Magazin online de cadouri
Avramescu Cosmin-Alexandru – 344C3

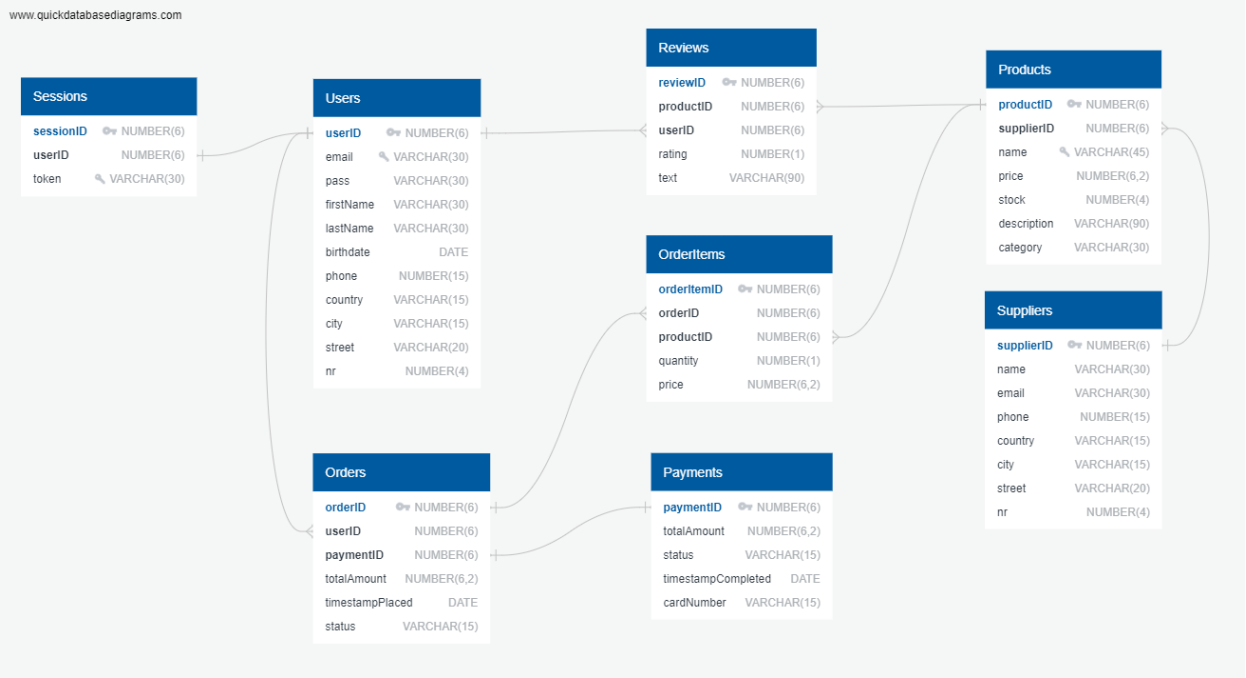
Descrierea Temei

Proiectul consta in proiectarea unei baze de date pentru un magazin online de cadouri, popularea cu date a acestei baze si afisarea datelor prin intermediul PowerBI.

1.1 Descrierea bazei de date

Baza de date contine 8 tabele: Sessions, Users, Orders, OrderItems, Payments, Products, Reviews si Suppliers, cu relatii one-to-many sau one-to-one dupa cum se observa in poza de mai jos.

1.2 Structura tabelelor si descrierea constrangerilor de integritate



a) **Sessions**: Acest tabel are scopul de a retine sesiunea activa a userului. Scopul sau este in principal de scalabilitate, deoarece se pot adauga ulterior tabele pentru cosul de cumparaturi al utilizatorului, iar produsele din acest cos sa se pastreze datorita sesiunii.

Coloana	Tip	Constrangere
sessionID	NUMBER(6)	Primary Key
userID	NUMBER(6)	Foreign Key -> Users
token	VARCHAR(30)	Unique

b) **Users**: Acest tabel are scopul de a retine date esentiale despre utilizatorii magazinului online de cadouri.

Coloana	Tip	Constrangere
userID	NUMBER(6)	Primary Key
email	VARCHAR(30)	Unique, NOT NULL
pass	VARCHAR(30)	NOT NULL
firstName	VARCHAR(30)	NOT NULL
lastName	VARCHAR(30)	NOT NULL
birthdate	DATE	NOT NULL
phone	VARCHAR(15)	Unique, NOT NULL
country	VARCHAR(15)	
city	VARCHAR(15)	
street	VARCHAR(20)	
nr	NUMBER(4)	

Astfel, avem nevoie sa stim credentialele contului, nume, prenume, data nasterii si adresa de livrare.

c) **Orders**: Acest tabel are scopul de a retine comenzile date de fiecare utilizator. Contine suma totala ce

Coloana	Tip	Constrangere
orderID	NUMBER(6)	Primary Key
userID	NUMBER(6)	Foreign Key -> Users
paymentID	NUMBER(6)	Foreign Key -> Payments
totalAmount	NUMBER(6,2)	NOT NULL
timestampPlaced	DATE	NOT NULL
status	VARCHAR(15)	

trebuie platita, statusul comenzii si data la care s-a plasat comanda.

d) **OrderItems**: Acest tabel are scopul de a retine fiecare produs din comanda unui utilizator, impreuna cu

Coloana	Tip	Constrangere
orderItemID	NUMBER(6)	Primary Key
orderID	NUMBER(6)	Foreign Key -> Orders
productID	NUMBER(6)	Foreign Key -> Products
quantity	NUMBER(1)	NOT NULL
price	NUMBER(6,2)	NOT NULL

cantitatea dorita si pretul (daca a comandat 2 produse de 50 de lei, aici se va retine pretul de 100 de lei).

e) **Products**: Acest tabel are scopul de a retine date despre fiecare produs, cum ar fi numele, descrierea,

Coloana	Tip	Constrangere
productID	NUMBER(6)	Primary Key
supplierID	NUMBER(6)	Foreign Key -> Orders
name	VARCHAR(45)	Unique, NOT NULL
price	NUMBER(6,2)	NOT NULL
stock	NUMBER(4)	
description	VARCHAR(90)	
category	VARCHAR(30)	NOT NULL

categoria din care face parte, stocul disponibil si pretul.

f) **Suppliers**: Acest tabel are scopul de a retine datele despre furnizorii de produse, cum ar fi date de contact

Coloana	Tip	Constrangere
supplierID	NUMBER(6)	Primary Key
name	VARCHAR(30)	NOT NULL
email	VARCHAR(30)	Unique, NOT NULL
phone	VARCHAR(15)	Unique, NOT NULL
country	VARCHAR(15)	NOT NULL
city	VARCHAR(15)	
street	VARCHAR(20)	
nr	NUMBER(4)	

si adresa furnizorilor.

g) **Reviews**: Acest tabel are scopul de a retine comentariile date de utilizatori produselor. Se retin informatiile despre rating-ul oferit si textul comentariului.

Coloana	Tip	Constrangere
reviewID	NUMBER(6)	Primary Key
productid	VARCHAR(30)	Foreign Key -> Products
userID	VARCHAR(30)	Foreign Key -> Users
rating	VARCHAR(15)	NOT NULL
text	VARCHAR(15)	NOT NULL

1.4 Descrierea procedurilor:

```
CREATE OR REPLACE package pck_gift_shop
AS
    PROCEDURE users_with_reviews_for_other_products_proc(reviews OUT sys_refcursor);
    PROCEDURE orders_with_delay_greater_than_7_days(orders OUT sys_refcursor);
    PROCEDURE favorite_product_categories(categories OUT sys_refcursor);
END pck_gift_shop;
```

- *Procedura 1:*

Afiseaza produsele care au primit review de la utilizatori care nu au cumparat de fapt acele produse. Utilitatea este ca se poate afisa pe site (“Achizitie verificata” – ca la Emag), daca utilizatorul care da comentariul chiar a cumparat produsul respectiv. Se merge pe lantul comenzii cu join si se compara daca id-ul de la OrderItem este la fel cu id-ul de la Product si se pune conditia ca id-ul userului din Order sa fie diferit de id-ul userului din Review.

```
CREATE OR REPLACE package body pck_gift_shop
AS
    PROCEDURE users_with_reviews_for_other_products_proc(reviews OUT sys_refcursor)
    IS
    BEGIN
        OPEN reviews FOR
        SELECT
            r.productID,
            p.name AS ProductName,
            u.userID AS BuyingUser,
            r.userID AS CommentingUser,
            r.reviewID,
            r.rating AS ReviewRating,
            r.text AS ReviewText
        FROM Orders o
            INNER JOIN Users u on o.userID = u.userID
            INNER JOIN OrderItems i on i.orderID=o.orderID
            INNER JOIN Reviews r on o.userID != r.userID
            INNER JOIN Products p on p.productID = r.productID and r.productID = i.productID;
    END users_with_reviews_for_other_products_proc;
```

- *Procedura 2:*

Afiseaza comenzile care au statusul din Payment “Completed” (adica userul a platit comanda), dar au statusul din Order “Processing”, iar diferenta dintre timestamp-ul cand s-a achitat plata si timestamp-ul curent este mai mare de 7 zile. Practic se afiseaza comenzile nerezolvate cu intarzieri mari pentru a se analiza sursa problemei. E posibil ca aceste probleme sa fie din cauza furnizorului de produs daca tot el face livrarea, asa ca se afiseaza toate datele furnizorului pentru a putea fi contactat.

```
PROCEDURE orders_with_delay_greater_than_7_days(orders OUT sys_refcursor)
IS
BEGIN
    OPEN orders FOR
    SELECT
        o.orderID,
        u.userID,
        o.totalAmount,
        p.timestampCompleted AS PaymentCompleted,
        EXTRACT(DAY FROM (CURRENT_TIMESTAMP - p.timestampCompleted)) AS days_difference,
        pr.productID,
        pr.name AS ProductName,
        SUM(i.quantity) AS TotalQuantity,
        s.supplierID,
        s.name AS SupplierName,
        s.country AS SupplierCountry,
        s.email,
        s.phone
    FROM
        Orders o
        INNER JOIN Users u ON o.userID = u.userID
        INNER JOIN Payments p ON o.paymentID = p.paymentID
        INNER JOIN OrderItems i ON o.orderID = i.orderID
        INNER JOIN Products pr ON i.productID = pr.productID
        INNER JOIN Suppliers s ON s.supplierID = pr.supplierID
    WHERE
        p.status = 'Completed'
        AND o.status = 'Processing'
    GROUP BY
        u.userID, o.orderID, o.totalAmount, o.timestampPlaced, p.timestampCompleted,
        pr.productID, pr.name, pr.stock, s.supplierID, s.name, s.country, s.email, s.phone
    HAVING
        CURRENT_TIMESTAMP - p.timestampCompleted > INTERVAL '7' DAY
    ORDER BY 1;
END orders_with_delay_greater_than_7_days;
```

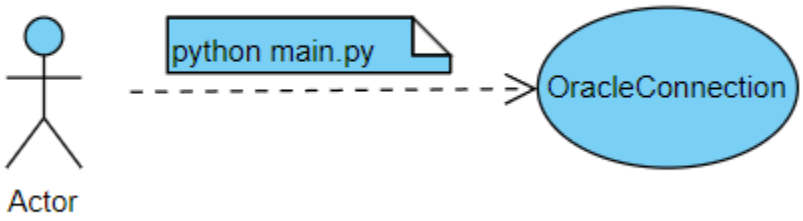
- *Procedura 3:*

Afiseaza pentru fiecare user categoriile de produs din care cumpara cel mai des. Utilitatea este ca acesta poate primi ulterior reclame personalizate in functie de categoriile sale preferate de produse. In clauza having se calculeaza numarul maxim de produse cumparate din fiecare categorie si apoi se afiseaza in primul select alaturi de numele categoriei, id-ul, tara si orasul utilizatorului.

```
    PROCEDURE favorite_product_categories(categories OUT sys_refcursor)
    IS
    BEGIN
        OPEN categories FOR
        SELECT
            u.userID,
            p.category,
            COUNT(DISTINCT p.productID) AS num_products_ordered,
            u.country,
            u.city
        FROM
            Users u
            INNER JOIN Orders o ON u.userID = o.userID
            INNER JOIN OrderItems i ON i.orderID = o.orderID
            INNER JOIN Products p ON p.productID = i.productID
        GROUP BY
            u.userID, p.category, u.country, u.city
        HAVING
            COUNT(DISTINCT p.productID) = (
                SELECT MAX(product_count)
                FROM (
                    SELECT COUNT(DISTINCT p_inner.productID) AS product_count
                    FROM Users u_inner
                    INNER JOIN Orders o_inner ON u_inner.userID = o_inner.userID
                    INNER JOIN OrderItems i_inner ON i_inner.orderID = o_inner.orderID
                    INNER JOIN Products p_inner ON p_inner.productID = i_inner.productID
                    WHERE u_inner.userID = u.userID
                    GROUP BY p_inner.category
                )
            )
        ORDER BY 1;
    END favorite_product_categories;
END pck_gift_shop;
```

2. Descrierea aplicatiei

2.1 Diagrama de clase

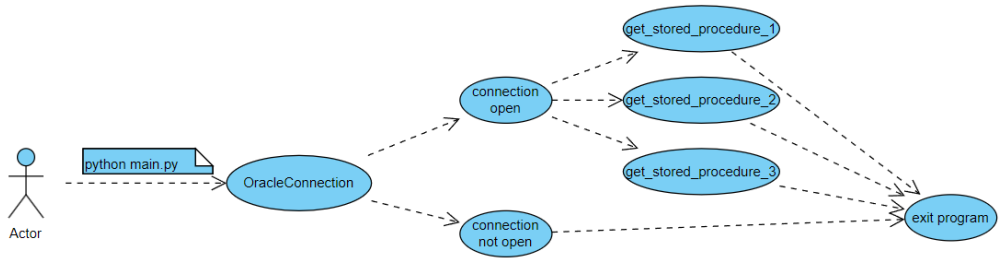


2.2 Structura clasei

Variabila	Valoare
self.host	localhost
self.port	1522
self.sid	XE
self.username	system
self.password	parolaAiaPuternic4

Metoda	Utilizare
openConnection()	deschide conexiunea la containerul bazei de date Oracle
closeConnection()	inchide conexiunea la baza de date
get_users_with_reviews_for_other_products()	apeleaza procedura stocata pentru raportul 1
get_orders_with_delay_greater_than_7_days()	apeleaza procedura stocata pentru raportul 2
get_favorite_product_categories()	apeleaza procedura stocata pentru raportul 3

2.3 Diagrama de stari si workflow-ul aplicatiei



2.4 Prezentarea modului prin care se face conexiunea cu baza de date

Crearea bazei de date cu numele my-oracle-db intr-un container Docker, identificata unic prin SID-ul XE si care ruleaza pe localhost pe portul 1521.

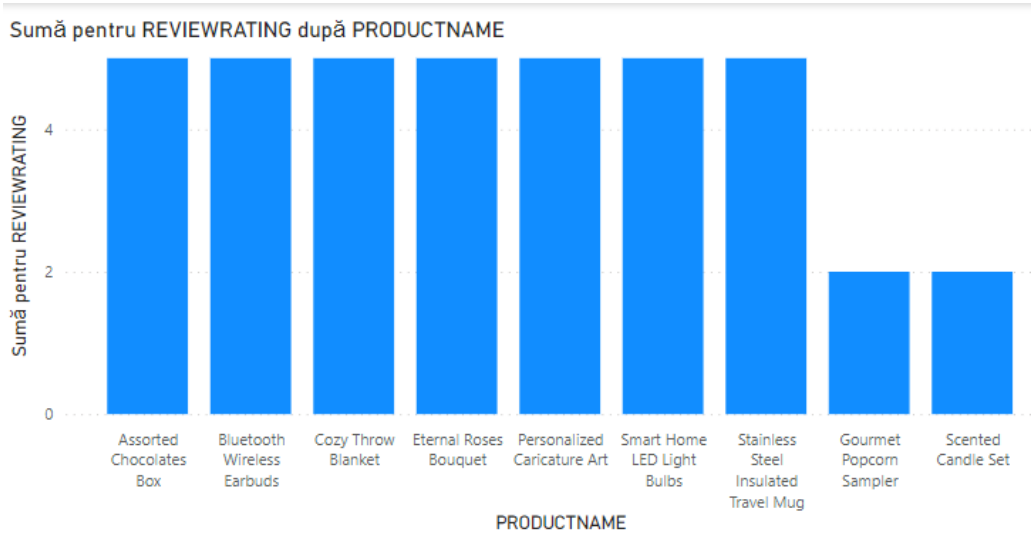
```
-docker login container-registry.oracle.com
-docker pull container-registry.oracle.com/database/express:latest
-docker run -d --name my-oracle-db -p 1521:1521 -p 5500:5500 -e ORACLE_PWD=parolaAiaPuternic4 -e Oracle_SID=XE -e ORACLE_ALLOW_REMOTE=true container-registry.oracle.com/database/express:latest
-pip install cx_Oracle
```

In python, se apeleaza cx_Oracle.makedsn(), cu parametri host, port si sid pentru a se crea un Domain Source Name (un connection string pe care il dam ca parametru catre cx_Oracle.connect(), alaturi de user si parola pentru a face conexiunea cu baza de date din container).

3. **Capturi de ecran pentru rapoartele grafice si concluzii**

Raport grafic 1:

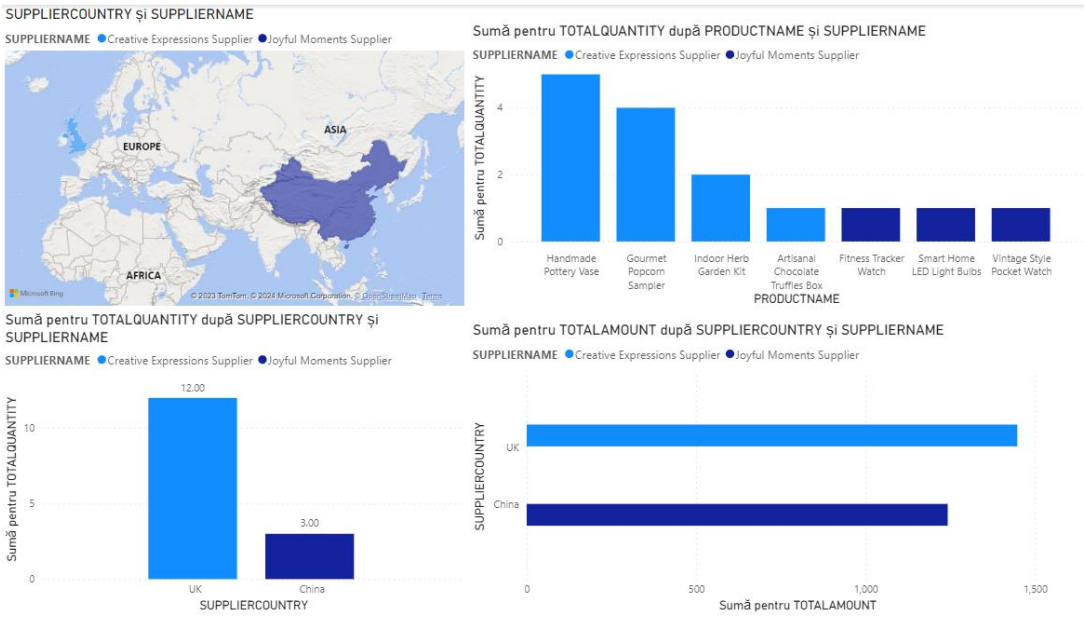
Se pot observa trenduri – daca este un flux de comentarii pozitive, se poate deduce ca acei oameni vor sa faca o reclama falsa produsului. Daca insa comentariile sunt prezente comentarii negative de la oameni care nu au cumparat produsul, acestea afecteaza cumpararea pe viitor a acelor produse. Din datele extrase, de la oamenii care au comentat la produse pe care nu le-au cumparat, sunt 2 comentarii negative, iar restul sunt pozitive. In PowerBI, se poate observa ca am extras si un tabel pentru a vedea BuyingUserId si CommentingUserId pentru a vedea daca exista un pattern (daca este un singur user care face asta sau sunt mai multi).



REVIEW_TEXT	BUYING_USER_ID	COMMENTING_USER_ID	REVIEW_RATING	PRODUCT_ID
Bluetooth earbuds have great sound quality.	1	4	5	1
I like it very much!	4	6	5	1
I needed so much something like this!	8	1	5	1
It does not taste as expected...	9	4	2	1
My dad was happy with this gift.	3	4	5	1
My girlfriend loves them!!	1	2	5	1
Our kid is very happy with this!	9	7	5	1
The candles are not scented at all...	2	3	2	1
Very good!	1	3	5	1

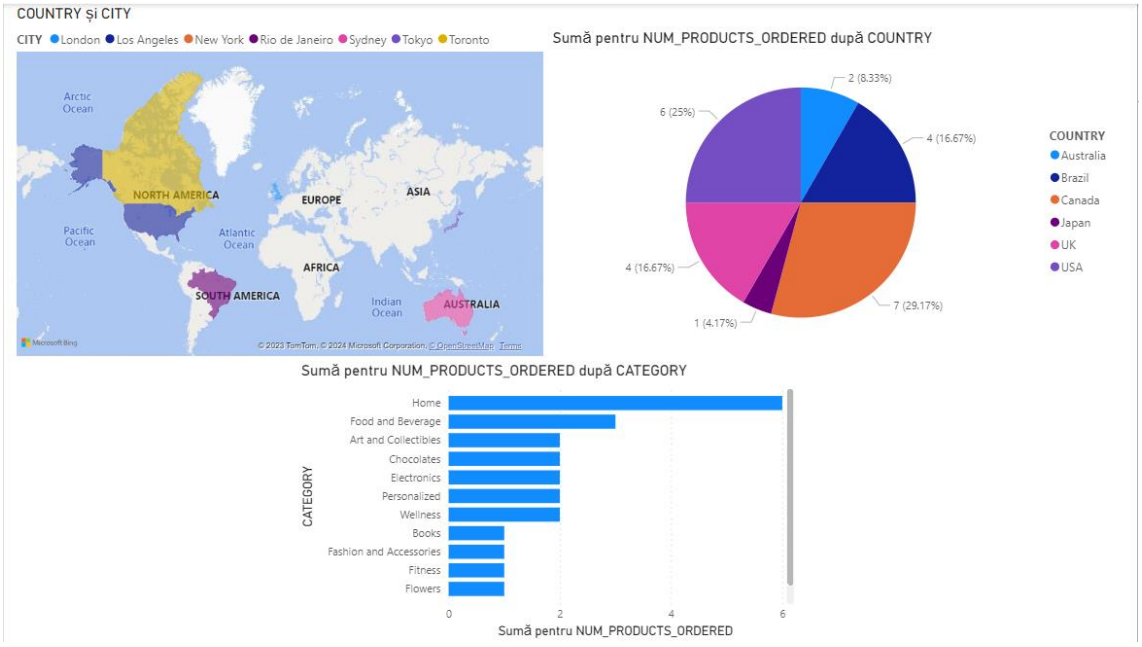
Raport grafic 2:

Se pot observa date despre furnizorii care au intarzieri mai mari de 7 zile de la data achitarii platii de catre utilizator. Cei 2 furnizori cu probleme sunt din UK, respective China. In graficul din stanga jos se observa cantitatea de produse intarziate (practice au fost comandate cate 3, 4, 5 produse de un tip deodata si astfel, s-au comandat mai multe produse din UK). Insa in graficul din dreapta jos se observa ca desi s-au comandat mai putine produse din China, valoarea acestora in bani este aproape la fel de mare ca la cele din UK. In graficul din dreapta sus putem care sunt produsele cu probleme, din ce tara este furnizorul (albastru inchis – China, albastru deschis - UK) si care este cantitatea comandata – pe Oy de la bar chart.



Raport grafic 3:

Se poate observa ca cele mai cumparate cadouri de pe site sunt de departe cele din categoria Home, urmand apoi pe locul 2 cele din categoria Food & Beverages. De asemenea, se poate observa in graficul din dreapta sus din ce tara s-au inregistrat cele mai multe comenzi pe site (Canada, urmata apoi de USA). In graficul din stanga sus se pot observa si orasele din care s-au facut comenzile.



4.Bibliografie:

1. Laboratoarele de pe OCW
2. Documentatia oficiala Oracle
3. Site-urile oficiale Oracle si Microsoft (pentru instalarea clientului, lucrul cu PowerBI, etc)
4. StackOverFlow la nevoie