בס"ד

**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,
  DoeDate 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;
```

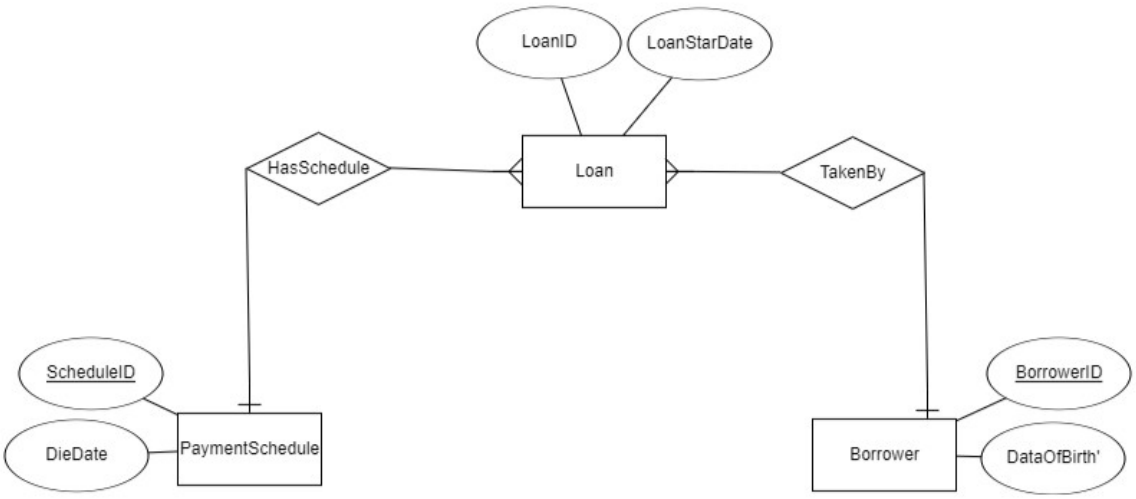
# קובץ insertTables.sql

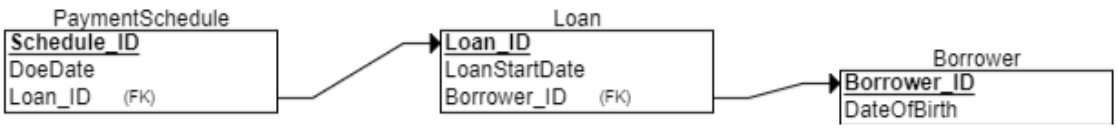
```
-- Insert data into Borrower table

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
```

# קובץ selectAll.sql

```
-- 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



בס"ד

# שלב 2

עכשיו נכתוב כאן סוגים שונים של שאילתות לוודא שמסד הנתונים שלנו עובד- וגם השאילתות מביא לנו את מה שרצוי.

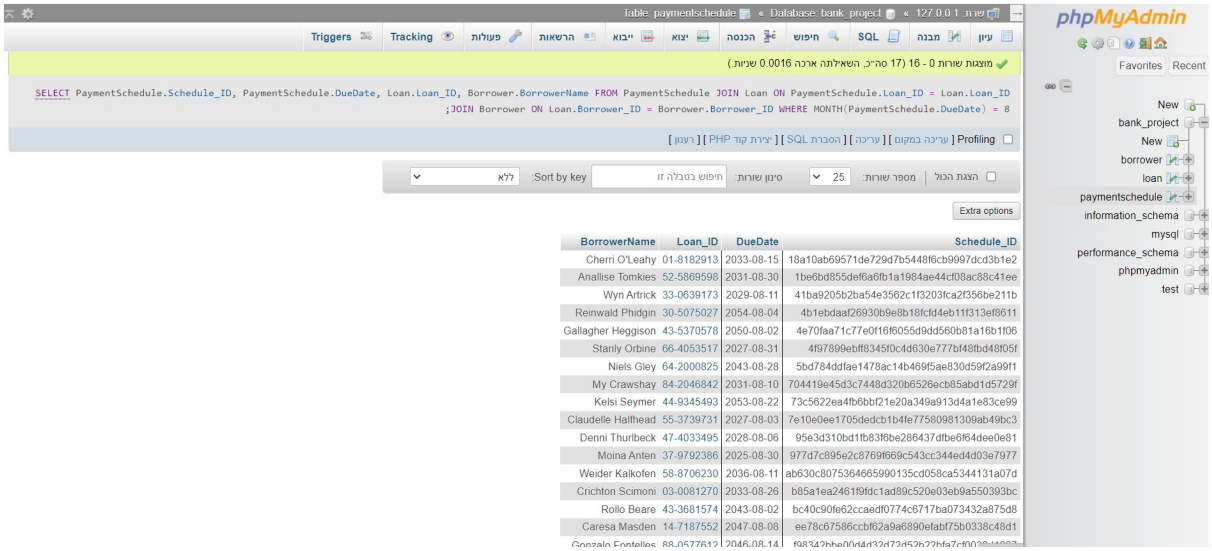
## SELECT

```
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;
```

---

```
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;
```

בס"ד




תמונה שהשאילתות עובדות

## **DELETE**

```
DELETE FROM Loan
WHERE YEAR(LoanStartDate) < 2020;
```

---

```
DELETE FROM PaymentSchedule
WHERE DueDate < CURDATE();
```

## **UPDATE**

```
UPDATE Borrower
SET BorrowerAddress = 'New Address'
WHERE Borrower_ID IN (
```

בס"ד

```
SELECT Borrower_ID FROM Loan WHERE
YEAR(LoanStartDate) = 2023
);
```

---

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