

מודל נתונים עבור אגף פיננסים בבנק

שמות מגישות: דוד אוהב ציון ומרקוס צ'אמה.

המערכת: בנק

היחידה הנבחרת: ישנם שלושה אגפים בבנק: לקוחות, פיננסים, משאבי אנוש. אנו נתמקד באגף פיננסים.

1. מבוא

מודל נתונים זה מתאר את המבנה של בסיס נתונים עבור אגף פיננסים בבנק. המודל כולל מספר ישויות, ביניהן חשבונות, טרנזקציות, הלוואות, כרטיסי אשראי, שיקים ופקדונות. הקשרים בין הישויות מוגדרים כ-1:Many.

מטרות המערכת:

- ניהול נתוני חשבונות
- מעקב אחר טרנזקציות פיננסיות
- ניהול הלוואות וכרטיסי אשראי
- מעקב אחר תשלומים באמצעות שיקים
- ניהול פקדונות

ישויות:

- חשבונות (Accounts):
 - תכונות:
 - מזהה חשבון (Account ID): מפתח ראשוני ייחודי לכל חשבון.
 - מזהה לקוח (Customer ID): מפתח זר המקשר לחשבון ללקוח שלו.
 - סוג חשבון (Account Type): סוג החשבון (חיסכון, עובר ושב, פקדון).
 - יתרת חשבון (Balance): סכום הכסף הזמין בחשבון.
 - תאריך פתיחת חשבון (Account Opening Date): התאריך שבו נפתח החשבון.
 - מפתח זר : Account ID
 - טרנזקציות (Transactions):
 - תכונות:
 - מזהה טרנזקציה (Transaction ID): מפתח ראשוני ייחודי לכל טרנזקציה.
 - מזהה חשבון (Account ID): מפתח זר המקשר את הטרנזקציה לחשבון.
 - תאריך טרנזקציה (Transaction Date): התאריך שבו בוצעה הטרנזקציה.
 - סכום (Amount): סכום הכסף שהועבר בטרנזקציה.
 - מפתח זר : Account ID

- הלוואות (Loans):

- תכונות:

- **מזהה הלוואה (Loan ID):** מפתח ראשוני ייחודי לכל הלוואה.
- סכום הלוואה (Loan Amount): סכום הכסף שהלווה הבנק ללקוח.
- ריבית (Interest Rate): שיעור הריבית על ההלוואה.
- תאריך התחלה (Start Date): התאריך שבו החלה ההלוואה.
- תאריך סיום (End Date): התאריך שבו תסתיים ההלוואה.
- מפתח זר : Account ID

- כרטיסי אשראי (Credit Cards):

- תכונות:

- **מזהה כרטיס (Card ID):** מפתח ראשוני ייחודי לכל כרטיס.
- מספר כרטיס (Card Number): מספר הכרטיס הייחודי.
- סוג כרטיס (Card Type): סוג הכרטיס (Debit, Regular).
- תאריך תפוגה (Expiration Date): התאריך שבו הכרטיס פג תוקפו.
- מסגרת אשראי (Credit Limit): סכום הכסף המקסימלי שניתן להוציא באמצעות הכרטיס.
- מפתח זר : Account ID

- שיקים (Checks):

- תכונות:

- **מזהה שיק (Check ID):** מפתח ראשוני ייחודי לכל שיק.
- מזהה חשבון (Account ID): מפתח זר המקשר את השיק לחשבון.
- מספר שיק (Check Number): מספר השיק הייחודי.
- סכום (Amount): סכום הכסף ששולם באמצעות השיק.
- תאריך הנפקה (Issue Date): התאריך שבו הונפק השיק.
- תאריך פרעון (Clearing Date): התאריך שבו השיק אמור להיפרע.
- מפתח זר : Account ID

- פקדונות (Deposits):

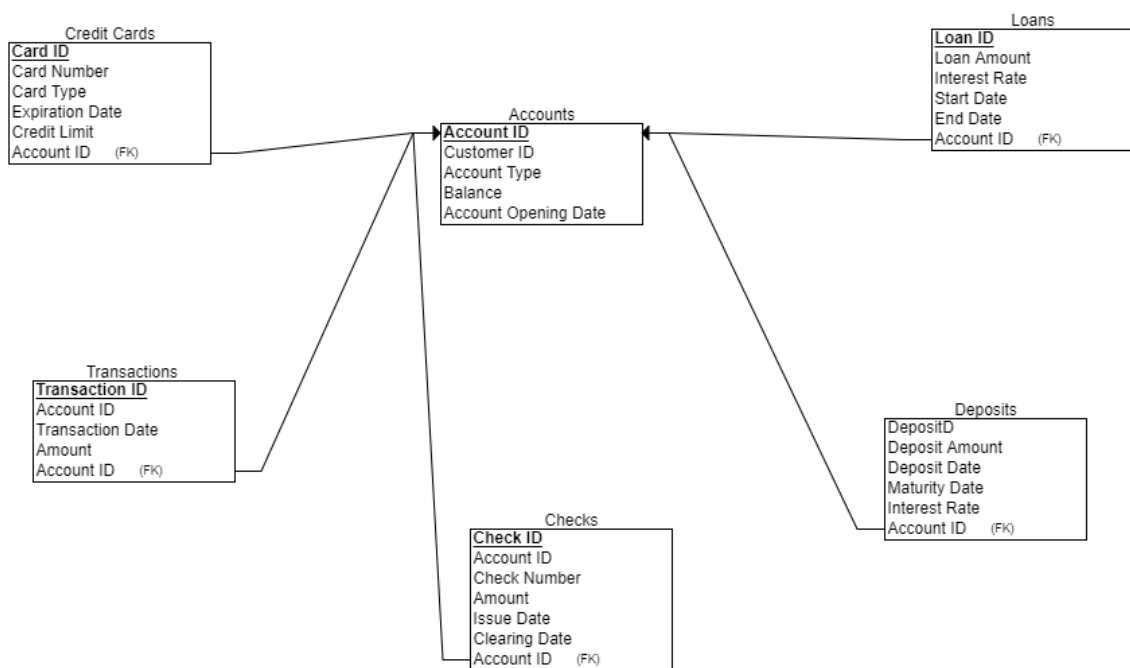
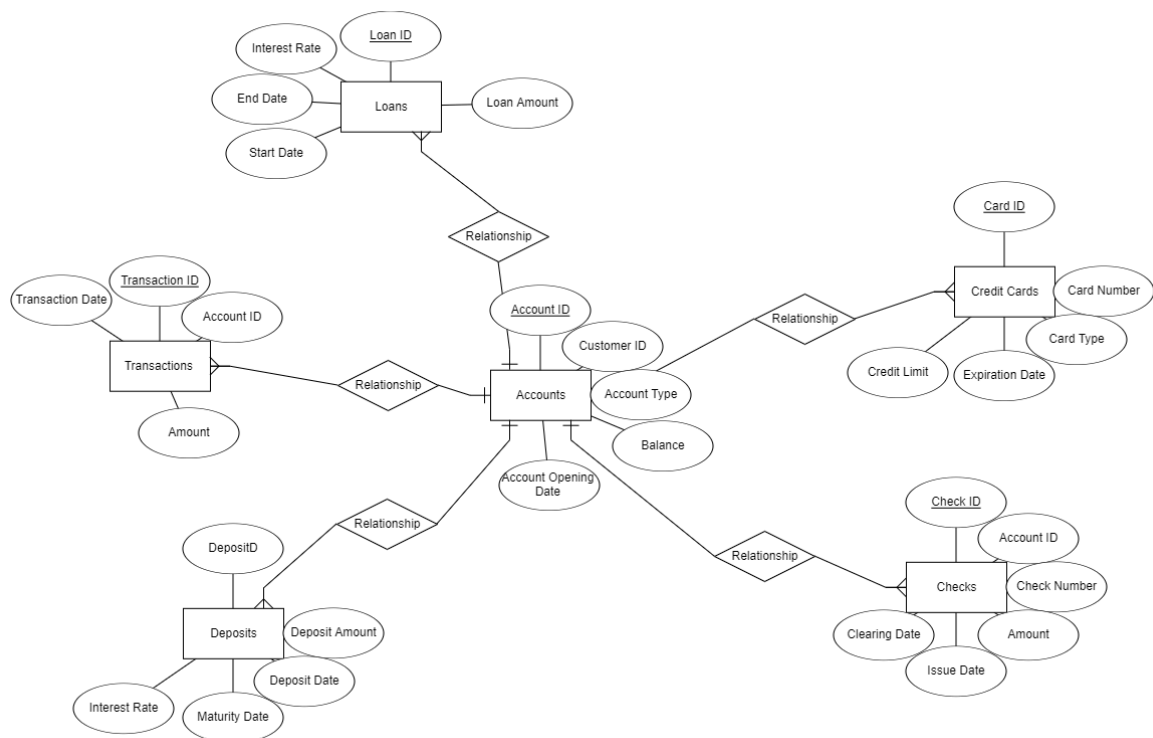
- תכונות:

- **מזהה פקדון (DepositID):** מפתח ראשוני ייחודי לכל פקדון.
- סכום פקדון (Deposit Amount): סכום הכסף שהופקד בחשבון.
- תאריך פקדון (Deposit Date): התאריך שבו בוצע הפקדון.
- תאריך פדיון (Maturity Date): התאריך שבו ניתן לפדות את הפקדון.
- ריבית (Interest Rate): שיעור הריבית על הפקדון.
- מפתח זר : Account ID

קשרים: כל הקשרים מוגדרים כ-Many:1

- לקוחות-חשבונות: לקוח אחד יכול להחזיק מספר חשבונות. אבל לכל חשבון לקוח אחד
- חשבונות-טרנזקציות: חשבון אחד יכול להכיל מספר טרנזקציות. אך כל טרנזקציה משויכת לחשבון אחד.
- לקוחות-הלוואות: לקוח אחד יכול לקחת מספר הלוואות. אך כל הלוואה שייכת לחשבון אחד

- לקוחות-כרטיסי אשראי: לקוח אחד יכול להחזיק מספר כרטיסי אשראי. אך כל כרטיס אשראי שייך לחשבון אחד
- חשבונות-שיקים: חשבון אחד יכול להוציא מספר שיקים. אך על שייך שייך לחשבון אחד.
- לקוחות-פקדונות: לקוח אחד יכול להחזיק מספר פקדונות. אך כל פקדון שייך לחשבון אחד



כל הטבלאות הינם ברמה של 3NF ואין צורך בנרמול, נוכיח זאת:

כל הטבלאות עומדות ביחס של 1NF מפני שכל השדות הם אטומיים.

כל הטבלאות עומדות ביחס של 2NF בכל הטבלאות המפתח הינו רק שדה אחד, ולכן לא תתכן תלות בחלק מן המפתח אלא בכולו.

כל הטבלאות עומדות ביחס של 3NF : אין קשר בין השדות השונים, הקשר היחיד הוא ע"י שדה המפתח.

יצירת הטבלאות:

createTables.sql

dropTables.sql

selectAll.sql

main

insertTables.s

SQL

Output

Statistics

```
CREATE TABLE Accounts
(
    Account_ID INT NOT NULL,
    Customer_ID INT NOT NULL,
    Account_Type varchar2(10),
    Balance INT NOT NULL,
    Account_Opening_Date DATE NOT NULL,
    PRIMARY KEY (Account_ID)
);

CREATE TABLE Transactions
(
    Transaction_ID INT NOT NULL,
    Amount DATE NOT NULL,
    Account_ID INT NOT NULL,
    Transaction_Date DATE,
    PRIMARY KEY (Transaction_ID),
    FOREIGN KEY (Account_ID) REFERENCES Accounts (Account_ID)
);

CREATE TABLE Loans
(
    Loan_ID INT NOT NULL,
    Customer_ID INT NOT NULL,
    Loan_Amount INT NOT NULL,
    Interest_Rate INT NOT NULL,
    Start_Date DATE NOT NULL,
    End_Date DATE NOT NULL,
    Account ID INT NOT NULL,
```

	CHECK_ID	CHECK_NUMBER	AMOUNT	ISSUE_DATE	CLEARING_DATE	ACCOUNT_ID
1	401	98765	200	18/05/2024	21/05/2024	1
2	402	12345	150	17/05/2024	20/05/2024	2
3	403	67890	250	19/05/2024	22/05/2024	3
4	404	23456	300	20/05/2024	23/05/2024	4
5	405	34567	450	21/05/2024	24/05/2024	5
6	406	45678	500	22/05/2024	25/05/2024	6
7	407	56789	550	23/05/2024	26/05/2024	7
8	408	67891	600	24/05/2024	27/05/2024	8
9	409	78912	650	25/05/2024	28/05/2024	9
10	410	89012	700	26/05/2024	29/05/2024	10

	DEPOSITID	CUSTOMER_ID	DEPOSIT_AMOUNT	DEPOSIT_DATE	MATURITY_DATE	INTEREST_RATE	ACCOUNT_ID
1	501	10	3000	15/05/2024	20/05/2025	3	1
2	504	40	4000	17/05/2024	22/05/2025	4	4
3	505	50	4500	18/05/2024	23/05/2025	4	5
4	506	60	5000	19/05/2024	24/05/2025	4	6
5	507	70	5500	20/05/2024	25/05/2025	4	7
6	508	80	6000	21/05/2024	26/05/2025	4	8
7	509	90	6500	22/05/2024	27/05/2025	5	9
8	510	100	7000	23/05/2024	28/05/2025	5	10
9	502	20	2000	10/05/2024	10/05/2025	3	2
10	503	30	3500	16/05/2024	21/05/2025	4	3

	CARD_ID	CUSTOMER_ID	CARD_NUMBER	CARD_TYPE	EXPIRATION_DATE	CREDIT_LIMIT	ACCOUNT_ID
1	301	10	123456789012	Visa	20/05/2027	15000	1
2	302	20	987654321012	Mastercard	20/05/2026	10000	2
3	303	30	111122223333	Visa	30/06/2028	12000	3
4	304	40	444455556666	Mastercard	15/04/2025	8000	4
5	305	50	777788889999	Visa	22/08/2027	9000	5
6	306	60	123443215678	Mastercard	31/12/2026	11000	6
7	307	70	876543219876	Visa	01/01/2029	14000	7
8	308	80	112233445566	Mastercard	20/07/2026	13000	8
9	309	90	998877665544	Visa	25/11/2027	16000	9
10	310	100	334455667788	Mastercard	30/05/2025	7000	10

	LOAN_ID	CUSTOMER_ID	LOAN_AMOUNT	INTEREST_RATE	START_DATE	END_DATE	ACCOUNT_ID
1	201	10	10000	7	20/05/2023	20/05/2024	1
2	202	20	5000	8	20/05/2022	20/05/2023	2
3	203	30	10000	7	20/05/2023	20/05/2024	3
4	204	40	5000	8	20/05/2022	20/05/2023	4
5	205	50	10000	7	20/05/2023	20/05/2024	5
6	206	60	5000	8	20/05/2022	20/05/2023	6
7	207	70	10000	7	20/05/2023	20/05/2024	7
8	208	80	5000	8	20/05/2022	20/05/2023	8
9	209	90	10000	7	20/05/2023	20/05/2024	9
10	210	100	5000	8	20/05/2022	20/05/2023	10

	TRANSACTION_ID	AMOUNT	ACCOUNT_ID	TRANSACTION_DATE	
▶ 1	101	500	1	18/05/2024	...
2	102	410	2	18/05/2024	...
3	103	487	3	18/05/2024	...
4	104	250	4	18/05/2024	...
5	105	452	5	18/05/2024	...
6	106	250	6	18/05/2024	...
7	107	553	7	18/05/2024	...
8	108	255	8	18/05/2024	...
9	109	550	9	18/05/2024	...
10	110	111	10	18/05/2024	...

נתונים אלו הוכנסו ידנית על ידי הקובץ InsertTable.sql

```

createTables.sql dropTables.sql selectAll.sql main insertTables.sql X
SQL Output Statistics
INSERT INTO Accounts (Account_ID, Customer_ID, Account_Type, Balance, Account_Opening_Date)
VALUES (1, 10, 'Savings', 1000, DATE '2024-05-20');

INSERT INTO Accounts (Account_ID, Customer_ID, Account_Type, Balance, Account_Opening_Date)
VALUES (2, 20, 'Checking', 2000, DATE '2024-05-15');

INSERT INTO Accounts (Account_ID, Customer_ID, Account_Type, Balance, Account_Opening_Date)
VALUES (3, 30, 'Checking', 2555, DATE '2024-05-16');

INSERT INTO Accounts (Account_ID, Customer_ID, Account_Type, Balance, Account_Opening_Date)
VALUES (4, 40, 'Checking', 2040, DATE '2024-05-19');
INSERT INTO Accounts (Account_ID, Customer_ID, Account_Type, Balance, Account_Opening_Date)
VALUES (5, 50, 'Savings', 2900, DATE '2024-05-18');
INSERT INTO Accounts (Account_ID, Customer_ID, Account_Type, Balance, Account_Opening_Date)
VALUES (6, 60, 'Checking', 9999, DATE '2024-05-12');
INSERT INTO Accounts (Account_ID, Customer_ID, Account_Type, Balance, Account_Opening_Date)
VALUES (7, 70, 'Savings', 1542, DATE '2024-05-07');
INSERT INTO Accounts (Account_ID, Customer_ID, Account_Type, Balance, Account_Opening_Date)
VALUES (8, 80, 'Checking', 8521, DATE '2024-05-06');
INSERT INTO Accounts (Account_ID, Customer_ID, Account_Type, Balance, Account_Opening_Date)
VALUES (9, 90, 'Savings', 4523, DATE '2024-05-05');
INSERT INTO Accounts (Account_ID, Customer_ID, Account_Type, Balance, Account_Opening_Date)
VALUES (10, 100, 'Checking', 4658, DATE '2024-05-01');

INSERT INTO Transactions (Transaction_ID, Amount, Account_ID, Transaction_Date)
VALUES (101, 500, 1, DATE '2024-05-18');

INSERT INTO Transactions (Transaction_ID, Amount, Account_ID, Transaction_Date)
VALUES (102, 410, 2, DATE '2024-05-18');

```

נראה שימוש ב-Data Generator להכנסת מידע באופן אוטומטי:

createTables.sql					
dropTables.sql					
selectAll.sql					
main					
insertTables.sql					
Data Generator					
ACCOUNTS					
< Owner					
> Table					
Number of records					
SYS					
ACCOUNTS					
400					
...					
Name	Type	Size	Data	Master	
ACCOUNT_ID	NUMBER		Random(11, 100000)	...	
CUSTOMER_ID	NUMBER		Signal(110, 420, 10, 1)	...	
ACCOUNT_TYPE	VARCHAR2	10	List('Savings', 'Checking')	...	
BALANCE	NUMBER		Random(1000, 1000000)	...	
ACCOUNT_OPENING_DATE	DATE		List(select ACCOUNT_OPENING_DATE from Accounts)	...	
*				...	

createTables.sql					
dropTables.sql					
selectAll.sql					
main					
insertTables.sql					
Data Generator					
CHECKS					
< Owner					
> Table					
Number of records					
SYS					
CHECKS					
10..20					
...					
Name	Type	Size	Data	Master	
CHECK_ID	NUMBER		Random(1000, 2000)	...	
CHECK_NUMBER	NUMBER		Random(0, 400)	...	
AMOUNT	NUMBER		Random(1, 1000)	...	
ISSUE_DATE	DATE		List(select Account_Opening_Date from accounts)	...	
CLEARING_DATE	DATE		List(select Account_Opening_Date from accounts)	...	
ACCOUNT_ID	NUMBER		List(select account_id from account)	...	
*				...	

createTables.sql dropTables.sql selectAll.sql main insertTables.sql Data Generator X

LOANS

< Owner Table Number of records
> SYS LOANS 10..20
...

Name	Type	Size	Data	M
LOAN_ID	NUMBER		Random(1, 10000)	***
LOAN_AMOUNT	NUMBER		Random(1, 10000)	***
INTEREST_RATE	NUMBER		Random(1, 7)	***
START_DATE	DATE		List(select Account_Opening_Date from accounts)	***
END_DATE	DATE		List(select Account_Opening_Date from accounts)	***
✓ ACCOUNT_ID	NUMBER		List(select Account_id from accounts)	***
*				***

create tables.sql drop tables.sql selectAll.sql main insert tables.sql loansgd.gd X

TRANSACTIONS

< Owner Table Number of records
> SYS TRANSACTIONS 400
...

Name	Type	Size	Data	Master
TRANSACTION_ID	NUMBER		Random(10, 1000)	***
AMOUNT	NUMBER		Random(50, 5000)	***
ACCOUNT_ID	NUMBER		List(select account_id from accounts)	***
✓ TRANSACTION_DATE	DATE		List(select Account_Opening_Date from accounts)	***
*				***

createTables.sql dropTables.sql selectAll.sql main insertTables.sql trangd.gd X

CREDIT_CARDS

< Owner Table Number of records
> SYS CREDIT_CARDS 400
...

Name	Type	Size	Data	Master
CARD_ID	NUMBER		Random(100, 1000)	***
CARD_NUMBER	NUMBER		Random(1000000, 9999999)	***
CARD_TYPE	VARCHAR2	10	List('viaz', 'mastercad', 'debit')	***
EXPIRATION_DATE	DATE		List(select Account_Opening_Date from accounts)	***
CREDIT_LIMIT	NUMBER		Signal(2000, 10000, 500, 1)	***
✓ ACCOUNT_ID	NUMBER		List(select account_id from accounts)	***
*				***

createTables.sql					
dropTables.sql					
selectAll.sql					
main					
insertTables.sql					
crditdg.gd					
DEPOSITS					
< Owner Table Number of records					
> SYS DEPOSITS 400					
...					
Name	Type	Size	Data	Master	
DEPOSITID	NUMBER		Random(20, 1000)	...	
DEPOSIT_AMOUNT	NUMBER		Random(20, 500)	...	
DEPOSIT_DATE	DATE			...	
MATURITY_DATE	DATE		List(select Account_Opening_Date from accounts)	...	
INTEREST_RATE	NUMBER		Random(1, 7)	...	
✓ ACCOUNT_ID	NUMBER		List(select account_id from accounts)	...	
*				...	

בונה סקריפט בפייטון שיכניס נתונים לטבלאות:

```

C:\Users> ohevd > Downloads > DataImporterFiles > pythonSql.py
1 import random
2 from faker import Faker
3
4 fake = Faker()
5
6
7 # Function to generate SQL insert statements
8 def generate_sql():
9     with open('insert_records.sql', 'w') as f:
10         # Generate 400 records for Accounts table
11         for account_id in range(1000, 1401):
12             customer_id = account_id
13             balance = random.randint(1000, 50000)
14             account_opening_date = fake.date_this_decade()
15             f.write(
16                 f"INSERT INTO Accounts (Account_ID, Customer_ID, Balance, Account_Opening_Date) VALUES ({account_id}, {customer_id}, {balance}, TO_DATE('{account_opening_
17
18         # Generate 400 records for Transactions table
19         for transaction_id in range(1000, 1401):
20             amount = random.randint(100, 10000)
21             account_id = random.randint(1, 400)
22             transaction_date = fake.date_this_decade()
23             f.write(
24                 f"INSERT INTO Transactions (Transaction_ID, Amount, Account_ID, Transaction_Date) VALUES ({transaction_id}, {amount}, {account_id}, TO_DATE('{transaction_
25
26         # Generate 400 records for Loans table
27         for loan_id in range(1000, 1401):
28             loan_amount = random.randint(5000, 100000)
29             interest_rate = random.randint(1, 10)
30             start_date = fake.date_this_decade()
31             end_date = fake.date_between(start_date=start_date, end_date='+5y')
32             account_id = random.randint(1, 400)
33             f.write(
34                 f"INSERT INTO Loans (Loan_ID, Loan_Amount, Interest_Rate, Start_Date, End_Date, Account_ID) VALUES ({loan_id}, {customer_id}, {loan_amount}, {interest_rat
35

```

```

> Users> ohevd > Downloads > DataImporterFiles > pythonSql.py
8 def generate_sql():
9
10
11
12
13
14
15
16
17 for card_id in range(1000, 1401):
18     card_number = fake.credit_card_number(card_type=None)
19     card_type = fake.credit_card_provider()
20     expiration_date = fake.date_this_decade()
21     credit_limit = random.randint(5000, 20000)
22     account_id = random.randint(1, 400)
23     f.write(
24         f"INSERT INTO Credit_Cards (Card_ID, Card_Number, Card_Type, Expiration_Date, Credit_Limit, Account_ID) VALUES ({card_id}, {customer_id}, {card_number}, '
25
26
27 # Generate 400 records for Checks table
28 for check_id in range(1000, 1401):
29     check_number = random.randint(100000, 999999)
30     amount = random.randint(100, 10000)
31     issue_date = fake.date_this_decade()
32     clearing_date = fake.date_between(start_date=issue_date, end_date='+1y')
33     account_id = random.randint(1, 400)
34     f.write(
35         f"INSERT INTO Checks (Check_ID, Check_Number, Amount, Issue_Date, Clearing_Date, Account_ID) VALUES ({check_id}, {check_number}, {amount}, TO_DATE('{issue
36
37
38 # Generate 400 records for Deposits table
39 for deposit_id in range(1000, 1401):
40     deposit_amount = random.randint(1000, 50000)
41     deposit_date = fake.date_this_decade()
42     maturity_date = fake.date_between(start_date=deposit_date, end_date='+5y')
43     interest_rate = random.randint(1, 10)
44     account_id = random.randint(1, 400)
45     f.write(
46         f"INSERT INTO Deposits (DepositID, Deposit_Amount, Deposit_Date, Maturity_Date, Interest_Rate, Account_ID) VALUES ({deposit_id}, {customer_id}, {deposit_a
47
48
49 if __name__ == "__main__":
50     generate_sql()
51

```

נראה הכנסה של מידע מקובץ טקסט לטבלה accounts:

```
*C:\Users\ohevd\Downloads\מסמך טקסט חדש.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
createTables.sql | dropTables.sql | insertTables.sql | selectAll.sql | מסמך טקסט חדש.txt
1 500,5000, Savings, 1000, 20/05/2024
2 600,6000, Savings, 2000, 21/05/2024
3 700,7000, Savings, 3000, 22/05/2024
4 800,8000, Savings, 4000, 23/05/2024
```

createTables.sql | dropTables.sql | selectAll.sql | main | insertTables.sql | text1.txt

Data from Textfile | Data to Oracle

File Data	Encoding
500,5000, "Savings", 1000, 20/05/2024	
600,6000, "Savings", 2000, 21/05/2024	
700,7000, "Savings", 3000, 22/05/2024	
800,8000, "Savings", 4000, 23/05/2024	

Configuration

General

Fieldcount: 5

☒ End at line-end

☐ Name in header

☒ Skip empty lines

Quote character: "

Comment line: --

Import lines: 1 ..

Field1 (+0 .. ",")

Field2 (+0 .. ",")

Field3 (+0 .. ",")

Field4 (+0 .. ",")

Field5 (+0 .. ",")

Field Start:

☒ Relative position: 0

☐ Absolute position

☐ Character

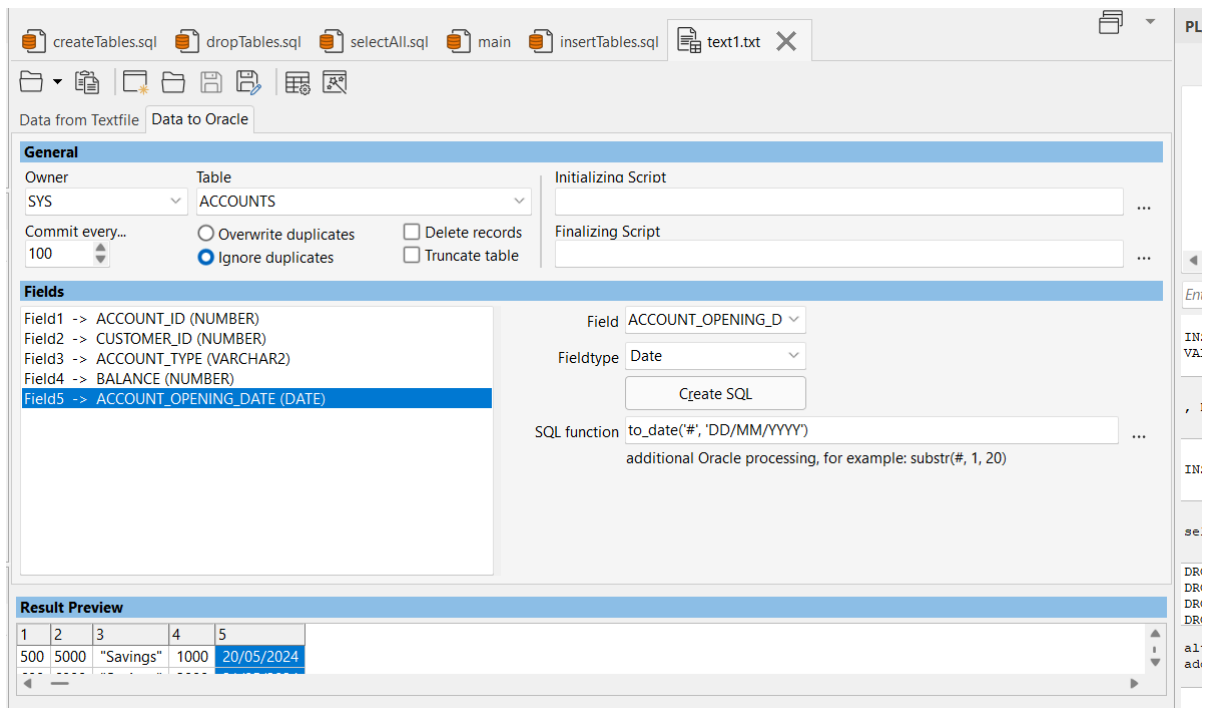
Field End:

☐ Length

☒ Character: ,

Filter:

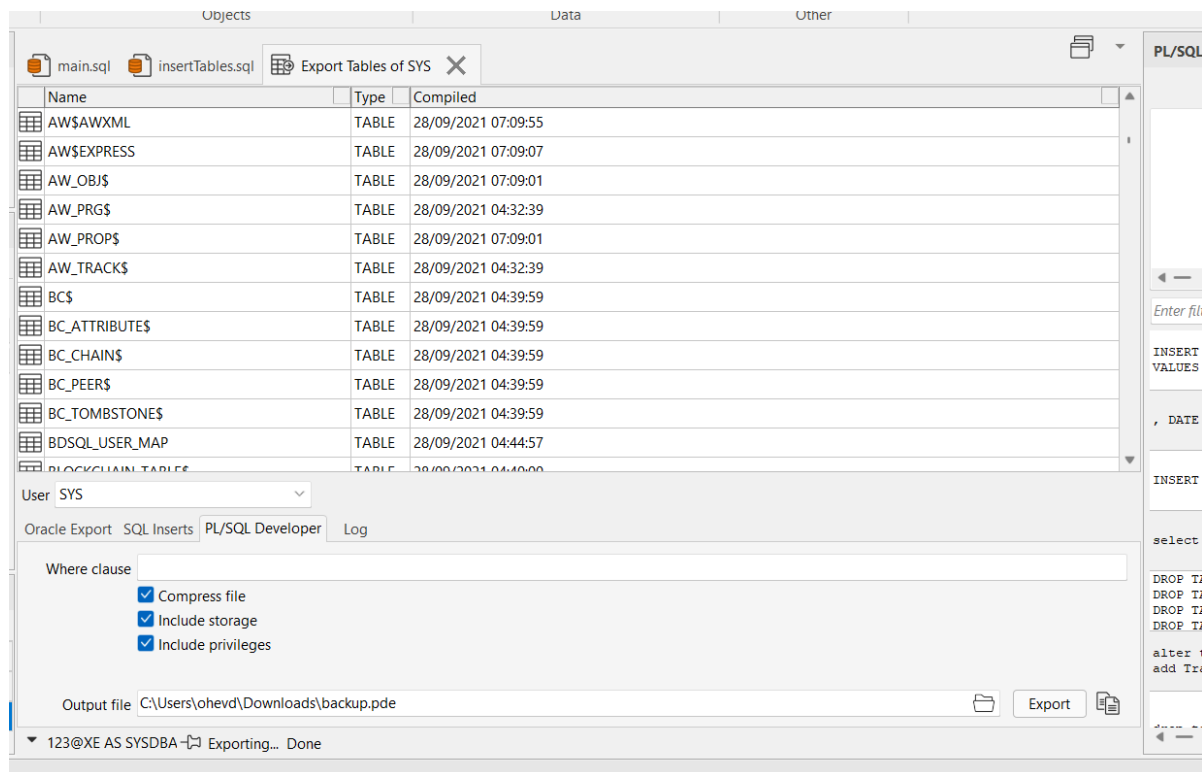
Apply



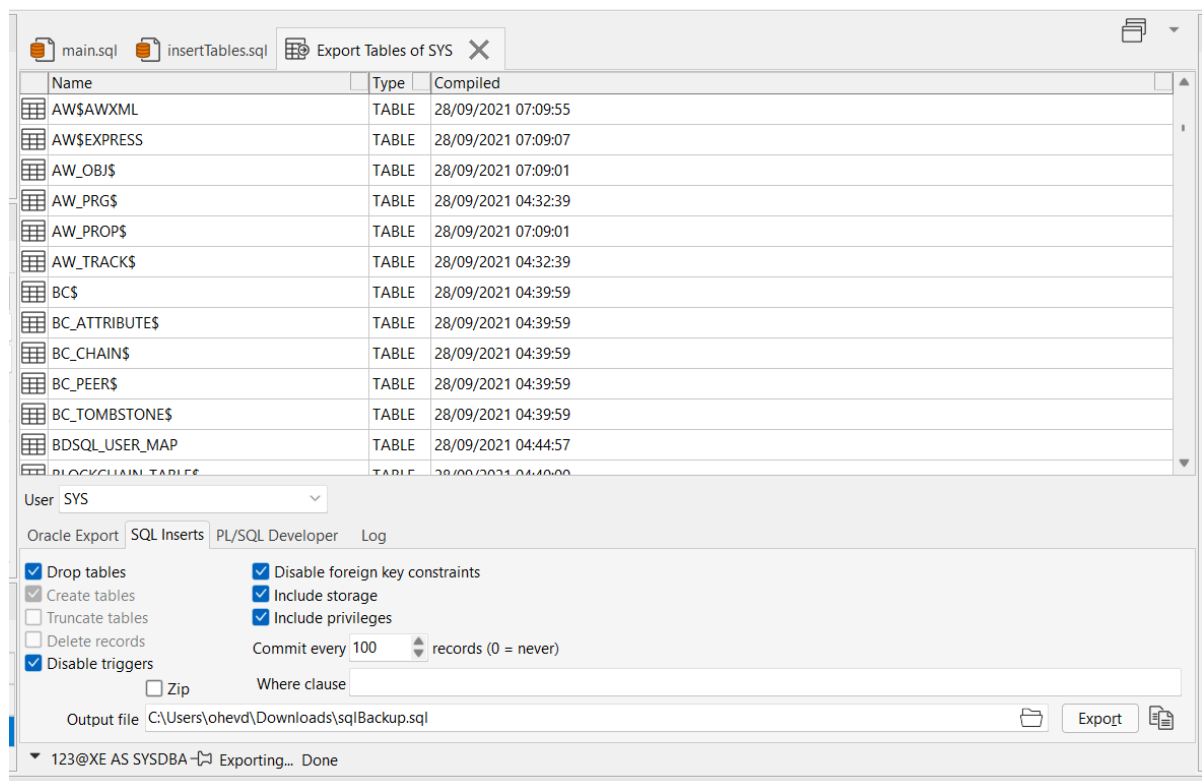
מדהים, ניתן גם בקלות לעשות סקריפט בפייתון שימלא לנו את הקובץ, ואז נמשוך את המידע מהקובץ בצורה זו.

בעצם הראנו ארבע אופציות איך להכניס מידע לטבלה: ידני, סקריפט, קובץ טקסט, Data Generator.

נבצע גיבוי למסד הנתונים בכדי שנוכל לפתוח אותו מכל מחשב.
נבצע export :



כמו כן ניתן לייצא את הגיבוי לקובץ SQL



קובץ הSQL נראה ככה:

```
sqlBackup21.5.24.sql X
C: > Users > ohevd > Downloads > sqlBackup21.5.24.sql
1 prompt PL/SQL Developer Export Tables for user SYS@XE
2 prompt Created by ohevd on י"ד י"ג שבט 21 י"א 2024
3 set feedback off
4 set define off
5
6 prompt Dropping ACCOUNTS...
7 drop table ACCOUNTS cascade constraints;
8 prompt Dropping CHECKS...
9 drop table CHECKS cascade constraints;
10 prompt Dropping CREDIT_CARDS...
11 drop table CREDIT_CARDS cascade constraints;
12 prompt Dropping DEPOSITS...
13 drop table DEPOSITS cascade constraints;
14 prompt Dropping LOANS...
15 drop table LOANS cascade constraints;
16 prompt Dropping TRANSACTIONS...
17 drop table TRANSACTIONS cascade constraints;
18 prompt Creating ACCOUNTS...
19 create table ACCOUNTS
20 (
21     account_id          INTEGER not null,
22     customer_id         INTEGER not null,
23     account_type        VARCHAR2(10),
24     balance              INTEGER not null,
25     account_opening_date DATE not null
26 )
27 tablespace SYSTEM
28     pctfree 10
29     pctused 40
30     initrans 1
31     maxtrans 255
32     storage
33     (
34         initial 64K
35         next 1M
36         minextents 1
```

נדגים מחיקה של הטבלאות והחזרתן:
נריץ את הקובץ דרופ אול:

main.sql

insertTables.sql

Export Tables of SYS

dropTables.sql

SQL

Output

Statistics

DROP TABLE Transactions;

DROP TABLE Loans;

DROP TABLE Credit_Cards;

DROP TABLE Deposits;

DROP TABLE Checks;

DROP TABLE Accounts;

Drop transactions

Drop loans

Drop credit_cards

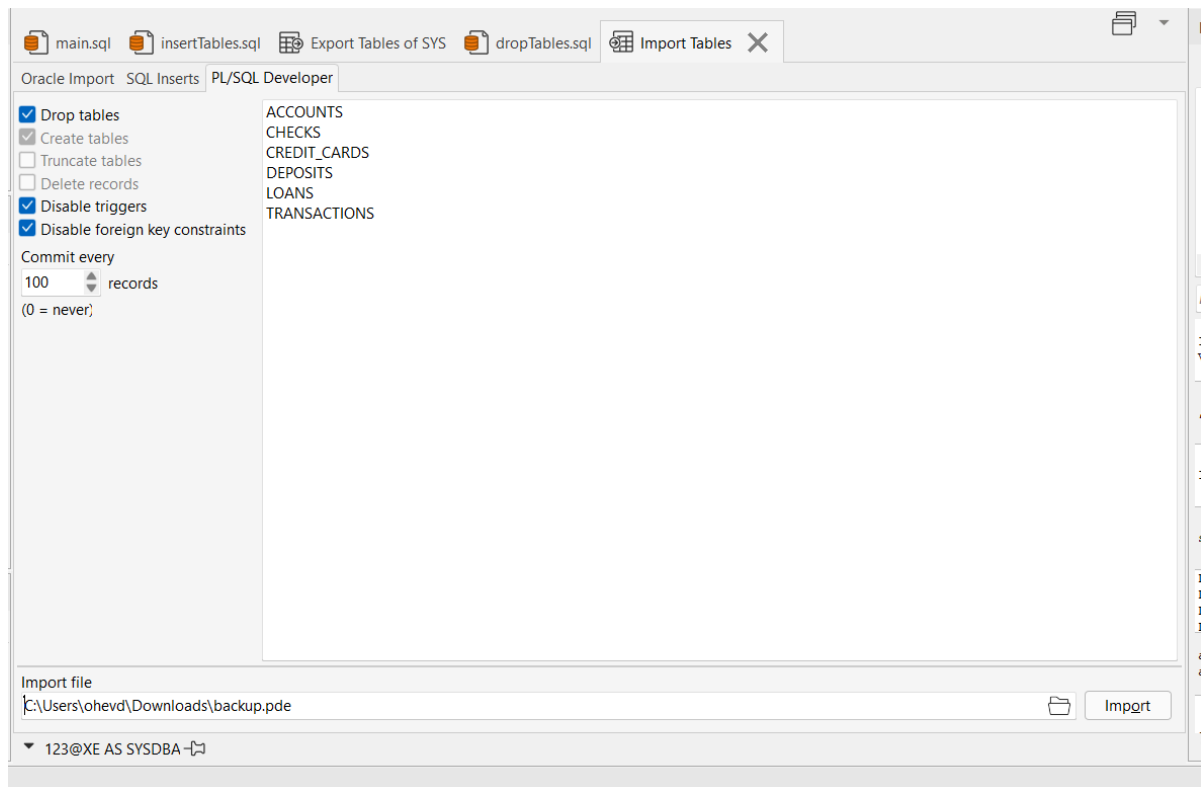
Drop deposits

Drop checks

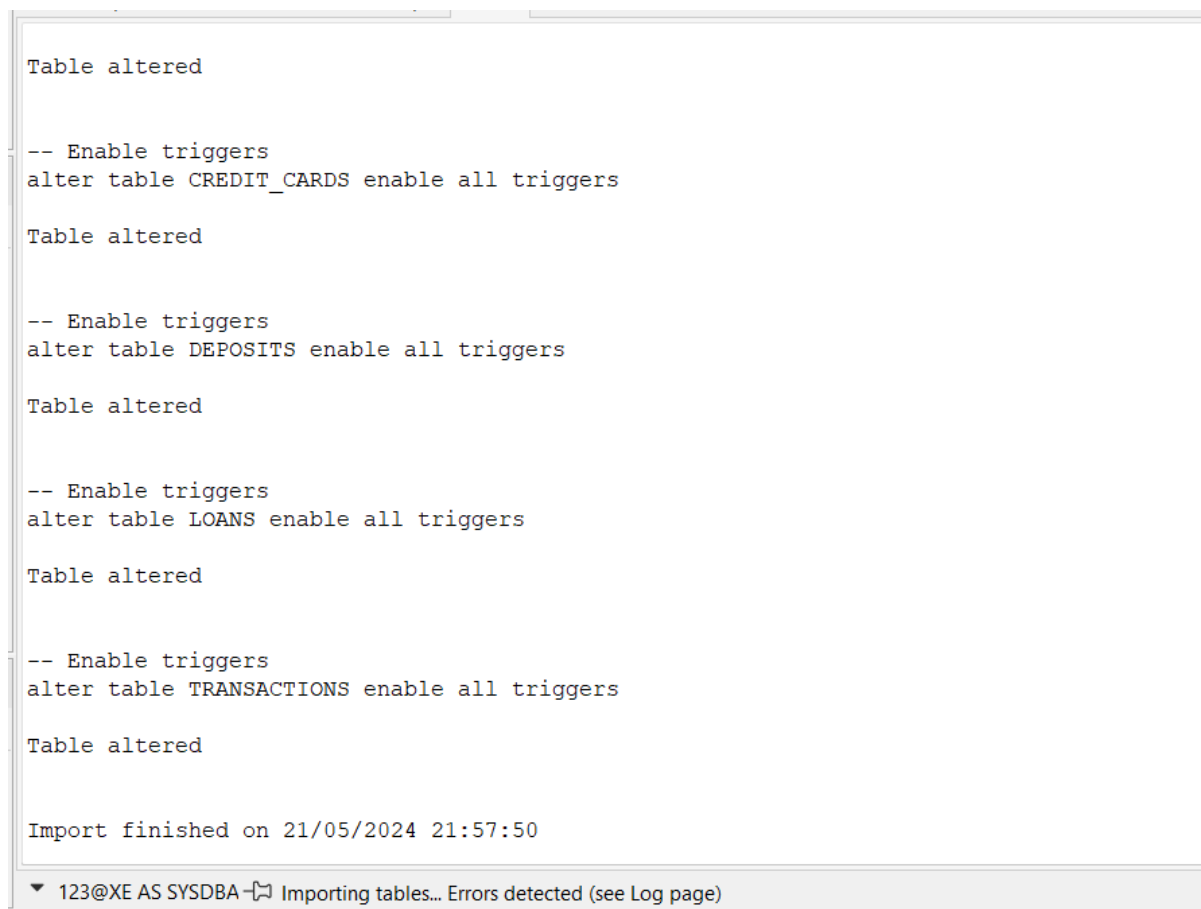
Drop accounts

(no result set)

נבצע שיחזור:



השיחזור הושלם:



נריץ את הקובץ selectAll בכדי לראות שהנתונים חזרו:

The screenshot shows the SQL Developer interface. The top toolbar includes buttons for main.sql, insertTables.sql, Export Tables of SYS, dropTables.sql, Import Tables, and selectAll.sql. The SQL window contains the following query:

```
select * from Transactions;  
select * from Loans;  
select * from Credit_Cards;  
select * from Deposits;  
select * from Checks;  
select * from Accounts;
```

Below the query window, there are tabs for 'Select transactions', 'Select loans', 'Select credit_cards', 'Select deposits', 'Select checks', and 'Select accounts'. The 'Select transactions' tab is active, displaying a results grid with the following data:

	TRANSACTION_ID	AMOUNT	ACCOUNT_ID	TRANSACTION_DATE	
1	101	500	1	18/05/2024	...
2	102	410	2	18/05/2024	...
3	103	487	3	18/05/2024	...
4	104	250	4	18/05/2024	...
5	105	452	5	18/05/2024	...
6	106	250	6	18/05/2024	...
7	107	553	7	18/05/2024	...
8	108	255	8	18/05/2024	...
9	109	550	9	18/05/2024	...
10	110	111	10	18/05/2024	...
11	7955	1923	85	15/05/2024	...
12	3562	1061	333	05/05/2024	...
13	6448	1421	697	06/05/2024	...

The status bar at the bottom indicates '1 of 13' rows and '123@XE AS SYSDBA [21:59:12] 13 rows selected in 0.088 seconds (more...)'. The status bar also shows '000' and a magnifying glass icon.

טא דא! הכל שב אלינו.

ע