

שם: ארי שישפורטיש
ת.ז. 319306684
קורס: מיני פרוייקט לבסיסי נתונים

דו"ח על בסיס נתונים- הלוואות בנק

פרוייקט העוסק בתחום הלוואות של הבנק. עוסק בנתונים והפרטים של הלווים ותאריכי ההלוואה.

DSD

- **Loan Table**
 - Loan_ID (PK)
 - LoanStartDate
 - Borrower_ID (FK)
- **PaymentSchedule Table**
 - Schedule_ID (PK)
 - DueDate
 - Loan_ID (FK)
- **Borrower Table**
 - Borrower_ID (PK)
 - DateOfBirth

Relationships:

- **Loan** (Loan_ID) → **Borrower** (Borrower_ID)
- **Loan** (Loan_ID) → **PaymentSchedule** (Loan_ID)

SQL

```
CREATE TABLE Borrower
```

```
(  
  Borrower_ID INT NOT NULL,  
  DateOfBirth DATE NOT NULL,  
  PRIMARY KEY (Borrower_ID)  
);
```

```
CREATE TABLE Loan
```

```
(  
  Loan_ID INT NOT NULL,  
  LoanStartDate DATE NOT NULL,  
  Borrower_ID INT NOT NULL,  
  PRIMARY KEY (Loan_ID),  
  FOREIGN KEY (Borrower_ID) REFERENCES Borrower(Borrower_ID)  
);
```

```
CREATE TABLE PaymentSchedule
```

```
(  
  Schedule_ID INT NOT NULL,  
  DueDate DATE NOT NULL,  
  Loan_ID INT NOT NULL,  
  PRIMARY KEY (Schedule_ID),  
  FOREIGN KEY (Loan_ID) REFERENCES Loan(Loan_ID)  
);
```

קובץ createTables.sql

-- Create the Loan table

```
CREATE TABLE Loan (
    LoanID INT PRIMARY KEY,
    LoanStartDate DATE NOT NULL,
    BorrowerID INT,
    FOREIGN KEY (BorrowerID) REFERENCES Borrower(BorrowerID)
);
```

-- Create the PaymentSchedule table

```
CREATE TABLE PaymentSchedule (
    ScheduleID INT PRIMARY KEY,
    DueDate DATE NOT NULL,
    LoanID INT,
    FOREIGN KEY (LoanID) REFERENCES Loan(LoanID)
);
```

-- Create the Borrower table

```
CREATE TABLE Borrower (
    BorrowerID INT PRIMARY KEY,
    BorrowerName VARCHAR(100) NOT NULL,
    BorrowerAddress VARCHAR(255) NOT NULL
);
```

קובץ dropTables.sql

-- Drop the PaymentSchedule table first

DROP TABLE PaymentSchedule;

-- Drop the Loan table second

DROP TABLE Loan;

-- Drop the Borrower table last

DROP TABLE Borrower;

קובץ insertTables.sql

-- Insert data into Borrower table

INSERT INTO Borrower (BorrowerID, BorrowerName, BorrowerAddress) VALUES (1, 'John Doe', '123 Elm Street');

-- Repeat with at least 200 records

-- Insert data into Loan table

INSERT INTO Loan (LoanID, LoanStartDate, BorrowerID) VALUES (1, '2024-01-01', 1);

-- Repeat with at least 200 records

-- Insert data into PaymentSchedule table

INSERT INTO PaymentSchedule (ScheduleID, DueDate, LoanID) VALUES (1, '2024-02-01', 1);

-- Repeat with at least 200 records

selectAll.sql קובץ

-- Select all data from Borrower table

SELECT * FROM Borrower;

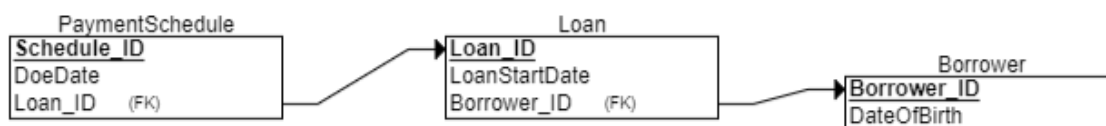
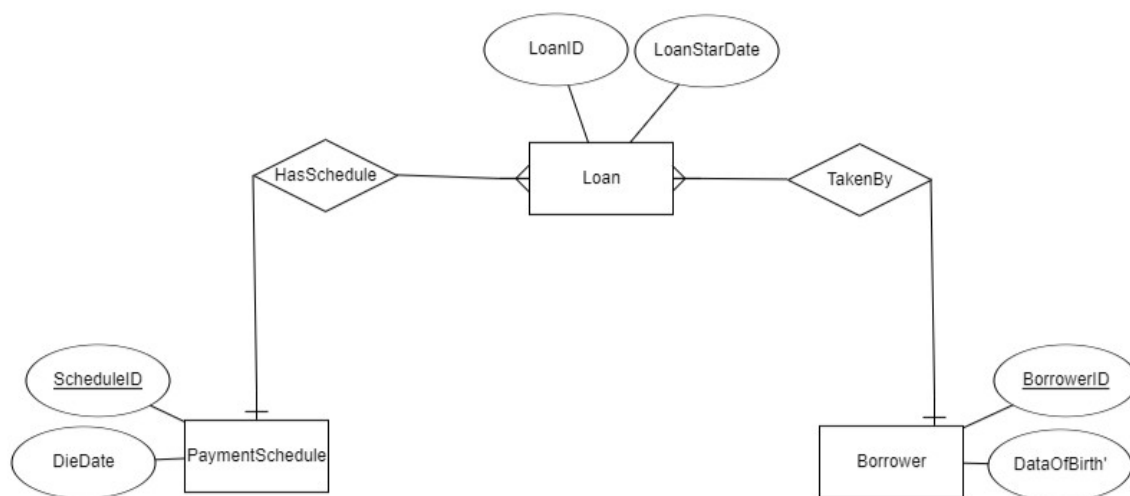
-- Select all data from Loan table

SELECT * FROM Loan;

-- Select all data from PaymentSchedule table

SELECT * FROM PaymentSchedule;

טבלאות ותרשימים



שלב 2

עכשיו נכתוב כאן סוגים שונים של שאילתות לוודא שמסד הנתונים שלנו עובד- וגם השאילתות מביא לנו את מה שרצוי. כמובן שיש צילומים של השאילתות, וגם תיעוד של הכנסת ערך לא תקין.

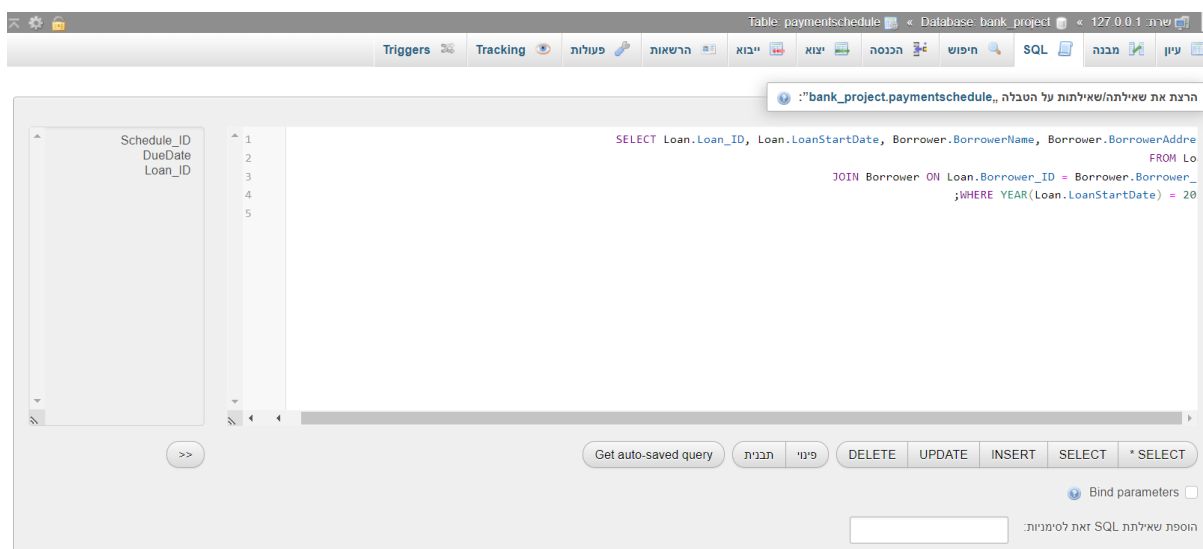
SELECT

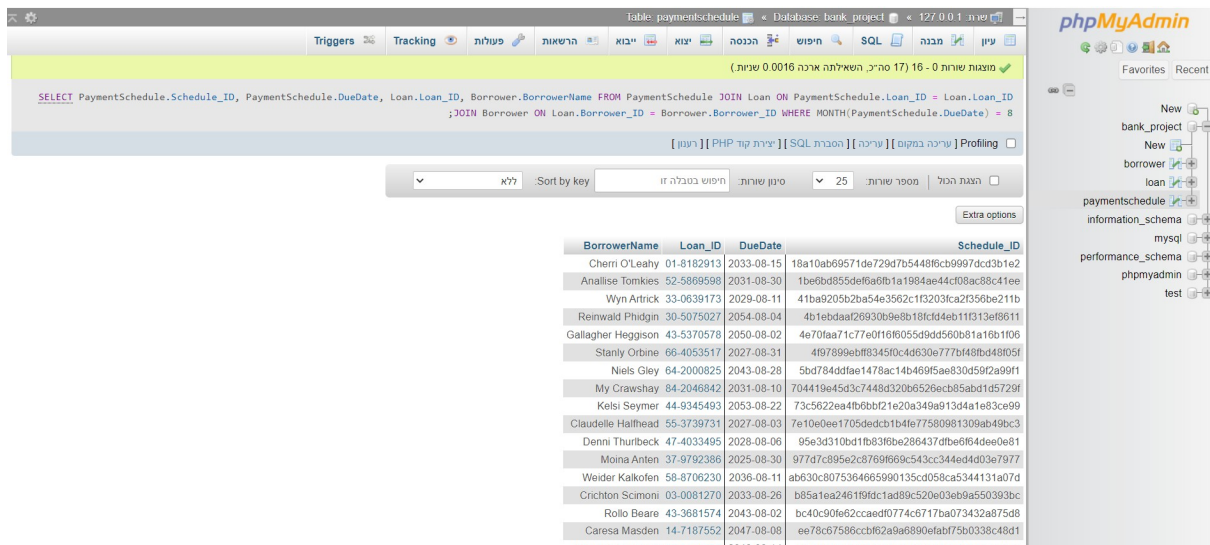
```
SELECT Loan.Loan_ID, Loan.LoanStartDate,
Borrower.BorrowerName, Borrower.BorrowerAddress
```

```
FROM Loan
```

```
JOIN Borrower ON Loan.Borrower_ID = Borrower.Borrower_ID
```

```
WHERE YEAR(Loan.LoanStartDate) = 2024;
```





The screenshot shows the phpMyAdmin interface with the 'paymentschedule' table selected. The SQL query is:

```
SELECT PaymentSchedule.Schedule_ID, PaymentSchedule.DueDate, Loan.Loan_ID, Borrower.BorrowerName FROM PaymentSchedule JOIN Loan ON PaymentSchedule.Loan_ID = Loan.Loan_ID ; JOIN Borrower ON Loan.Borrower_ID = Borrower.Borrower_ID WHERE MONTH(PaymentSchedule.DueDate) = 8
```

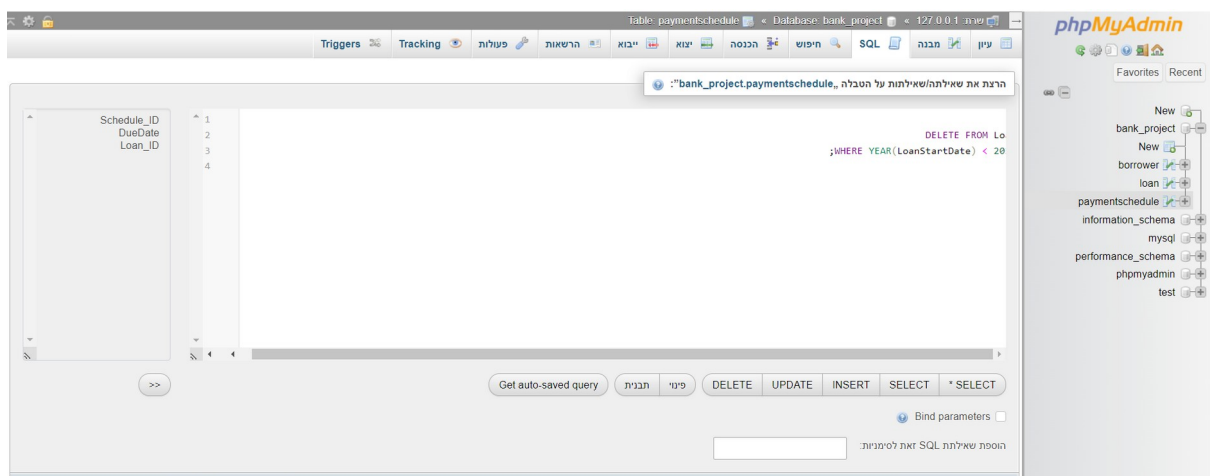
The results table shows the following data:

BorrowerName	Loan_ID	DueDate	Schedule_ID
Cherri O'Leahy	01-8182913	2033-08-15	18a10ab69571de729d7b5448f6cb9997dcd3b1e2
Analise Tomkies	52-5869598	2031-08-30	1be6bd855de6a6b1a1984ae44cf08ac88c41ee
Wyn Artrick	33-0639173	2029-08-11	41ba9205b2ba54e3562c1f3203fca2f356be211b
Reinwald Phidgin	30-5075027	2054-08-04	4b1ebdaaf29930b9e8b18fcd4eb11f313ef8611
Gallagher Heggison	43-5370578	2050-08-02	4e70faa71c77e0f16f6055d9dd560b81a16b1f06
Stanly Orbine	66-4053517	2027-08-31	4f97899ebf8345f0c4d630e777bf48bd48f05f
Niels Gley	64-2000825	2043-08-28	5bd784ddfae1478ac14b469f5ae830d592a99f1
My Crawshay	84-2046842	2031-08-10	704419e45d3c7448d320b6526ecb85abd1d5729f1
Kelsi Seymour	44-9345493	2053-08-22	73c5622ea4fb6bbf21e20a349a913d4a1e83ce99
Claudelle Halfhead	55-3739731	2027-08-03	7e10e0ee1705dedcb1b4fe77580981309ab49bc3
Denni Thurbeck	47-4033495	2028-08-06	95e3d310bd1fb83f6be286437dfbe6f64dee0e81
Mona Antien	37-9792386	2025-08-30	977d7c895e2c8769f669c543cc344ed4d03e7977
Weider Kalkofen	58-8706230	2036-08-11	ab630c8075364665990135cd058ca5344131a07d
Crichton Scimoni	03-0081270	2033-08-26	b85a1ea2461f9fdcf1ad89c520e03eb9a550393bc
Rollo Beare	43-3681574	2043-08-02	bc40c90fe62ccaedf0774c6717ba073432a875d8
Caresa Masden	14-7187552	2047-08-08	ee78c67586c6b62a9a6890efabf75b0338c48d1
Gonzain Frontales	88,0577612	2046,08,14	f08343b8a0f0b4d4d47d74757a7c7a7a77-f0b363644583

תמונה שהשאלות עובדות

DELETE

DELETE FROM Loan
WHERE YEAR(LoanStartDate) < 2020;

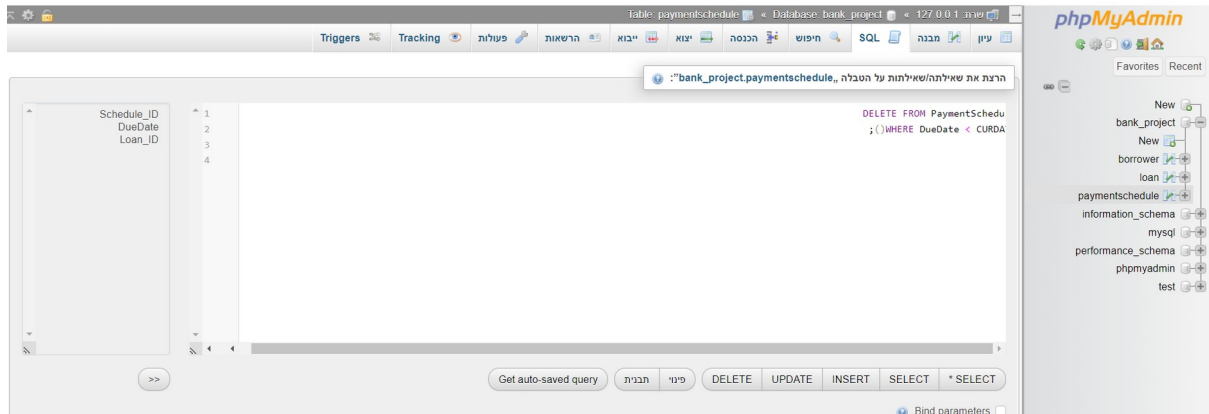


The screenshot shows the phpMyAdmin interface with the 'paymentschedule' table selected. The SQL query is:

```
DELETE FROM Lo  
WHERE YEAR(LoanStartDate) < 20
```

The interface includes a sidebar with the table structure, a main area for the query, and a bottom bar with buttons for 'DELETE', 'UPDATE', 'INSERT', 'SELECT', and '* SELECT'. The 'DELETE' button is highlighted.

DELETE FROM PaymentSchedule
WHERE DueDate < CURDATE();



UPDATE

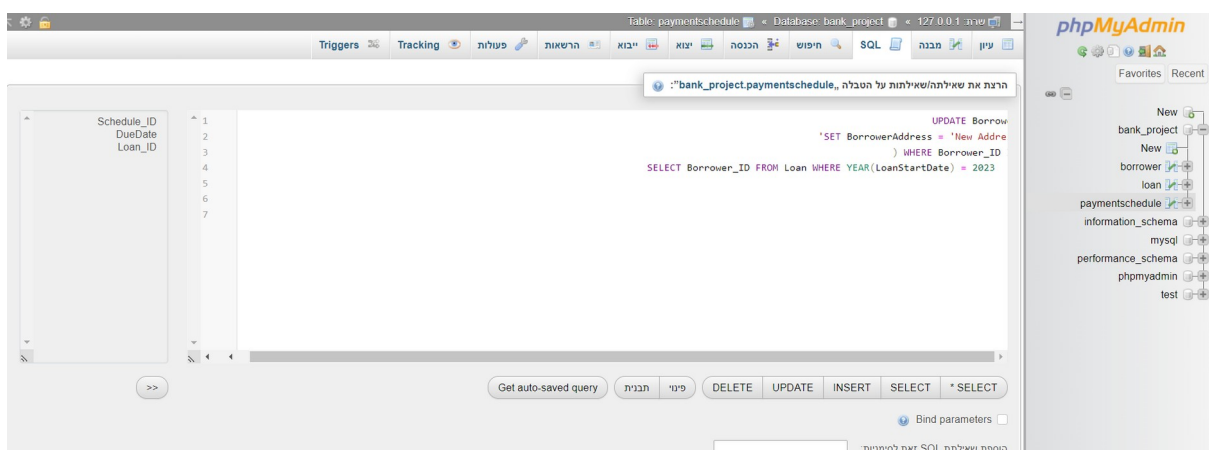
UPDATE Borrower

SET BorrowerAddress = 'New Address'

WHERE Borrower_ID IN (

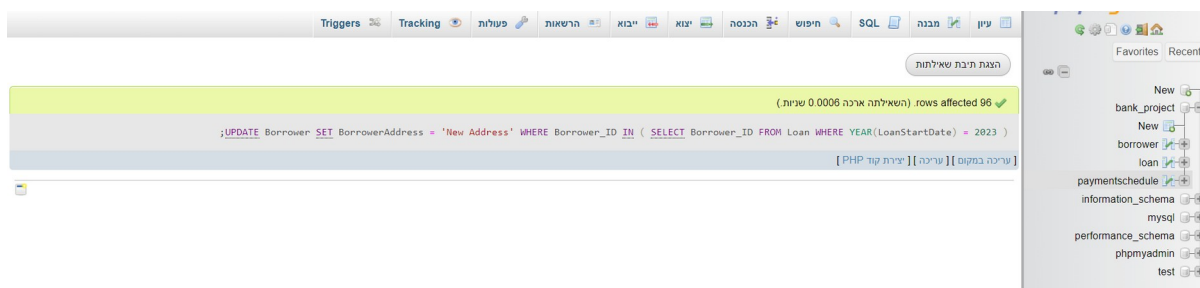
SELECT Borrower_ID FROM Loan WHERE
YEAR(LoanStartDate) = 2023

);

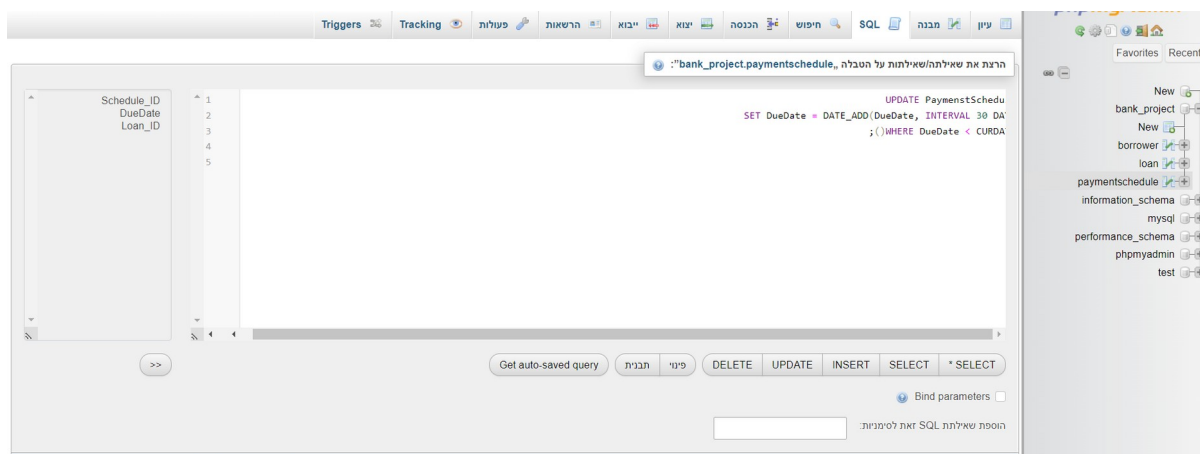




UPDATE PaymentSchedule
SET DueDate = DATE_ADD(DueDate, INTERVAL 30 DAY)
WHERE DueDate < CURDATE();



דוגמא לשגיאה בהכנסה ערך לא נכון



שלב 3

בשלב זה אנחנו מוספים בעצם פונקציות- אשר מקבלות ערך מסוים ומחזירות תשובה\ ערך אחר אשר עובר לפי סינון רצוי
 ופרצדורה עושה דברים יותר גמישים לטובה זה, ובכך בחרנו לייצר פונקציה אחת
 שתבדוק אם להלוואה שולם במלואם לפי לוח התשלומים
 :ויצרנו פרצדורה אשר מעדכנת את מצב ההלוואה על פי התשלומים שבוצעו

פונקציה:

```
CREATE FUNCTION CheckLoanPaid(p_Loan_ID VARCHAR(50))
  RETURNS BOOLEAN BEGIN DECLARE v_TotalPayments INT;
  DECLARE v_PaidPayments INT; -- ספירת התשלומים הכוללים SELECT
    COUNT(*) INTO v_TotalPayments FROM PaymentSchedule
    WHERE Loan_ID = p_Loan_ID; -- ספירת התשלומים ששולמו SELECT
    COUNT(*) INTO v_PaidPayments FROM Payments WHERE
    Loan_ID = p_Loan_ID AND PaymentStatus = 'Paid'; -- השוואת
    RETURN (v_TotalPayments =
      v_PaidPayments); END;
```

פרצדורה:

```
CREATE PROCEDURE UpdateLoanStatus(p_Loan_ID
  VARCHAR(50)) BEGIN DECLARE v_IsPaid BOOLEAN; -- קריאה
  CheckLoanPaid SET v_IsPaid =
  CheckLoanPaid(p_Loan_ID); -- עדכון מצב ההלוואה בהתאם לתוצאה IF
  v_IsPaid THEN UPDATE Loan SET LoanStatus = 'Paid' WHERE
  Loan_ID = p_Loan_ID; ELSE UPDATE Loan SET LoanStatus =
  'Unpaid' WHERE Loan_ID = p_Loan_ID; END IF; END;
```

בנוסף לפונקציות ופרצדורות: אנחנו גם נכתוב תוכנית ראשית אשר תפעיל את אותם
 פונקציות ופרצדורות.

תוכנית ראשית:

```
CREATE PROCEDURE ProcessLoanUpdates(p_Loan_ID
  VARCHAR(50)) BEGIN -- קריאה לפונקציה CheckLoanPaid
  UpdateLoanStatus CALL UpdateLoanStatus(p_Loan_ID); END;
```