

1. Descrierea temei economice

În contextul modern, gestionarea eficientă a resurselor unei biblioteci – fie că este vorba despre cărți, utilizatori, sau servicii auxiliare – reprezintă o provocare economică și operațională. O bază de date devine indispensabilă pentru a simplifica aceste procese, reducând costurile și crescând productivitatea.

Scopul și impactul economic al gestionării unei biblioteci cu ajutorul unei baze de date:

- O bibliotecă modernă poate genera venituri din servicii adiționale (ex. abonamente premium, xerox, sau acces la resurse digitale). O bază de date permite monitorizarea acestor fluxuri financiare pentru a evalua profitabilitatea fiecărei activități.
- Automatizarea în ceea ce privește alertele referitoare la stoc/termene de returnare ajută la reducerea cheltuielilor cu resursele umane și hârtia utilizată pentru evidențele manuale.
- Toate tranzacțiile (împrumuturi, returnări, plăți) sunt înregistrate electronic, ceea ce elimină riscurile de fraudă sau erori.
- Accesul la rapoarte financiare detaliate ajută managerii să ia decizii bine informate cu privire la alocarea resurselor.
- O bibliotecă care utilizează o bază de date poate gestiona mai eficient o creștere a numărului de utilizatori și resurse. Aceasta permite extinderea serviciilor fără a crește proporțional costurile.

2. Baza de date va trebui să poată acoperi următoarele funcții:

• **Gestiunea inventarului, a colecției de cărți:**

- Adăugare/ștergere cărți din baza de date.
- Actualizare stoc după un împrumut/returnare.
- Evidența completă a cărților disponibile pentru împrumutat.
- Semnalarea cărților cu stoc redus.

Exemplu: În cazul în care exemplarele unei cărți sunt insuficiente, baza de date poate genera rapoarte care să sprijine deciziile de achiziție.

• **Monitorizarea împrumuturilor:**

- Urmărirea istorică a cărților împrumutate.
- Semnalarea termenului limită de returnare, prevenind întârzierile și pierderile economice.
- `data_returnare`: dacă rămâne NULL, cartea este considerată nereturnată.

Statusurile împrumuturilor sunt Activ, Întarziat și Închis, calculate în funcție de limita de zile din abonament:

- **Activ:** Împrumuturile care sunt în termen (înainte de expirarea `limita_zile`).
ex. `data_imprumut + limita_zile >= data_curentă`.
- **Întarziat:** Împrumuturile care au depășit limita de zile față de data împrumutului.
ex. `data_imprumut + limita_zile < data_curentă`.
- **Închis:** Împrumuturi returnate (cu `data_returnare` completată).

Exemplu `limita_zile` pentru tipuri de abonamen:

- Basic: 14 zile
- Premium: 30 zile
- VIP: 60 zile
- Student: 21 zile

• **Gestionarea utilizatorilor (abonaților):**

- Înregistrarea utilizatorilor cu date personale și abonamente.

- Evidența activității fiecărui utilizator (împrumuturi, plăți, servicii utilizate).
Exemplu: Un utilizator cu abonament activ poate accesa mai multe servicii (împrumuturi multiple, reduceri la xerox sau perioada mai mare pentru returnare)
- **Servicii auxiliare:**
 - Gestiunea plăților pentru abonamente sau alte servicii (ex. xerox).
 - Raportarea veniturilor din aceste servicii pentru analize economice.
Exemplu: Raportarea datelor (ex. cele mai solicitate cărți, venituri lunare) ajută la îmbunătățirea strategiilor de achiziție și serviciilor oferite

3. Structura și descrierea bazei de date

Baza de date proiectată pentru o bibliotecă va conține cel puțin următoarele tabele:

Tabela Abonamente- Definirea tipurilor de abonamente oferite de bibliotecă.

- Cheie primară (PK): abonament_id
- Atribute:
 - abonament_id: Identificator unic pentru abonament.
 - tip_abonament: Tipul de abonament (ex. Basic, Premium).
 - descriere: Detalii despre abonament.
 - taxa_lunara: Costul abonamentului lunar.
 - limita_carti: Numărul maxim de cărți ce pot fi împrumutate simultan.
 - limita_zile: Numărul maxim de zile de împrumut.
 - reducere_servicii: Reducere procentuală pentru servicii.
 - penalizare: Penalizare procentuală pentru întârziere.
 - este_activ: Abonamentul este activ.

Tabela Utilizatori - Gestionarea informațiilor despre utilizatorii bibliotecii.

- Cheie primară (PK): utilizator_id
- Cheie externă (FK): abonament_id → Abonamente(abonament_id)
- Relație: Fiecare utilizator aparține unui abonament.
- Atribute:
 - utilizator_id: Identificator unic pentru utilizator.
 - nume: Numele complet al utilizatorului.
 - email: Adresa de email (unic).
 - telefon: Număr de telefon.
 - data_inregistrare: Data înregistrării utilizatorului.
 - abonament_id: Referință către tipul de abonament din tabela Abonamente.
 - abonament_activ: Daca abonamentul este activ

Tabela Carti - Evidența cărților disponibile în bibliotecă.

- Cheie primară (PK): **carte_id**
- Atribute:
 - carte_id: Identificator unic pentru carte.
 - titlu: Titlul cărții.
 - autor: Autorul cărții.
 - categorie: Categoria cărții (ex. Istorie, Psihologie).
 - exemplare_disponibile: Numărul de exemplare disponibile.
 - este_disponibila: Indicator dacă cartea este disponibilă (true/false).

Tabela Imprumuturi - Evidența împrumuturilor efectuate de utilizatori.

- Cheie primară (PK): **imprumut_id**
- Chei externe (FK):
 - utilizator_id → Utilizatori(utilizator_id)

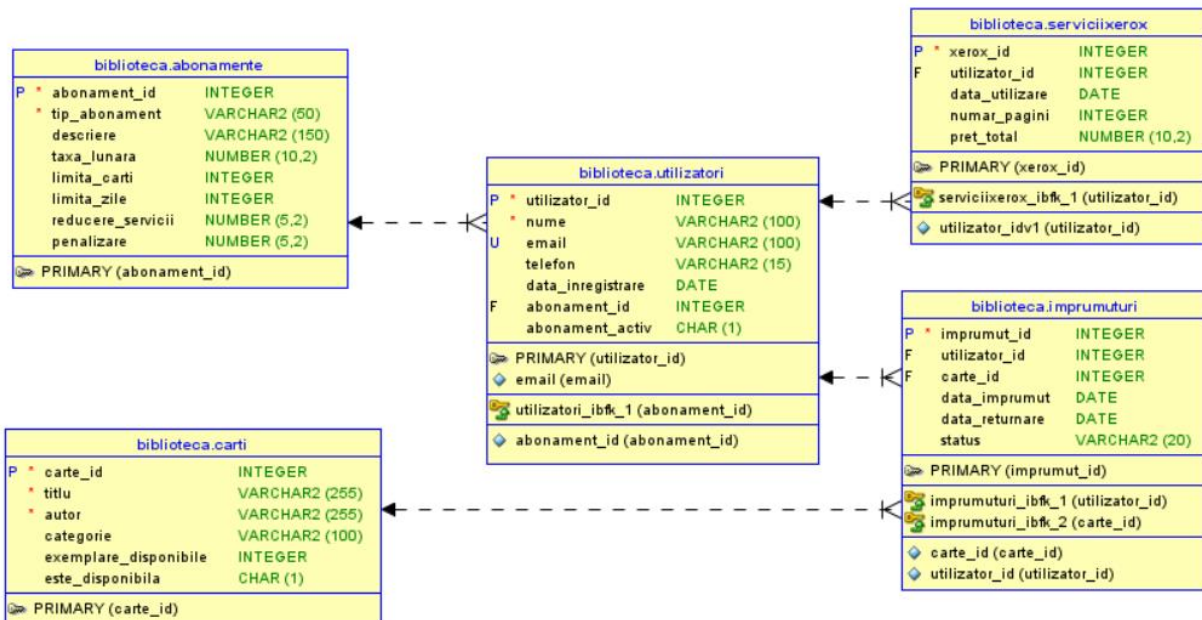
- $\text{carte_id} \rightarrow \text{Carti}(\text{carte_id})$
- Relație: Fiecare împrumut este asociat unui utilizator și unei cărți.
- Atribute:
 - imprumut_id : Identificator unic pentru împrumut.
 - utilizator_id : Referință către utilizatorul care a împrumutat cartea.
 - carte_id : Referință către cartea împrumutată.
 - data_imprumut : Data la care s-a efectuat împrumutul.
 - data_returnare : Data returnării cărții.
 - status : Statusul împrumutului (ex. Activ, Intarziat, Inchis).

Tabela ServiciiXerox - Gestionarea serviciilor de xerox utilizate de clienți.

- Cheie primară (PK): **xerox_id**
- Cheie externă (FK): **utilizator_id** → **Utilizatori(utilizator_id)**
- Relație: Fiecare serviciu de xerox este asociat unui utilizator.
- Atribute:
 - xerox_id : Identificator unic pentru serviciul de xerox.
 - utilizator_id : Referință către utilizatorul care a utilizat serviciul.
 - numar_pagini : Numărul de pagini xeroxate.
 - cost_total : Costul total calculat în funcție de un preț fix pe pagina și eventualele reduceri din abonament.
 - data_serviciu : Data la care a fost utilizat serviciul.

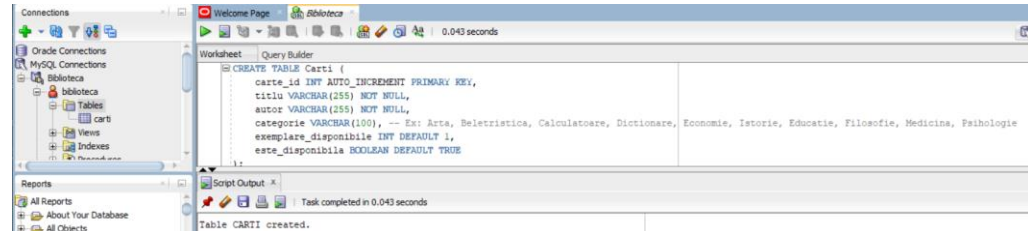
Relații:

- Abonamente → Utilizatori (1:N):
 - Fiecare tip de abonament (abonament_id) poate fi utilizat de mai mulți utilizatori.
- Utilizatori → Imprumuturi (1:N):
 - Fiecare utilizator (utilizator_id) poate efectua mai multe împrumuturi.
- Carti → Imprumuturi (1:N):
 - Fiecare carte (carte_id) poate fi împrumutată de mai mulți utilizatori în momente diferite.
- Utilizatori → ServiciiXerox (1:N):
 - Fiecare utilizator (utilizator_id) poate utiliza serviciile de xerox de mai multe ori.



4. Comenzi LLD, LMD

```
CREATE TABLE Carti (  
    carte_id INT AUTO_INCREMENT PRIMARY KEY,  
    titlu VARCHAR(255) NOT NULL,  
    autor VARCHAR(255) NOT NULL,  
    categorie VARCHAR(100), -- Ex: Arta, Beletristica, Calculatoare, Dictionare, Economie, Istorie, Educatie, Filosofie,  
    Medicina, Psihologie  
    exemplare_disponibile INT DEFAULT 1,  
    este_disponibila BOOLEAN DEFAULT TRUE  
);
```



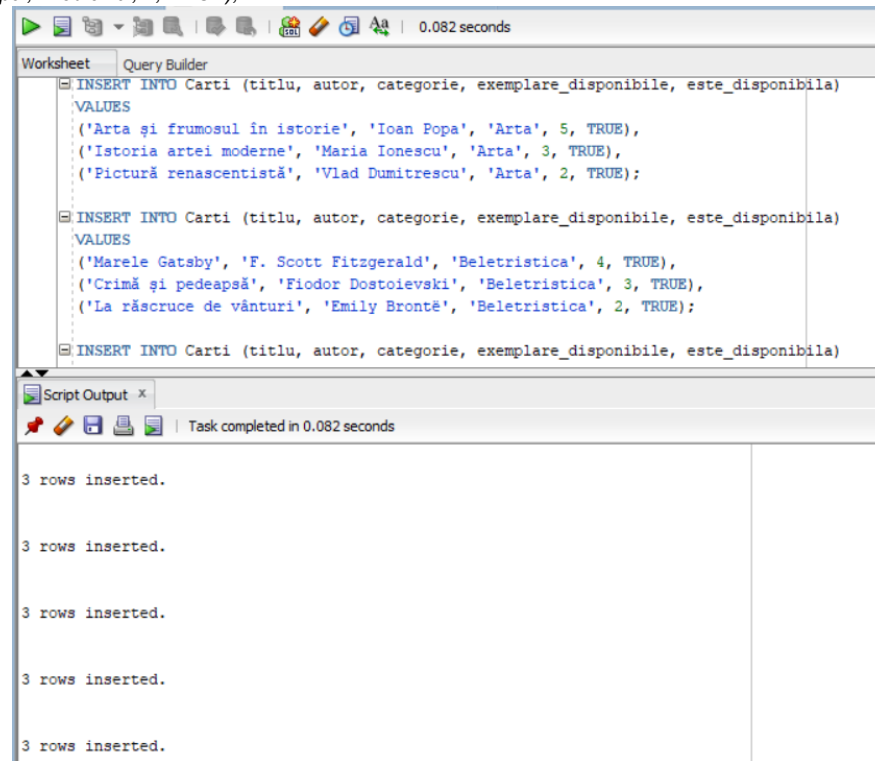
```
INSERT INTO Carti (titlu, autor, categorie, exemplare_disponibile, este_disponibila) VALUES ('Arta și frumosul în istorie',  
'Ioan Popa', 'Arta', 5, TRUE), ('Istoria artei moderne', 'Maria Ionescu', 'Arta', 3, TRUE), ('Pictură renașcentistă', 'Vlad  
Dumitrescu', 'Arta', 2, TRUE);
```

```
INSERT INTO Carti (titlu, autor, categorie, exemplare_disponibile, este_disponibila) VALUES ('Marele Gatsby', 'F. Scott  
Fitzgerald', 'Beletristica', 4, TRUE), ('Crimă și pedeapsă', 'Fiodor Dostoievski', 'Beletristica', 3, TRUE), ('La răscruce de  
vânturi', 'Emily Brontë', 'Beletristica', 2, TRUE);
```

```
INSERT INTO Carti (titlu, autor, categorie, exemplare_disponibile, este_disponibila) VALUES ('Introducere în programare',  
'Andrei Mihai', 'Calculatoare', 5, TRUE), ('Algoritmi și structuri de date', 'Elena Radu', 'Calculatoare', 3, TRUE), ('Bazele  
dezvoltării web', 'Cristian Stan', 'Calculatoare', 4, TRUE);
```

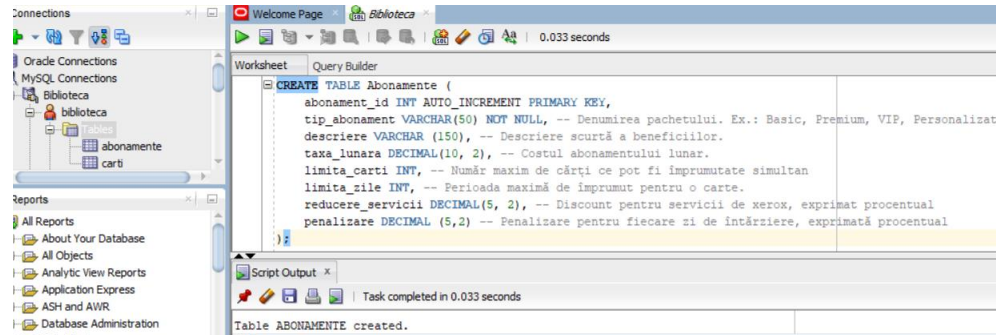
```
INSERT INTO Carti (titlu, autor, categorie, exemplare_disponibile, este_disponibila) VALUES ('Psihologia emoțiilor', 'Maria  
Ionescu', 'Psihologie', 4, TRUE), ('Introducere în psihoterapie', 'George Petrescu', 'Psihologie', 3, TRUE), ('Dezvoltare  
personală', 'Elena Mihai', 'Psihologie', 2, TRUE);
```

```
INSERT INTO Carti (titlu, autor, categorie, exemplare_disponibile, este_disponibila) VALUES ('Anatomia umană', 'Ioan  
Matei', 'Medicina', 5, TRUE), ('Tratamente moderne', 'Adriana Stan', 'Medicina', 3, TRUE), ('Bolile cardiovasculare', 'Dr.  
Radu Popa', 'Medicina', 2, TRUE);
```

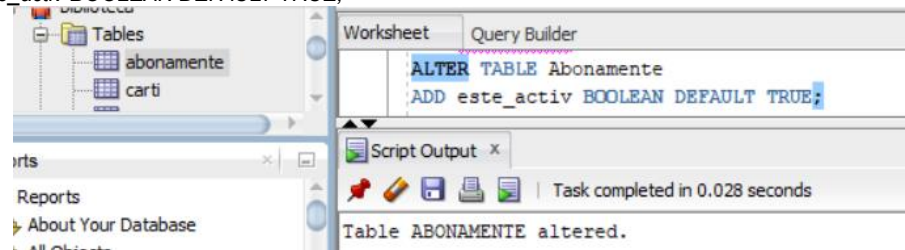


```
CREATE TABLE Abonamente (  
    abonament_id INT AUTO_INCREMENT PRIMARY KEY,  
    tip_abonament VARCHAR(50) NOT NULL, -- Denumirea pachetului. Ex.: Basic, Premium, VIP, Personalizat  
    descriere VARCHAR (150), -- Descriere scurtă a beneficiilor.
```

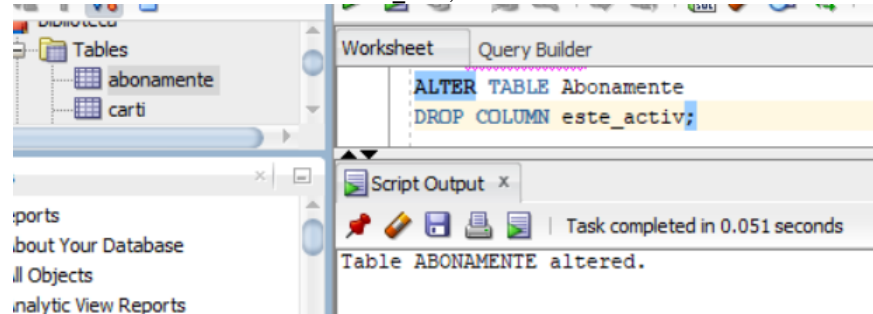
taxa_lunara DECIMAL(10, 2), -- Costul abonamentului lunar.
limita_carti INT, -- Număr maxim de cărți ce pot fi împrumutate simultan
limita_zile INT, -- Perioada maximă de împrumut pentru o carte.
reducere_servicii DECIMAL(5, 2), -- Discount pentru servicii de xerox, exprimat procentual
penalizare DECIMAL (5,2) -- Penalizare pentru fiecare zi de întârziere, exprimată procentual
);



ALTER TABLE Abonamente
 ADD este_activ BOOLEAN DEFAULT TRUE;



ALTER TABLE Abonamente DROP COLUMN este_activ;



INSERT INTO Abonamente (tip_abonament, descriere, taxa_lunara, limita_carti, limita_zile, reducere_servicii, penalizare)
 VALUES ('Basic', 'Abonament de bază pentru utilizatori casual', 20.00, 2, 14, 5.00, 0.50);

INSERT INTO Abonamente (tip_abonament, descriere, taxa_lunara, limita_carti, limita_zile, reducere_servicii, penalizare)
 VALUES ('Premium', 'Acces extins cu beneficii suplimentare', 50.00, 5, 21, 10.00, 0.30);

INSERT INTO Abonamente (tip_abonament, descriere, taxa_lunara, limita_carti, limita_zile, reducere_servicii, penalizare)
 VALUES ('VIP', 'Pachet exclusiv pentru utilizatori frecvenți', 100.00, 10, 30, 20.00, 0.20);

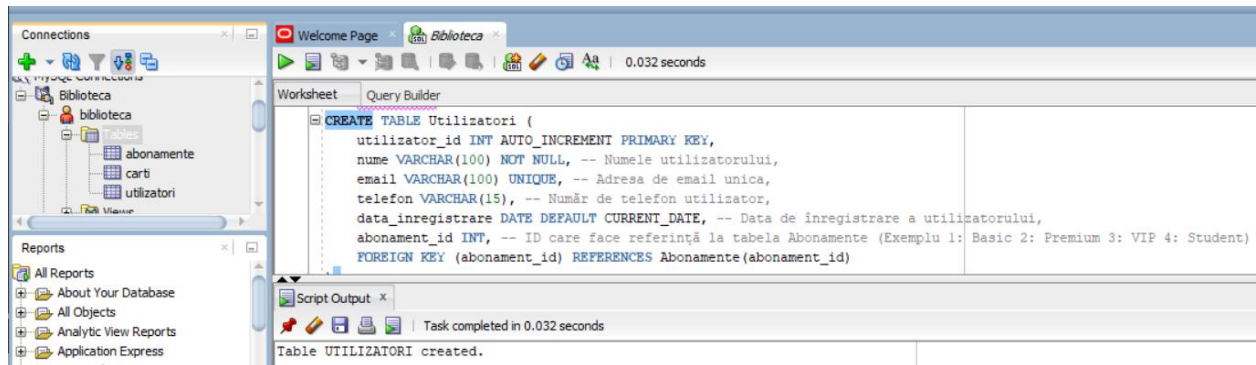
INSERT INTO Abonamente (tip_abonament, descriere, taxa_lunara, limita_carti, limita_zile, reducere_servicii, penalizare)
 VALUES ('Student', 'Special pentru studenți cu acces la materiale educaționale', 15.00, 3, 21, 15.00, 0.40);



```

CREATE TABLE Utilizatori (
    utilizator_id INT AUTO_INCREMENT PRIMARY KEY,
    nume VARCHAR(100) NOT NULL, -- Numele utilizatorului,
    email VARCHAR(100) UNIQUE, -- Adresa de email unica,
    telefon VARCHAR(15), -- Număr de telefon utilizator,
    data_inregistrare DATE DEFAULT CURRENT_DATE, -- Data de înregistrare a utilizatorului,
    abonament_id INT, -- ID care face referință la tabela Abonamente (Exemplu 1: Basic 2: Premium 3: VIP 4: Student)
    FOREIGN KEY (abonament_id) REFERENCES Abonamente(abonament_id)
);

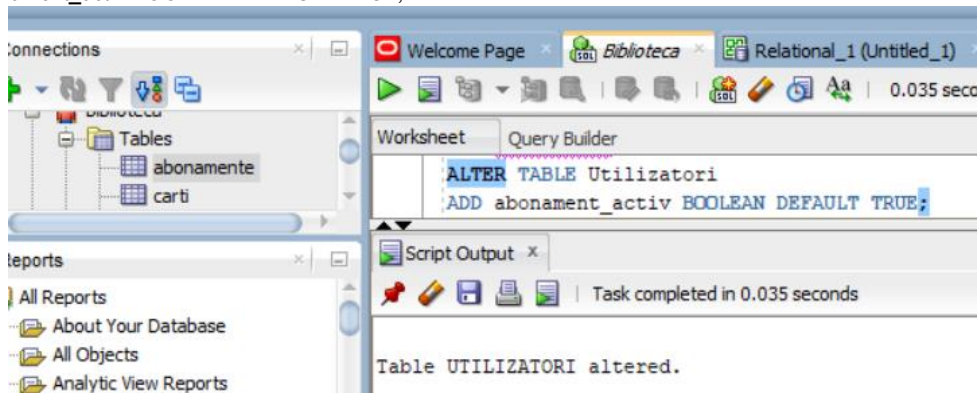
```



```

ALTER TABLE Utilizatori
ADD abonament_activ BOOLEAN DEFAULT TRUE;

```



```

INSERT INTO Utilizatori (nume, email, telefon, data_inregistrare, abonament_id, abonament_activ)
VALUES ('Andrei Popescu', 'andrei.popescu@example.com', '0712345678', '2023-01-10', 1, TRUE);
INSERT INTO Utilizatori (nume, email, telefon, data_inregistrare, abonament_id, abonament_activ)
VALUES ('Maria Ionescu', 'maria.ionescu@example.com', '0722345678', '2023-02-15', 2, TRUE);
INSERT INTO Utilizatori (nume, email, telefon, data_inregistrare, abonament_id, abonament_activ)
VALUES ('Alex Dumitrescu', 'alex.dumitrescu@example.com', '0732345678', '2023-03-20', 3, TRUE);
INSERT INTO Utilizatori (nume, email, telefon, data_inregistrare, abonament_id, abonament_activ)
VALUES ('Ioana Marinescu', 'ioana.marinescu@example.com', '0742345678', '2023-04-01', 4, TRUE);
INSERT INTO Utilizatori (nume, email, telefon, data_inregistrare, abonament_id, abonament_activ)
VALUES ('Cristian Stan', 'cristian.stan@example.com', '0752345678', '2023-05-12', 1, TRUE);

```



```

INSERT INTO Utilizatori (nume, email, telefon, data_inregistrare, abonament_id, abonament_activ)
VALUES ('Andrei Popescu', 'andrei.popescu@example.com', '0712345678', '2023-01-10', 1, TRUE);
INSERT INTO Utilizatori (nume, email, telefon, data_inregistrare, abonament_id, abonament_activ)
VALUES ('Maria Ionescu', 'maria.ionescu@example.com', '0722345678', '2023-02-15', 2, TRUE);
INSERT INTO Utilizatori (nume, email, telefon, data_inregistrare, abonament_id, abonament_activ)
VALUES ('Alex Dumitrescu', 'alex.dumitrescu@example.com', '0732345678', '2023-03-20', 3, TRUE);
INSERT INTO Utilizatori (nume, email, telefon, data_inregistrare, abonament_id, abonament_activ)
VALUES ('Ioana Marinescu', 'ioana.marinescu@example.com', '0742345678', '2023-04-01', 4, TRUE);
INSERT INTO Utilizatori (nume, email, telefon, data_inregistrare, abonament_id, abonament_activ)
VALUES ('Cristian Stan', 'cristian.stan@example.com', '0752345678', '2023-05-12', 1, TRUE);

```

Script Output x

Task completed in 0.042 seconds

```

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

```

UPDATE Utilizatori set nume = 'Maria Ghizdavin Dragu', email='maria.ghizdavin.dragu@example.com' where email='maria.ionescu@example.com'

0.028 seconds

Worksheet Query Builder

```

UPDATE Utilizatori set nume = 'Maria Ghizdavin Dragu',
email='maria.ghizdavin.dragu@example.com' where email='maria.ionescu@example.com'

```

Script Output x

Task completed in 0.028 seconds

```

1 row updated.

```

DELETE from Utilizatori where email='cristian.stan@example.com';

Worksheet Query Builder

```

DELETE from Utilizatori where email='cristian.stan@example.com';

```

Script Output x

Task completed in 0.02 seconds

```

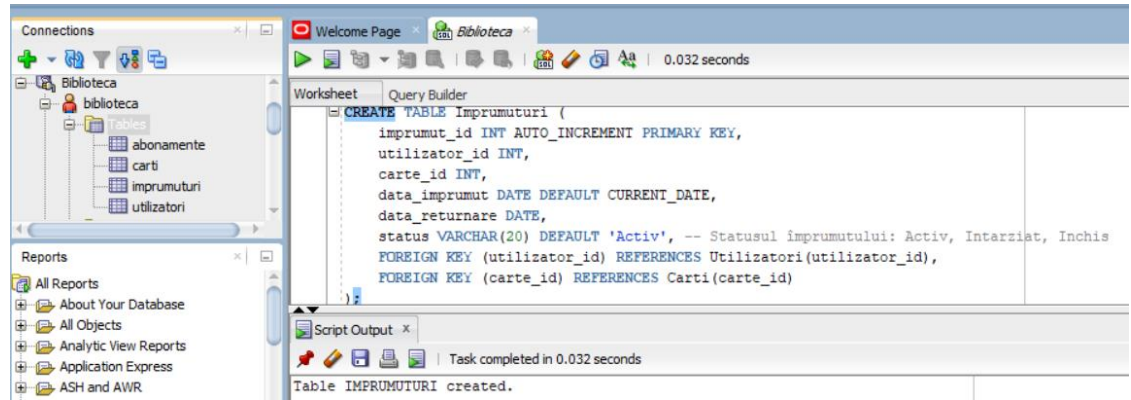
1 row deleted.

```

```

CREATE TABLE Imprumuturi (
    imprumut_id INT AUTO_INCREMENT PRIMARY KEY,
    utilizator_id INT,
    carte_id INT,
    data_imprumut DATE DEFAULT CURRENT_DATE,
    data_returnare DATE,
    status VARCHAR(20) DEFAULT 'Activ', -- Statusul împrumutului: Activ, Intarziat, Inchis
    FOREIGN KEY (utilizator_id) REFERENCES Utilizatori(utilizator_id),
    FOREIGN KEY (carte_id) REFERENCES Carti(carte_id)
);

```



```
INSERT INTO Imprumuturi (utilizator_id, carte_id, data_imprumut, data_returnare, status)
VALUES
```

```
(1, 1, '2025-01-12', NULL, 'Activ'), -- Basic: În termen (14 zile)
(2, 15, '2024-12-24', NULL, 'Activ'), -- Premium: În termen (30 zile)
(3, 7, '2024-11-28', NULL, 'Activ'), -- VIP: În termen (60 zile)
(4, 10, '2025-01-07', NULL, 'Activ'); -- Student: În termen (21 zile)
```

-- Împrumuturi cu status "Intarziat"

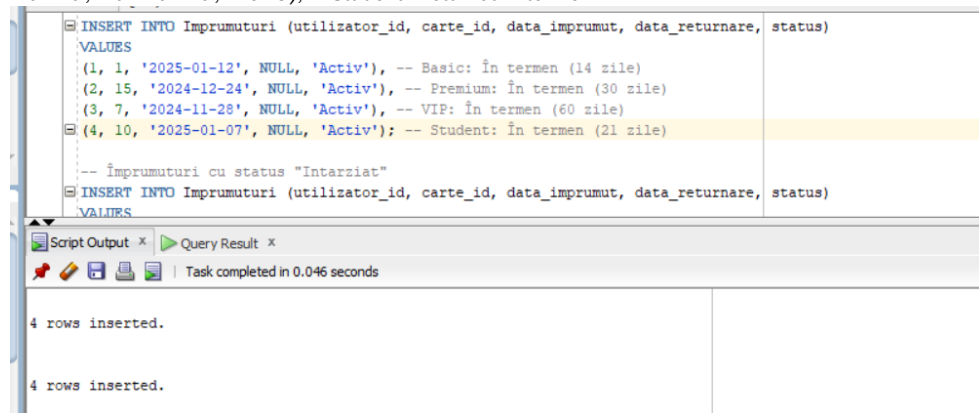
```
INSERT INTO Imprumuturi (utilizator_id, carte_id, data_imprumut, data_returnare, status)
VALUES
```

```
(1, 1, '2024-01-01', NULL, 'Intarziat'), -- Basic: Termen depășit (14 zile)
(2, 5, '2024-01-05', NULL, 'Intarziat'), -- Premium: Termen depășit (30 zile)
(3, 15, '2023-12-01', NULL, 'Intarziat'), -- VIP: Termen depășit (60 zile)
(4, 11, '2024-10-10', NULL, 'Intarziat'); -- Student: Termen depășit (21 zile)
```

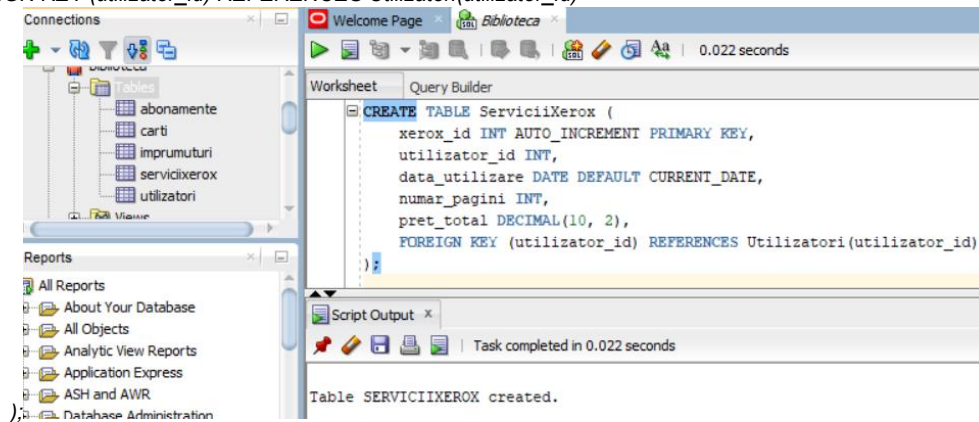
-- Împrumuturi cu status "Inchis"

```
INSERT INTO Imprumuturi (utilizator_id, carte_id, data_imprumut, data_returnare, status)
VALUES
```

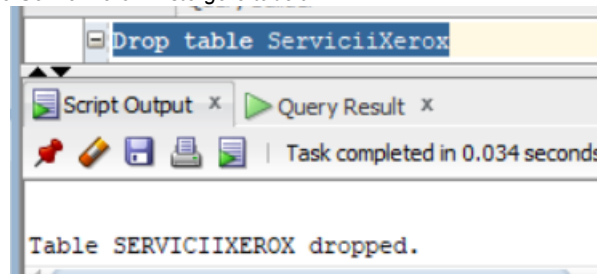
```
(1, 1, '2024-01-01', '2024-01-10', 'Inchis'), -- Basic: Returnat în termen
(2, 4, '2024-01-05', '2024-01-20', 'Inchis'), -- Premium: Returnat în termen
(3, 6, '2023-12-01', '2024-01-10', 'Inchis'), -- VIP: Returnat în termen
(4, 9, '2024-01-10', '2024-01-25', 'Inchis'); -- Student: Returnat în termen
```




```
CREATE TABLE ServiciiXerox (
    xerox_id INT AUTO_INCREMENT PRIMARY KEY,
    utilizator_id INT,
    data_utilizare DATE DEFAULT CURRENT_DATE,
    numar_pagini INT,
    pret_total DECIMAL(10, 2),
    FOREIGN KEY (utilizator_id) REFERENCES Utilizatori(utilizator_id)
```

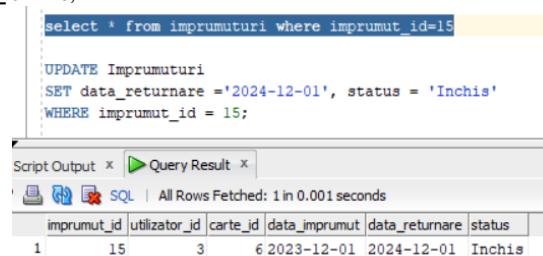


Drop table ServiciiXerox -- stergere tabela



Returnare carte

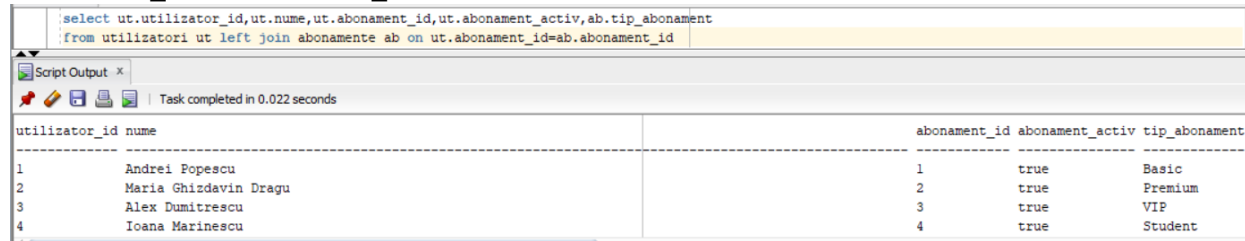
```
UPDATE Imprumuturi
SET data_returnare = '2024-12-01', status = 'Inchis'
WHERE imprumut_id = 15;
```



5. Interogari SQL

- Utilizatorii si tipul lor de abonament

select ut.utilizator_id,ut.abonament_id,ab.tip_abonament from utilizatori ut left join abonamente ab on ut.abonament_id=ab.abonament_id



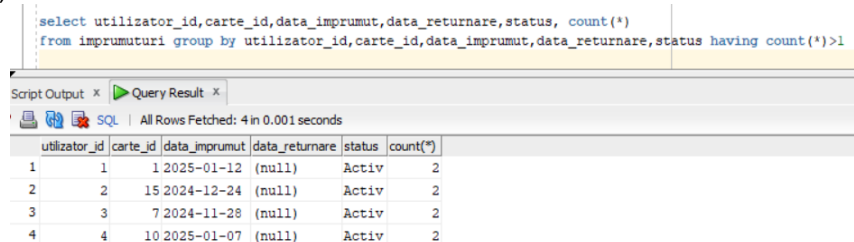
Script Output X

Task completed in 0.022 seconds

utilizator_id	nume	abonament_id	abonament_activ	tip_abonament
1	Andrei Popescu	1	true	Basic
2	Maria Ghizdavin Dragu	2	true	Premium
3	Alex Dumitrescu	3	true	VIP
4	Ioana Marinescu	4	true	Student

- Rânduri duplicate din Imprumuturi

select utilizator_id,carte_id,data_imprumut,data_returnare,status, count(*)
from imprumuturi group by utilizator_id,carte_id,data_imprumut,data_returnare,status having
count(*)>1



Script Output X

Query Result X

All Rows Fetched: 4 in 0.001 seconds

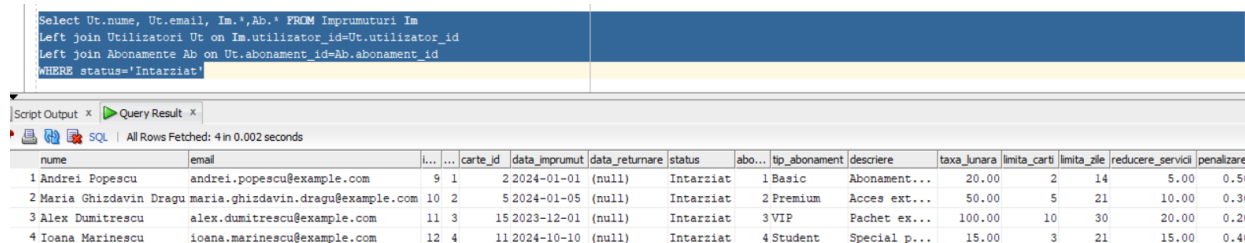
utilizator_id	carte_id	data_imprumut	data_returnare	status	count(*)
1	1	1 2025-01-12	(null)	Activ	2
2	2	15 2024-12-24	(null)	Activ	2
3	3	7 2024-11-28	(null)	Activ	2
4	4	10 2025-01-07	(null)	Activ	2

Corectie – Subinterogare

DELETE FROM Imprumuturi WHERE (utilizator_id, carte_id, data_imprumut, data_returnare, status) IN (
SELECT utilizator_id, carte_id, data_imprumut, data_returnare, status FROM (SELECT utilizator_id, carte_id,
data_imprumut, data_returnare, status, COUNT(*) FROM Imprumuturi GROUP BY utilizator_id, carte_id,
data_imprumut, data_returnare, status HAVING COUNT(*) > 1) AS subquery);

- Ce utilizatori au imprumuturi întârziate

Select Ut.nume, Ut.email, Im.*,Ab.* FROM Imprumuturi Im
Left join Utilizatori Ut on Im.utilizator_id=Ut.utilizator_id
Left join Abonamente Ab on Ut.abonament_id=Ab.abonament_id
WHERE status='Intarziat'



Script Output X

Query Result X

All Rows Fetched: 4 in 0.002 seconds

nume	email	i...	car	data_imprumut	data_returnare	status	abo...	tip_abonament	descriere	taxa_lunara	limita_carti	limita_zile	reducere_servici	penalizare
1 Andrei Popescu	andrei.popescu@example.com	9	1	2 2024-01-01	(null)	Intarziat	1 Basic	Abonament...		20.00	2	14	5.00	0.50
2 Maria Ghizdavin Dragu	maria.ghizdavin.dragu@example.com	10	2	5 2024-01-05	(null)	Intarziat	2 Premium	Acces ext...		50.00	5	21	10.00	0.30
3 Alex Dumitrescu	alex.dumitrescu@example.com	11	3	15 2023-12-01	(null)	Intarziat	3 VIP	Pachet ex...		100.00	10	30	20.00	0.20
4 Ioana Marinescu	ioana.marinescu@example.com	12	4	11 2024-10-10	(null)	Intarziat	4 Student	Special p...		15.00	3	21	15.00	0.40

- Cărți împrumutate în acest moment (status Activ)

SELECT i.imprumut_id, u.nume AS utilizator, c.titlu AS carte, i.data_imprumut, i.status
FROM Imprumuturi i
JOIN Utilizatori u ON i.utilizator_id = u.utilizator_id
JOIN Carti c ON i.car

<pre> SELECT i.imprumut_id, u.nume AS utilizator, c.titlu AS carte, i.data_imprumut, i.status FROM Imprumuturi i JOIN Utilizatori u ON i.utilizator_id = u.utilizator_id JOIN Carti c ON i.carte_id = c.carte_id WHERE i.status = 'Activ'; </pre>																																			
<div>Script Output x Query Result x</div> <div>All Rows Fetched: 4 in 0.002 seconds</div> <table> <tr> <th>imprumut_id</th><th>nume</th><th>titlu</th><th>data_imprumut</th><th>status</th><th></th></tr> <tr> <td>1</td><td>17 Andrei Popescu</td><td>Artă și frumosul în istorie</td><td>2025-01-12</td><td>Activ</td><td></td></tr> <tr> <td>2</td><td>18 Maria Ghizdavin Dragu</td><td>Bolile cardiovasculare</td><td>2024-12-24</td><td>Activ</td><td></td></tr> <tr> <td>3</td><td>19 Alex Dumitrescu</td><td>Introducere în programare</td><td>2024-11-28</td><td>Activ</td><td></td></tr> <tr> <td>4</td><td>20 Ioana Marinescu</td><td>Psihologia emoțiilor</td><td>2025-01-07</td><td>Activ</td><td></td></tr> </table>						imprumut_id	nume	titlu	data_imprumut	status		1	17 Andrei Popescu	Artă și frumosul în istorie	2025-01-12	Activ		2	18 Maria Ghizdavin Dragu	Bolile cardiovasculare	2024-12-24	Activ		3	19 Alex Dumitrescu	Introducere în programare	2024-11-28	Activ		4	20 Ioana Marinescu	Psihologia emoțiilor	2025-01-07	Activ	
imprumut_id	nume	titlu	data_imprumut	status																															
1	17 Andrei Popescu	Artă și frumosul în istorie	2025-01-12	Activ																															
2	18 Maria Ghizdavin Dragu	Bolile cardiovasculare	2024-12-24	Activ																															
3	19 Alex Dumitrescu	Introducere în programare	2024-11-28	Activ																															
4	20 Ioana Marinescu	Psihologia emoțiilor	2025-01-07	Activ																															

- Top utilizatori care au împrumutat cele mai multe cărți

```

SELECT u.utilizator_id, u.nume, COUNT(i.imprumut_id) AS numar_carti_imprumutate
FROM Imprumuturi i
JOIN Utilizatori u ON i.utilizator_id = u.utilizator_id
GROUP BY u.utilizator_id, u.nume
ORDER BY numar_carti_imprumutate DESC
LIMIT 10;

```

<pre> SELECT u.utilizator_id, u.nume, COUNT(i.imprumut_id) AS numar_carti_imprumutate FROM Imprumuturi i JOIN Utilizatori u ON i.utilizator_id = u.utilizator_id GROUP BY u.utilizator_id, u.nume ORDER BY numar_carti_imprumutate DESC LIMIT 10; </pre>																																			
<div>Script Output x Query Result x</div> <div>All Rows Fetched: 4 in 0.001 seconds</div> <table> <tr> <th>utilizator_id</th><th>nume</th><th>numar_carti_imprumutate</th><th></th><th></th><th></th></tr> <tr> <td>1</td><td>2 Maria Ghizdavin Dragu</td><td>3</td><td></td><td></td><td></td></tr> <tr> <td>2</td><td>4 Ioana Marinescu</td><td>3</td><td></td><td></td><td></td></tr> <tr> <td>3</td><td>1 Andrei Popescu</td><td>3</td><td></td><td></td><td></td></tr> <tr> <td>4</td><td>3 Alex Dumitrescu</td><td>3</td><td></td><td></td><td></td></tr> </table>						utilizator_id	nume	numar_carti_imprumutate				1	2 Maria Ghizdavin Dragu	3				2	4 Ioana Marinescu	3				3	1 Andrei Popescu	3				4	3 Alex Dumitrescu	3			
utilizator_id	nume	numar_carti_imprumutate																																	
1	2 Maria Ghizdavin Dragu	3																																	
2	4 Ioana Marinescu	3																																	
3	1 Andrei Popescu	3																																	
4	3 Alex Dumitrescu	3																																	

- Numărul de abonamente active pe categorii

```

SELECT a.tip_abonament, COUNT(u.utilizator_id) AS numar_abonati
FROM Abonamente a
LEFT JOIN Utilizatori u ON a.abonament_id = u.abonament_id
WHERE u.abonament_activ = TRUE
GROUP BY a.tip_abonament;

```

<pre> SELECT a.tip_abonament, COUNT(u.utilizator_id) AS numar_abonati FROM Abonamente a LEFT JOIN Utilizatori u ON a.abonament_id = u.abonament_id WHERE u.abonament_activ = TRUE GROUP BY a.tip_abonament; </pre>																																			
<div>Script Output x Query Result x</div> <div>All Rows Fetched: 4 in 0.001 seconds</div> <table> <tr> <th>tip_abonament</th><th>numar_abonati</th><th></th><th></th><th></th><th></th></tr> <tr> <td>1 Basic</td><td>1</td><td></td><td></td><td></td><td></td></tr> <tr> <td>2 Premium</td><td>1</td><td></td><td></td><td></td><td></td></tr> <tr> <td>3 Student</td><td>1</td><td></td><td></td><td></td><td></td></tr> <tr> <td>4 VIP</td><td>1</td><td></td><td></td><td></td><td></td></tr> </table>						tip_abonament	numar_abonati					1 Basic	1					2 Premium	1					3 Student	1					4 VIP	1				
tip_abonament	numar_abonati																																		
1 Basic	1																																		
2 Premium	1																																		
3 Student	1																																		
4 VIP	1																																		

- Rata de utilizare a fiecărei categorii de abonamente

```

SELECT a.tip_abonament, COUNT(i.imprumut_id) AS numar_total_imprumuturi
FROM Abonamente a
JOIN Utilizatori u ON a.abonament_id = u.abonament_id
JOIN Imprumuturi i ON u.utilizator_id = i.utilizator_id
GROUP BY a.tip_abonament
ORDER BY numar_total_imprumuturi DESC;

```

```

SELECT a.tip_abonament, COUNT(i.imprumut_id) AS numar_total_imprumuturi
FROM Abonamente a
JOIN Utilizatori u ON a.abonament_id = u.abonament_id
JOIN Imprumuturi i ON u.utilizator_id = i.utilizator_id
GROUP BY a.tip_abonament
ORDER BY numar_total_imprumuturi DESC;

```

tip_abonament	numar_total_imprumuturi
1 VIP	3
2 Premium	3
3 Basic	3
4 Student	3

- Cărțile cel mai des împrumutate

```

SELECT c.titlu, c.autor, COUNT(i.imprumut_id) AS numar_imprumuturi
FROM Carti c
JOIN Imprumuturi i ON c.carte_id = i.carte_id
GROUP BY c.carte_id, c.titlu, c.autor
ORDER BY numar_imprumuturi DESC
LIMIT 10;

```

```

SELECT c.titlu, c.autor, COUNT(i.imprumut_id) AS numar_imprumuturi
FROM Carti c
JOIN Imprumuturi i ON c.carte_id = i.carte_id
GROUP BY c.carte_id, c.titlu, c.autor
ORDER BY numar_imprumuturi DESC
LIMIT 10;

```

titlu	autor	numar_imprumuturi
1 Arta și frumosul în istorie	Ioan Popa	2
2 Bolile cardiovasculare	Dr. Radu Popa	2
3 Istoria artei moderne	Maria Ionescu	1
4 Marele Gatsby	F. Scott Fitzgerald	1
5 Crimă și pedeapsă	Fiodor Dostoievski	1
6 La răscruce de vânturi	Emily Brontë	1
7 Introducere în programare	Andrei Mihai	1
8 Bazele dezvoltării web	Cristian Stan	1
9 Psihologia emoțiilor	Maria Ionescu	1
10 Introducere în psihoterapie	George Petrescu	1

- Servicii Xerox utilizate de utilizatori într-un anumit interval de timp

```

SELECT u.ume AS utilizator, s.tip_serviciu, s.data_serviciu, s.numar_pagini, s.cost_total
FROM ServiciiXerox s
JOIN Utilizatori u ON s.utilizator_id = u.utilizator_id
WHERE s.data_serviciu BETWEEN '2024-01-01' AND '2024-12-31';

```

```

SELECT u.ume AS utilizator, s.data_utilizare, s.numar_pagini, s.pret_total
FROM ServiciiXerox s
JOIN Utilizatori u ON s.utilizator_id = u.utilizator_id
WHERE s.data_utilizare BETWEEN '2024-01-01' AND '2024-12-31';

```

nume	data_utilizare	numar_pagini	pret_total
------	----------------	--------------	------------

- Cărți care nu mai sunt disponibile

```

SELECT c.titlu, c.autor, c.exemplare_disponibile
FROM Carti c
WHERE c.exemplare_disponibile = 0;

```

```
SELECT c.carte_id,c.titlu, c.autor, c.exemplare_disponibile
FROM Carti c
WHERE c.exemplare_disponibile = 0;
```

carte_id	titlu	autor	exemplare_disponibile
1	15 Bolile cardiovasculare	Dr. Radu Popa	0

- Utilizatori care au depășit limita de zile permisă de abonament

```
SELECT u.num AS utilizator, c.titlu AS carte, a.tip_abonament,
a.limita_zile, DATEDIFF(CURDATE(), i.data_imprumut) AS zile_imprumut
FROM Imprumuturi i
JOIN Utilizatori u ON i.utilizator_id = u.utilizator_id
JOIN Carti c ON i.carte_id = c.carte_id
JOIN Abonamente a ON u.abonament_id = a.abonament_id
WHERE DATEDIFF(CURDATE(), i.data_imprumut) > a.limita_zile AND i.status = 'Activ';
```

```
SELECT u.num AS utilizator, c.titlu AS carte, a.tip_abonament,
a.limita_zile, DATEDIFF(CURDATE(), i.data_imprumut) AS zile_imprumut
FROM Imprumuturi i
JOIN Utilizatori u ON i.utilizator_id = u.utilizator_id
JOIN Carti c ON i.carte_id = c.carte_id
JOIN Abonamente a ON u.abonament_id = a.abonament_id
WHERE DATEDIFF(CURDATE(), i.data_imprumut) > a.limita_zile AND i.status = 'Activ';
```

num	titlu	tip_abonament	limita_zile	zile_imprumut
1 Alex Dumitrescu	Introducere în programare VIP		30	45

- Venituri totale generate din serviciile de xerox

```
SELECT SUM(s.cost_total) AS venituri_totale
FROM ServiciiXerox s;
```

```
SELECT SUM(s.pret_total) AS venituri_totale
FROM ServiciiXerox s;
```

venituri_totale

- Numărul de utilizatori pentru fiecare tip de abonament (activi/inactivi)

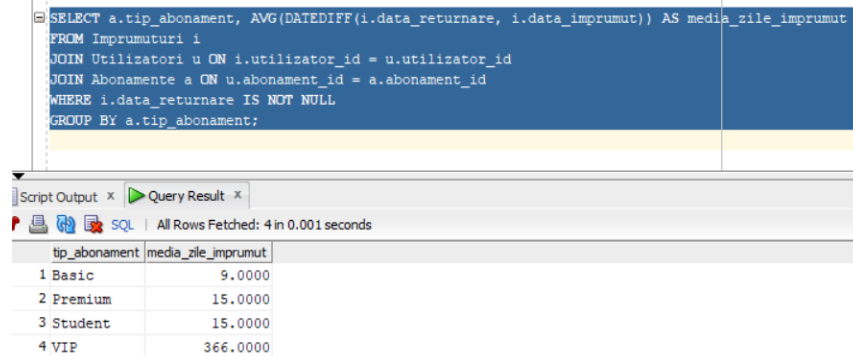
```
SELECT a.tip_abonament,
COUNT(CASE WHEN u.abonament_activ = TRUE THEN 1 ELSE NULL END) AS activi,
COUNT(CASE WHEN u.abonament_activ = FALSE THEN 1 ELSE NULL END) AS inactivi
FROM Abonamente a
LEFT JOIN Utilizatori u ON a.abonament_id = u.abonament_id
GROUP BY a.tip_abonament;
```

```
SELECT a.tip_abonament,
COUNT(CASE WHEN u.abonament_activ = TRUE THEN 1 ELSE NULL END) AS activi,
COUNT(CASE WHEN u.abonament_activ = FALSE THEN 1 ELSE NULL END) AS inactivi
FROM Abonamente a
LEFT JOIN Utilizatori u ON a.abonament_id = u.abonament_id
GROUP BY a.tip_abonament;
```

tip_abonament	activi	inactivi
1 Basic	1	0
2 Premium	1	0
3 Student	1	0
4 VIP	1	0

- Media zilelor de împrumut pentru fiecare tip de abonament

```
SELECT a.tip_abonament, AVG(DATEDIFF(i.data_returnare, i.data_imprumut)) AS media_zile_imprumut
FROM Imprumuturi i
JOIN Utilizatori u ON i.utilizator_id = u.utilizator_id
JOIN Abonamente a ON u.abonament_id = a.abonament_id
WHERE i.data_returnare IS NOT NULL
GROUP BY a.tip_abonament;
```

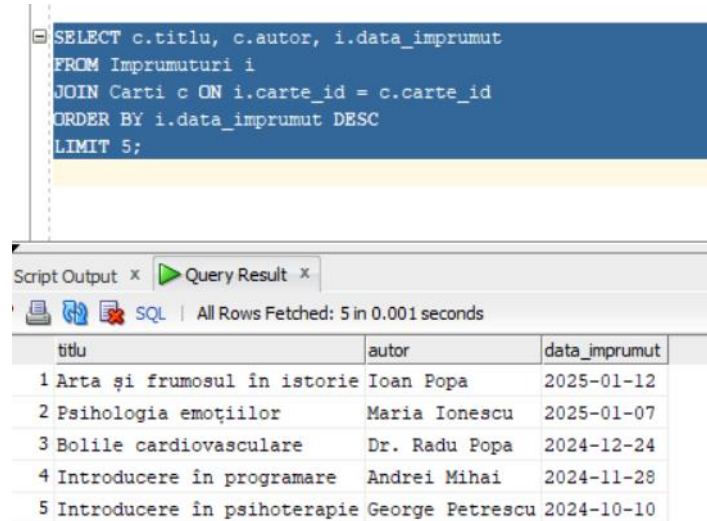


The screenshot shows the SQL query in a text editor and its results in a table. The query calculates the average number of days for which each subscription type is borrowed. The results table has two columns: 'tip_abonament' and 'media_zile_imprumut'.

tip_abonament	media_zile_imprumut
1 Basic	9.0000
2 Premium	15.0000
3 Student	15.0000
4 VIP	366.0000

- Cărțile care au fost împrumutate cel mai recent

```
SELECT c.titlu, c.autor, i.data_imprumut
FROM Imprumuturi i
JOIN Carti c ON i.carte_id = c.carte_id
ORDER BY i.data_imprumut DESC
LIMIT 5;
```

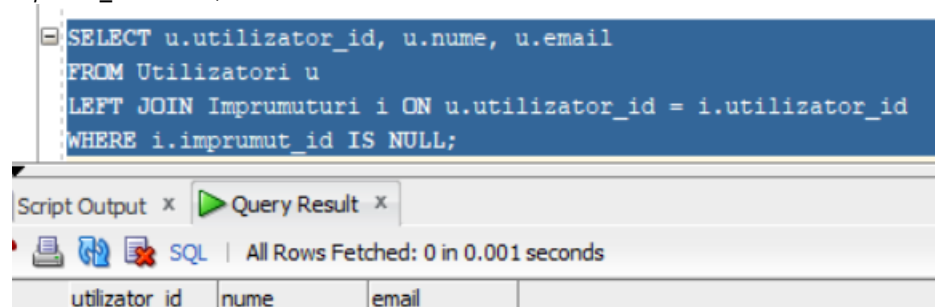


The screenshot shows the SQL query in a text editor and its results in a table. The query lists the 5 most recently borrowed books. The results table has three columns: 'titlu', 'autor', and 'data_imprumut'.

titlu	autor	data_imprumut
1 Arta și frumosul în istorie	Ioan Popa	2025-01-12
2 Psihologia emoțiilor	Maria Ionescu	2025-01-07
3 Bolile cardiovasculare	Dr. Radu Popa	2024-12-24
4 Introducere în programare	Andrei Mihai	2024-11-28
5 Introducere în psihoterapie	George Petrescu	2024-10-10

- Utilizatori care nu au împrumutat niciodată cărți

```
SELECT u.utilizator_id, u.num, u.email
FROM Utilizatori u
LEFT JOIN Imprumuturi i ON u.utilizator_id = i.utilizator_id
WHERE i.imprumut_id IS NULL;
```



The screenshot shows the SQL query in a text editor and its results in a table. The query lists users who have never borrowed a book. The results table has three columns: 'utilizator_id', 'nume', and 'email'. The 'All Rows Fetched' count is 0.

utilizator_id	nume	email
---------------	------	-------

- Prețul final al unui serviciu în funcție de reducerea din abonament

```
SELECT
    sx.xerox_id,
    u.ume AS utilizator,
    a.tip_abonament,
    sx.numar_pagini,
    sx.pret_total AS pret_initial,
    a.reducere_servicii AS reducere_procentuala,
    ROUND(sx.pret_total * (1 - a.reducere_servicii / 100), 2) AS pret_final
FROM ServiciiXerox sx
JOIN Utilizatori u ON sx.utilizator_id = u.utilizator_id
JOIN Abonamente a ON u.abonament_id = a.abonament_id;
```

xerox_id	nume	tip_abonament	numar_pagini	pret_total	reducere_se...	pret_final

- Inițialele utilizatorilor împreună cu informațiile despre abonament

```
SELECT
    u.utilizator_id,
    CONCAT(SUBSTR(u.ume, 1, 1), ' ', SUBSTR(SUBSTR(u.ume, LOCATE(' ', u.ume) + 1), 1, 1), ' ') AS initiale,
    a.tip_abonament,
    u.email
FROM Utilizatori u
JOIN Abonamente a ON u.abonament_id = a.abonament_id;
```

utilizator_id	initiale	tip_abonament	email
1	1 A.P.	Basic	andrei.popescu@example.com
2	2 M.G.	Premium	maria.ghizdavin.dragu@example.com
3	3 A.D.	VIP	alex.dumitrescu@example.com
4	4 I.M.	Student	ioana.marinescu@example.com

- Utilizatori care nu au împrumutat cărți

```
SELECT utilizator_id, nume
FROM Utilizatori

MINUS

SELECT utilizator_id, nume
FROM Utilizatori
JOIN Imprumuturi ON Utilizatori.utilizator_id = Imprumuturi.utilizator_id;
```