT NULL

* TESTER

    - employee_id int NOT NULL UNIQUE (Foreign Key)
    - tester_id int NOT NULL (Primary Key)
    - experience int NOT NULL
    - error_rate int NOT NULL

* ANALYST

    - employee_id int NOT NULL UNIQUE (Foreign Key)
    - analyst_id int NOT NULL (Primary Key)
    - net_growth int NOT NULL
    - success_chance int NOT NULL

* CLIENTS

    - client_id int NOT NULL (Primary Key)
    - phone varchar(20) NOT NULL
    - email varchar(255)

* ORGANISATION

    - client_id int NOT NULL UNIQUE (Foreign Key)
    - organisation_id int NOT NULL (Primary Key)
    - name varchar(255) NOT NULL
    - CUI int NOT NULL UNIQUE
    - net_worth int

* LIBRARY

    - library_id int NOT NULL (Primary Key)
    - library_title varchar(255) NOT NULL

* PERSON

    - client_id int NOT NULL UNIQUE (Foreign Key)
    - library_id int NOT NULL UNIQUE (Foreign Key)
    - person_id int NOT NULL (Primary Key)
    - first_name varchar(255) NOT NULL
    - last_name varchar(255) NOT NULL
    - age int NOT NULL
    - fidelity int NOT NULL
    - username varchar(255) NOT NULL

* PRODUCT

    - product_id int NOT NULL (Primary Key)
    - name varchar(255) NOT NULL
    - price int NOT NULL
    - platform varchar(255) NOT NULL

* PUBLISHER

    - publisher_id int NOT NULL (Primary Key)
    - name varchar(255) NOT NULL
    - commission int NOT NULL
    - bank_account varchar(255) NOT NULL

* GAMES

    - product_id int NOT NULL UNIQUE (Foreign Key)
    - publisher_id int NOT NULL UNIQUE (Foreign Key)
    - game_id int NOT NULL (Primary Key)
    - genre varchar(255) NOT NULL
    - release_date date NOT NULL
    - age_rating char NOT NULL
    - VR varchar(3)

* SOFTWARE

    - product_id int NOT NULL UNIQUE (Foreign Key)
    - software_id int NOT NULL (Primary Key)
    - version varchar(20) NOT NULL

* STREAMING_SOFTWARE

    - software_id int NOT NULL UNIQUE (Foreign Key)
    - stream_soft_id int NOT NULL (Primary Key)
    - stream_platform varchar(255) NOT NULL

* DEV_UTILITIES

    - software_id int NOT NULL UNIQUE (Foreign Key)
    - dev_utils_id int NOT NULL (Primary Key)
    - license_type varchar(255) NOT NULL

* PAYMENT

    - payment_id int NOT NULL (Primary Key)
    - payment_type varchar(255) NOT NULL
    - receipt_id int NOT NULL
    - receipt_date date NOT NULL
    - mentions varchar(255) NULL

* ORDERS

    - order_id int NOT NULL (Primary Key)
    - payment_id int NOT NULL UNIQUE (Foreign Key)
    - client_id int NOT NULL (Foreign Key)
    - order_date date NOT NULL
    - user_ip varchar(20) NOT NULL
    - discount int

* PROJECT

    - project_id int NOT NULL (Primary Key)
    - name varchar(255) NOT NULL
    - completion int NOT NULL

* UNFINISHED_GAMES

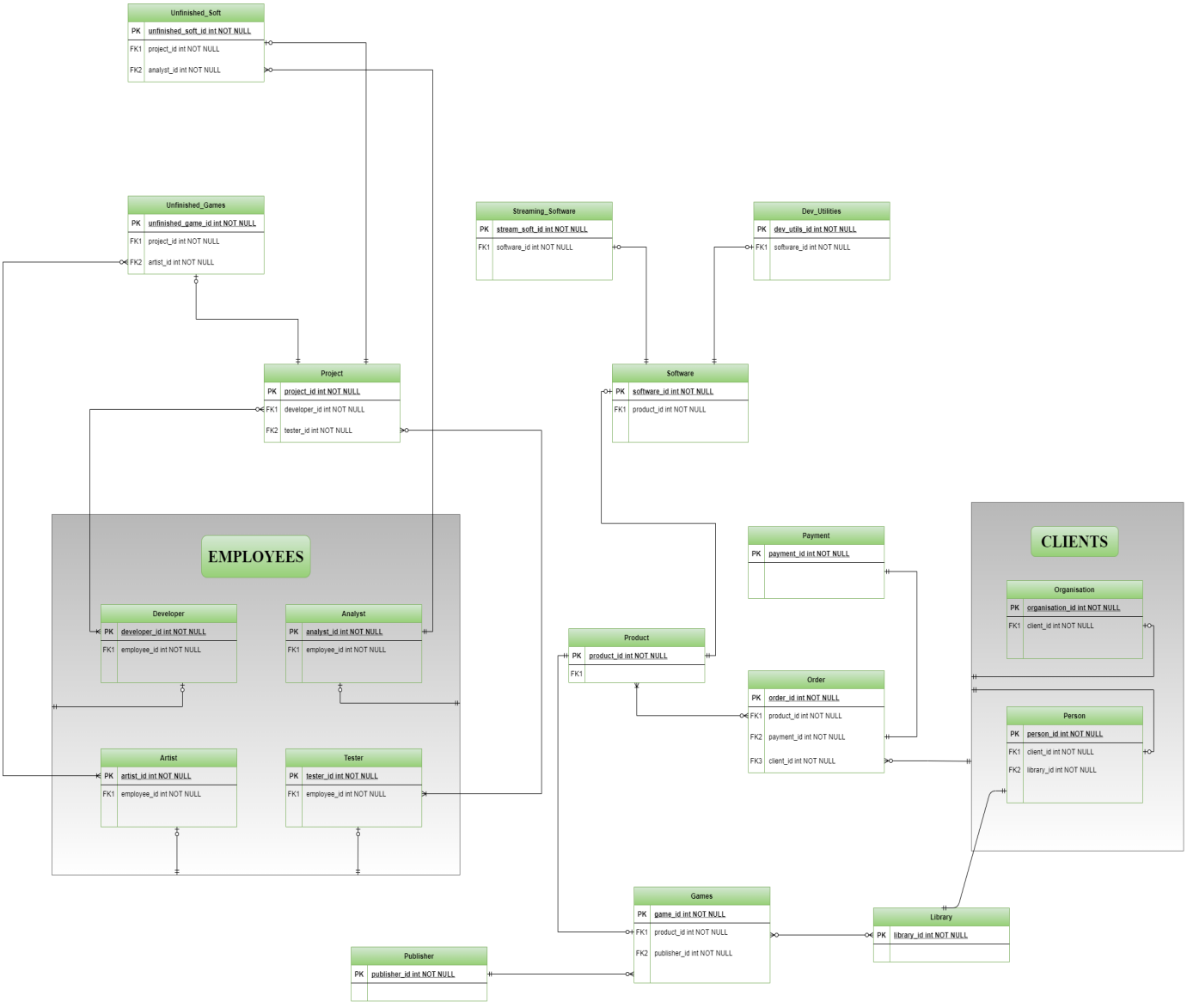
    - unfinished_game_id int NOT NULL (Primary Key)
    - project_id int NOT NULL UNIQUE (Foreign Key)
    - estimated_release_date date
    - early_access varchar(3) NOT NULL

* UNFINISHED_SOFT

    - unfinished_soft_id int NOT NULL (Primary Key)
    - project_id int NOT NULL UNIQUE (Foreign Key)
    - analyst_id int NOT NULL UNIQUE
    - stage varchar(255) NOT NULL

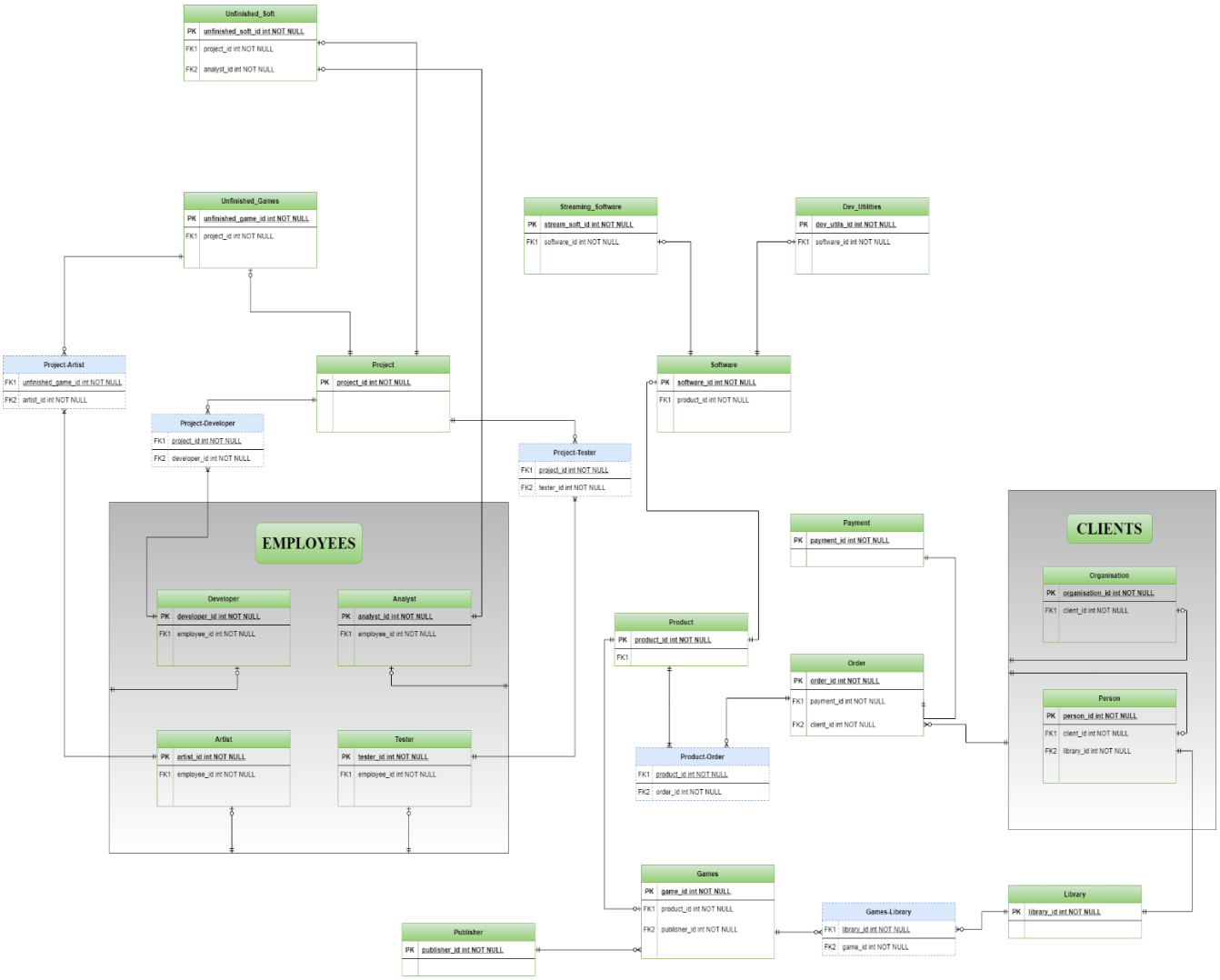
6) *Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.*

ERD



7) *Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.*

Diagrama Conceptuala



8) *Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.*

EMPLOYEES (employee_id, first name, last name, salary)

DEVELOPER (developer_id, employee_id, seniority, position)

ARTIST (artist_id, employee_id, rating)

ANALYST (net_growth, success_chance)

TESTER (tester_id, employee_id, experience, error_rate)

CLIENTS (client_id, phone, email)

ORGANISATION (organization_id, client_id, name, CUI, net_worth)

PERSON (person_id, client_id, library_id, first name, last name, age, username, fidelity)

PRODUCT (product_id, name, price, platform)

GAMES(game__id, product_id, publisher_id, genre, release_date, rating, VR)

SOFTWARE (software_id, product_id, version)

STREAMING_SOFTWARE (stream_soft_id, software_id, stream_platform)

DEV_UTILITIES (dev_utils_id, software_id, license_type)

LIBRARY (library_id, library_title)

GAMES_LIBRARY (game_id, library_id)

PUBLISHER (publisher_id, name, commission, bank_account)

ORDERS (order_id, payment_id, client_id, order_date, user_ip, discount)

PRODUCT_ORDER (product_id, order_id)

PAYMENT (payment_id, payment_type, receipt_id, receipt_date, mentions)

PROJECT (project_id, name, completion)

PROJECT_TESTER (project_id, tester_id)

PROJECT_DEVELOPER (project_id, developer_id)

PROJECT_ARTIST (unfinished_game_id, artist_id)

UNFINISHED_SOFT (unfinished_soft_id, project_id, analyst_id, stage)

UNFINISHED GAMES (unfinished_game_id, project_id, estimated_release_date, early_access)

9) *Realizarea normalizării până la forma normală 3 (FN1-FN3).*

GAMES_LIBRARY NON NF1

game_id	library_id
601	4001
602	4001,4002,4003
603	4002
604	4003

GAME_LIBRARY NF1

game_id	library_id
601	4001
602	4001
602	4002
602	4003
603	4002
604	4003

ORDER non NF2

order_id	client_id	payment_id	order_date	discount	client_name
10001	2001	6001	16/03/2021	5	Starnet
10002	3002	6002	18/05/2021	NULL	Kirigaya
10003	3002	6004	03/02/2021	10	Kirigaya

ORDER_CLIENT NF2

order_id	client_id	payment_id	order_date	discount
10001	2001	6001	16/03/2021	5
10002	3002	6002	18/05/2021	NULL
10003	3002	6004	03/02/2021	10

CLIENT NF2

client_id	client_name
2001	Starnet
3002	Kirigaya

PRODUCT non NF3

product_id	game_id	publisher_id	price	name
501	605	901	10	Bethesda
502	607	902	15	EA
503	608	901	60	Bethesda
504	609	903	25	Valve

PRODUCT NF3

product_id	game_id	publisher_id	price
501	605	901	10
502	607	902	15
503	608	901	60
504	609	903	25

GAMES NF3

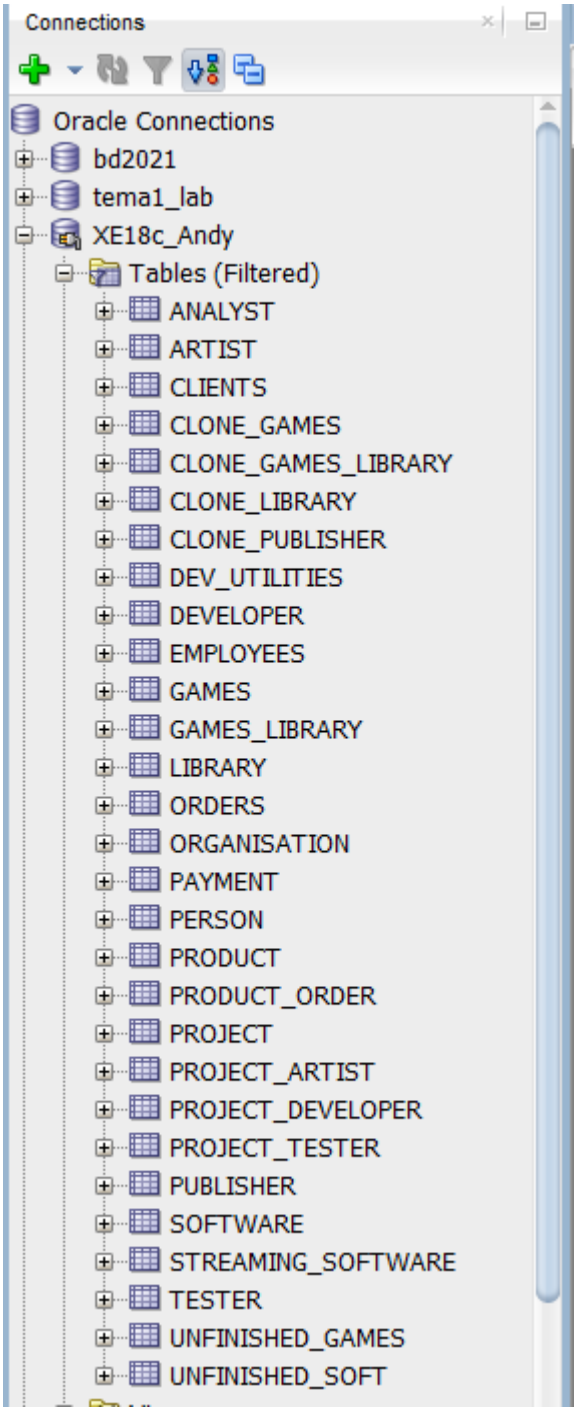
game_id	publisher_id
605	901
607	902
608	901
609	903

PUBLISHER NF3

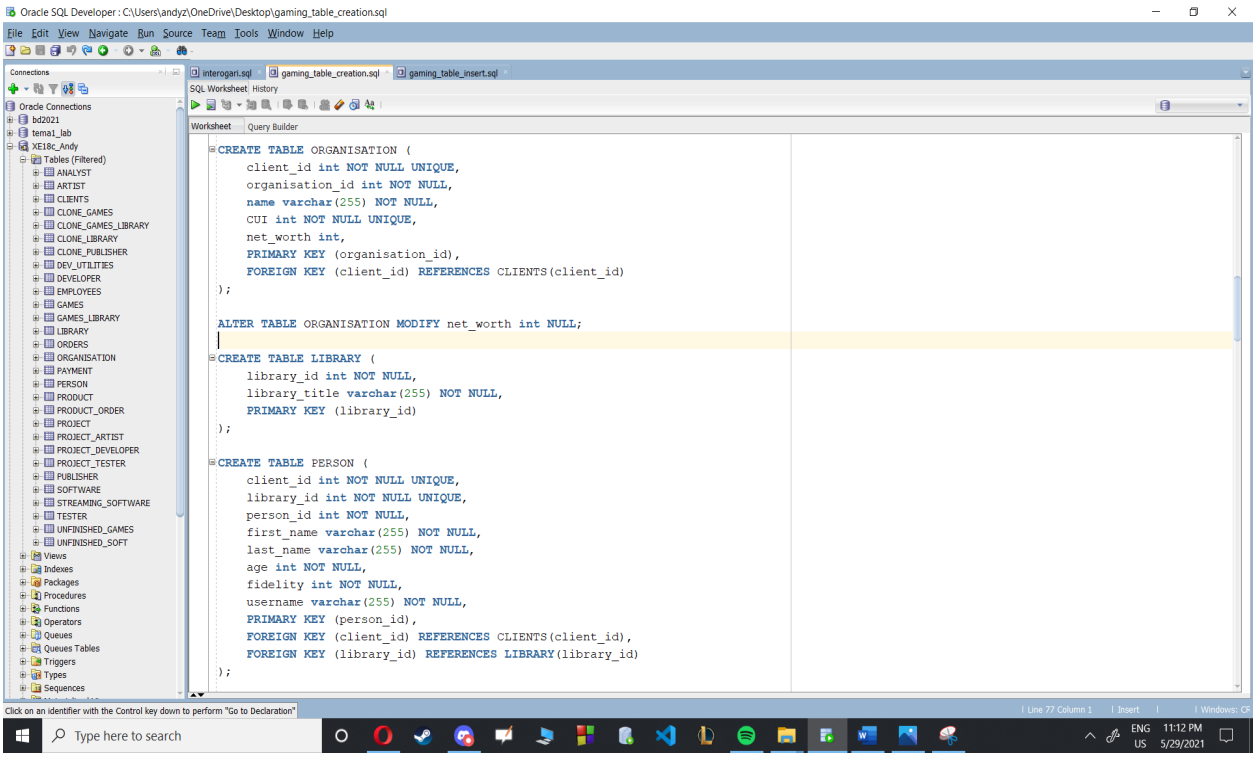
publisher_id	name
901	Bethesda
902	EA
901	Bethesda
903	Valve

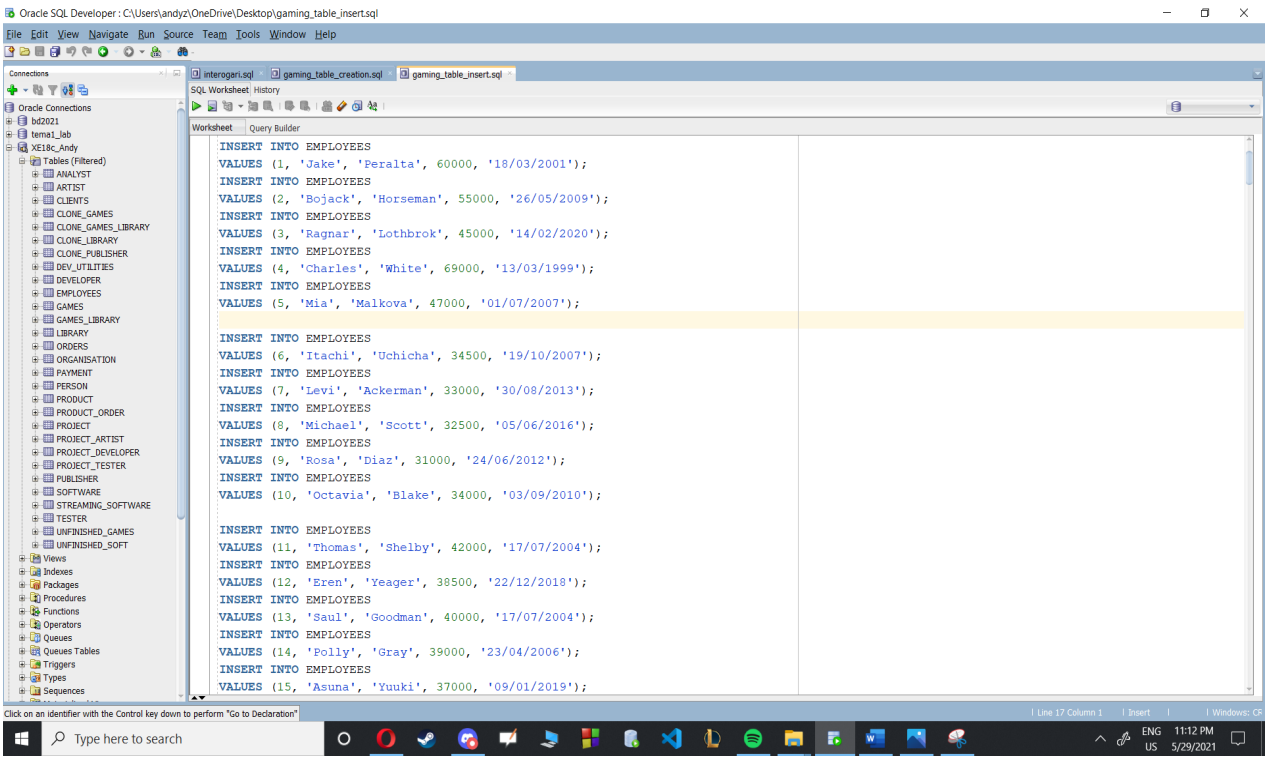
10) *Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative)*

In screenshot-ul de mai jos se poate observa ca au fost create cu succes toate tabelele din schema conceptuala



Cateva exemple de comenzi folosite pentru crearea si inserarea in tabele:





11) *Formulați în limbaj natural și implementați 5 cereri SQL complexe.*

I. *Sa se afiseze numele si emailul PERSOANELOR care au litera 'k' in nume si care au finalizat o comanda (order), inclusiv cei care nu au email, si mentiunile clientilor dupa efectuarea platii.*

```
15
16 SELECT p.first_name, p.last_name, c.email, o.order_id, pm.mentions
17 FROM PERSON p JOIN CLIENTS c ON ( c.client_id = p.client_id )
18               JOIN ORDERS o ON ( o.client_id = p.client_id )
19               JOIN PAYMENT pm ON ( pm.payment_id = o.payment_id )
20 WHERE LOWER(p.first_name) LIKE '%k%' OR LOWER(p.last_name) LIKE '%k%'
21 ORDER BY p.first_name;
```

Query Result

SQL | All Rows Fetched: 2 in 0.007 seconds

	FIRST_NAME	LAST_NAME	EMAIL	ORDER_ID	MENTIONS
1	Jack	Sparrow	(null)	10004	(null)
2	Rick	Grimes	rick.grimes@gmail.com	10001	Great!

II. *Sa se afieze numele, salariul si anii vechime ai angajatiilor care au salariul mai mic decat media salariilor, care au fost angajati dupa anul 2011, pozitia lor in companie, ordonati crescator dupa salariu.*

```
25
26 WITH aux(average) AS
27     (SELECT AVG(salary) FROM employees)
28 SELECT e.first_name, e.last_name, round(months_between(sysdate, e.hire_date)/12) seniority, e.salary
29 FROM employees e, aux
30 WHERE e.salary < aux.average
31 AND EXTRACT ( YEAR FROM e.hire_date ) > 2011
32 ORDER BY e.salary;
```

Query Result

SQL | All Rows Fetched: 6 in 0.052 seconds

	FIRST_NAME	LAST_NAME	SENIORITY	SALARY
1	Rosa	Diaz	9	31000
2	Michael	Scott	5	32500
3	Levi	Ackerman	8	33000
4	Asuna	Yuuki	2	37000
5	Eren	Yeager	2	38500
6	Gia	Paige	4	39000

III. *Sa se afiseze toate numele jocurilor, id-ul lor si libraria in care se afla, pentru jocurile care fac parte din librarii ce contin jocuri de genul 'mmorpg'.*

```
36
37 SELECT pr.name Game, gg.game_id, lib.library_id, lib.library_title
38 FROM LIBRARY lib JOIN GAMES_LIBRARY gamesl ON ( lib.library_id = gamesl.library_id )
39                  JOIN GAMES gg ON ( gamesl.game_id = gg.game_id )
40                  JOIN PRODUCT pr ON ( gg.product_id = pr.product_id )
41     WHERE lib.library_title IN
42         ( SELECT l.library_title Title
43             FROM LIBRARY l JOIN GAMES_LIBRARY gl ON ( l.library_id = gl.library_id )
44                            JOIN GAMES g ON ( gl.game_id = g.game_id )
45             WHERE LOWER(g.genre) LIKE 'mmorpg');
46
```

Query Result

SQL | All Rows Fetched: 6 in 0.004 seconds

	GAME	GAME_ID	LIBRARY_ID	LIBRARY_TITLE
1	Minecraft	601	4002	kirito/s Library
2	Skyrim	602	4002	kirito/s Library
3	Witcher 3	603	4002	kirito/s Library
4	Rust	604	4005	gamerspub/s Library
5	Black Desert Online	606	4005	gamerspub/s Library
6	Black Desert Online	606	4002	kirito/s Library

IV. *Sa se afiseze numele si prenumele angajatilor care sunt artisti si lucreaza in prezent la un joc neterminat, impreuna cu anotimpul in care au fost angajati.*

```
49
50 SELECT em.employee_id, em.first_name, em.last_name,
51 decode( extract(month from em.hire_date), 12, 'Winter', 1, 'Winter', 2, 'Winter',
52                                            3, 'Spring', 4, 'Spring', 5, 'Spring',
53                                            6, 'Summer', 7, 'Summer', 8, 'Summer',
54                                            9, 'Autumn', 10, 'Autumn', 11, 'Autumn') Season
55 FROM EMPLOYEES em
56 WHERE em.last_name IN
57 (SELECT e.last_name
58     FROM UNFINISHED_GAMES uf JOIN PROJECT_ARTIST pa ON ( uf.unfinished_game_id = pa.unfinished_game_id )
59                              JOIN ARTIST a ON ( pa.artist_id = a.artist_id )
60                              JOIN EMPLOYEES e ON ( a.employee_id = e.employee_id )
61                              );
62
```

Query Result

SQL | All Rows Fetched: 4 in 0.065 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SEASON
1	6	Itachi	Uchicha	Autumn
2	7	Levi	Ackerman	Summer
3	8	Michael	Scott	Summer
4	9	Rosa	Diaz	Summer

V. *Sa se afiseze id-ul clientilor si nr lor de comenzi, alaturi de mail-ul lor, iar pentru clientii care au dat mai mult de o comanda, sa se afiseze statutul de "Active"*

```
65
66 SELECT c.client_id, NVL(c.email, 'Not Specified'),
67 CASE WHEN COUNT(*) > 1 THEN 'Active' ELSE 'Inactive' END Status
68 FROM CLIENTS c JOIN ORDERS o ON ( o.client_id = c.client_id )
69 GROUP BY c.client_id, c.email;
70
71
```

Query Result

SQL | All Rows Fetched: 5 in 0.057 seconds

	CLIENT_ID	NVL(C.EMAIL,'NOTSPECIFIED')	STATUS
1	1010	cdprojektred@gmail.com	Inactive
2	1001	rick.grimes@gmail.com	Inactive
3	1003	gregory.house@yahoo.com	Inactive
4	1009	Not Specified	Inactive
5	1005	gamers.pub@outlook.com	Inactive

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

```
74 -- 1. Sa se adauge 10% in plus la salariul angajatilor care au vechimea de minim 10 ani.
75
76 UPDATE EMPLOYEES
77 SET SALARY = SALARY + 0.1 * SALARY
78 WHERE employee_id IN
79       (SELECT e.employee_id
80        FROM EMPLOYEES e
81        WHERE ROUND(EXTRACT( YEAR FROM sysdate ) - EXTRACT( YEAR FROM e.hire_date ) ) > 9 );
82
83 -- 2. Sa se reduca pretul jocurilor pentru copii ( age rating E ) cu 25%.
84
85 UPDATE PRODUCT
86 SET PRICE = PRICE - 0.25 * PRICE
87 WHERE product_id IN
88       (SELECT p.product_id
89       FROM PRODUCT p JOIN GAMES g ON ( g.product_id = p.product_id )
90       WHERE g.age_rating like 'E');
```

```
93 -- 3. Sa se stearga din baza de date toate companiile la care nu se cunoaste valorea neta.
94
95 DELETE from ORGANISATION og
96 WHERE og.organisation_id IN
97          ( SELECT o.organisation_id
98            FROM ORGANISATION o
99            WHERE o.net_worth IS NULL);
100
```

13) Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 10)

```
105 create sequence PUBLISHER_SEQ increment by 1 start with 904;
106 drop sequence PUBLISHER_SEQ;
107
108 insert into PUBLISHER values(PUBLISHER_SEQ.NEXTVAL, 'Iron Gate Studios', 5, 'SE5729256165998787184373');
109 insert into PUBLISHER values(PUBLISHER_SEQ.NEXTVAL, 'EA', 60, 'XK551151817765264628');
110
```

Query Result | Script Output

Task completed in 0.032 seconds

```
Sequence PUBLISHER_SEQ dropped.


Sequence PUBLISHER_SEQ created.


1 row inserted.
```

14) Crearea unei vizualizări compuse. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.

Creare :

```
122 CREATE TABLE clone_library AS SELECT * FROM LIBRARY;
123
124 CREATE TABLE clone_games_library AS SELECT * FROM GAMES_LIBRARY;
125
126 CREATE TABLE clone_games AS SELECT * FROM GAMES;
127
128 CREATE TABLE clone_publisher AS SELECT * FROM PUBLISHER;
129
130
131 CREATE OR REPLACE VIEW GAMES_VIEWS AS
132 SELECT cl.library_title Title, cl.library_id, cg.game_id, cg.genre, cp.publisher_id, cp.name
133             FROM clone_library cl JOIN clone_games_library cgl ON ( cl.library_id = cgl.library_id )
134                                   JOIN clone_games cg ON ( cgl.game_id = cg.game_id )
135                                   JOIN clone_publisher cp ON ( cp.publisher_id = cg.publisher_id );
136
```

Operatie Permisa :

```
138 -- Sa se afiseze toate jocurile de genul RPG cu publisherul BETHESDA si sa se specifice in ce librarii se regasesc acestea.
139
140 -- Operatie posibila
141
142 SELECT gv.game_id, gv.title Library, gv.library_id ID
143 FROM GAMES_VIEWS gv
144 WHERE gv.name like 'Bethesda';
```

Operatie Interzisa :

```
146 -- Operatie imposibila
147
148 INSERT INTO GAMES_VIEWS
149 VALUES ('Fake Library', 4211, 679, 'Fake', 911, 'Fake Publisher');
150
```

15) *Crearea unui index care să optimizeze o cerere de tip căutare cu 2 criterii. Specificați cererea.*

Cererea : Sa se afisez numele complet al angajatilor al caror prenume se termina in litera „y”, sau al caror nume de familie se termina in gruparea de litere „man”.

```
154 CREATE INDEX search_employees
155 ON EMPLOYEES(first_name, last_name);
156
157 SELECT first_name, last_name
158 FROM EMPLOYEES
159 WHERE last_name LIKE '%man' OR first_name LIKE '%y';
160
161 drop index search_employees;
```

Script Output | Query Result

SQL | All Rows Fetched: 7 in 0.006 seconds

	FIRST_NAME	LAST_NAME
1	Bojack	Horseman
2	Levi	Ackerman
3	Saul	Goodman
4	Polly	Gray
5	Marty	Byrde
6	Barney	Stinson
7	Jeese	Pinkman