

שלב 3

דוד אוהב ציון ומרקוס צ'אמה.

בעקבות האינפלציה במשק, הוחלט בבנק על מתן הקלות ללקוחות פעילים בבנק. עבור לקוחות פעילים אלו הבנק מוכן לקצץ עד 5% מהריבית על ההלוואות. ובכל מקרה ריבית על הלוואה לא תפחת מ-2%.

קרטונים ללקוח פעיל:

- ביצע לפחות 5 העברות בשנתיים האחרונות.
- פדה לפחות 5 שיקים בשנה האחרונה.
- ביצע לפחות 5 הפקדות מאז פתיחת החשבון.

בהתאם לזה, הבנק מוכן לקצץ את אחוזי הריבית ע"פ הנוסחה הנ"ל:
(סכום השקים שנפדו בשנה האחרונה + סכום ההפקדות בשנה האחרונה) \ מספר החודשים שהחשבון קיים) * 0.01

ובכל מקרה סכום ההנחה לא יעלה על 5% - כלומר לקוח עם הלוואה בגובה ריבית 7% וזכאות להנחה בגובה 6% יזכה להנחה רק בגובה 5% ובהתאם לזה גובה הריבית על הלוואה שלו יעודכן ל-2%.

נבנה פונקציה שבודקת לאיזה לקוח מגיע הנחה ועוד פונקציה שמחזירה את גובה ההנחה. כמו כן נבנה פרוצדורה שמעדכנת את בסיס הנתונים. נבנה פונקציה Main שתבצע את כל התהליך.

פונקציה שבודקת האם ללקוח מגיע הנחה:

```

create or replace noneditionable function is_active5(p_account_id IN NUMBER) return boolean is
FunctionResult boolean;
v_transfers_count integer;
v_checks_count integer;
v_deposits_count integer;
begin
SELECT COUNT(*) INTO v_transfers_count
FROM Transactions t
WHERE account_id = p_account_id
AND t.transaction_date >= ADD_MONTHS(SYSDATE, -24);

SELECT COUNT(*) INTO v_checks_count
FROM Checks ch
WHERE ch.account_id = p_account_id
AND ch.clearing_date >= ADD_MONTHS(SYSDATE, -12);

SELECT COUNT(*) INTO v_deposits_count
FROM Deposits dp
WHERE dp.account_id = p_account_id;

IF v_transfers_count >= 5 AND v_checks_count >= 5 AND v_deposits_count >= 5 THEN
FunctionResult := TRUE;
ELSE
FunctionResult := FALSE;
END IF;

begin
SELECT COUNT(*) INTO v_transfers_count
FROM Transactions t
WHERE account_id = p_account_id
AND t.transaction_date >= ADD_MONTHS(SYSDATE, -24);

SELECT COUNT(*) INTO v_checks_count
FROM Checks ch
WHERE ch.account_id = p_account_id
AND ch.clearing_date >= ADD_MONTHS(SYSDATE, -12);

SELECT COUNT(*) INTO v_deposits_count
FROM Deposits dp
WHERE dp.account_id = p_account_id;

IF v_transfers_count >= 5 AND v_checks_count >= 5 AND v_deposits_count >= 5 THEN
FunctionResult := TRUE;
ELSE
FunctionResult := FALSE;
END IF;
return(FunctionResult);
end is_active5;

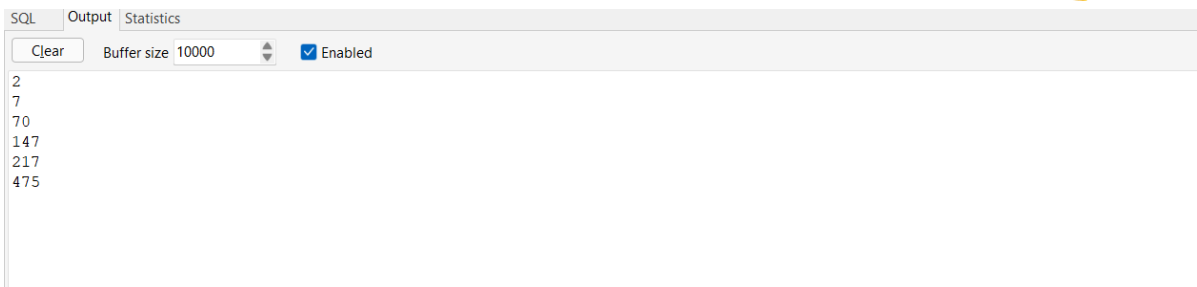
```

נראה את מספרי הלקוחות שעבורם הפונקציה תחזיר אמת:

נריץ פונקציה שתבדוק :

```
---main
DECLARE
    CURSOR customer_cursor IS
        SELECT account_ID FROM Accounts;
BEGIN
    FOR customer_record IN customer_cursor LOOP
        if is_active5(customer_record.account_id) then
            dbms_output.put_line(customer_record.account_id);
        end if;
    END LOOP;
END;
```

והתוצאה 👍



SQL Output Statistics

Clear Buffer size 10000 Enabled

2
7
70
147
217
475

כלומר מצאנו 6 לקוחות שמגיע להם הנחה בגובה הריבית בהלוואות.
פונקציה שמחזירה את גובה ההנחה עבור כל לקוח שנמצא זכאי:

```

1 CREATE OR REPLACE FUNCTION get_discount(p_account_id IN NUMBER)
2 RETURN NUMBER IS
3     v_checks_sum NUMBER;
4     v_deposits_sum NUMBER;
5     v_account_open_date DATE;
6     v_months_open NUMBER;
7     func_result NUMBER;
8 BEGIN
9
10    SELECT NVL(SUM(ch.amount),0) INTO v_checks_sum
11    FROM Checks ch
12    WHERE ch.account_id = p_account_id
13          AND ch.clearing_date >= ADD_MONTHS(SYSDATE, -12);
14
15
16    SELECT NVL(SUM(dep.Deposit_Amount),0) INTO v_deposits_sum
17    FROM Deposits dep
18    WHERE dep.account_id = p_account_id
19          AND dep.deposit_date >= ADD_MONTHS(SYSDATE, -12);
20
21
22    select a.account_opening_date
23    INTO v_account_open_date
24    from accounts a
25
26
27    FROM Deposits dep
28    WHERE dep.account_id = p_account_id
29          AND dep.deposit_date >= ADD_MONTHS(SYSDATE, -12);
30
31
32    select a.account_opening_date
33    INTO v_account_open_date
34    from accounts a
35    where a.account_id = p_account_id;
36
37
38    v_months_open := MONTHS_BETWEEN(SYSDATE, v_account_open_date);
39
40
41    func_result := ((v_deposits_sum + v_checks_sum) / v_months_open) *0.01;
42
43
44    IF func_result > 5 THEN
45        func_result := 5;
46    END IF;
47
48    RETURN ROUND (NVL(func_result,0),2);
49
50 END get_discount;

```

נבצע הריצת בדיקה: הדפס עבור כל מי שזכאי את גובה ההנחה לה הוא זכאי:

```

DECLARE
    CURSOR customer_cursor IS
        SELECT account_ID FROM Accounts;
BEGIN
    FOR customer_record IN customer_cursor LOOP
        if is_active5(customer_record.account_id) then
            dbms_output.put_line(get_discount(customer_record.account_id));
        end if;
    END LOOP;
END;

```

```
5
5
3.37
1.76
2.07
2.45
```

כלומר מצאנו שכל הלקוחות הזכאים להנחה מסויימת בריבית על ההלוואה, ושני לקוחות זכאים לגובהה המקסימלי של ההנחה.
עכשיו רק נותר ליצור פרוצדורה שבודקת האם יש ללקוח הלוואות, מכל אותן הלאות היא תבחר את ההלוואה עם הריבית הגבוהה ביותר, ותתן ללקוח את ההנחה (תשנה את הנתונים)

```
CREATE OR REPLACE PROCEDURE apply_discount(p_account_id IN NUMBER,p_discount IN NUMBER
IS
    v_loan_id NUMBER;
    v_max_interest NUMBER;
    v_new_interest NUMBER;
BEGIN
    BEGIN
        SELECT loan_id, interest_rate
        INTO v_loan_id, v_max_interest
        FROM (
            SELECT loan_id, interest_rate
            FROM Loans
            WHERE account_id = p_account_id
            ORDER BY interest_rate DESC
        )
        WHERE ROWNUM = 1;

        v_new_interest := v_max_interest - p_discount;

        IF v_new_interest < 2 THEN
            v_new_interest := 2;

            v_new_interest := v_max_interest - p_discount;

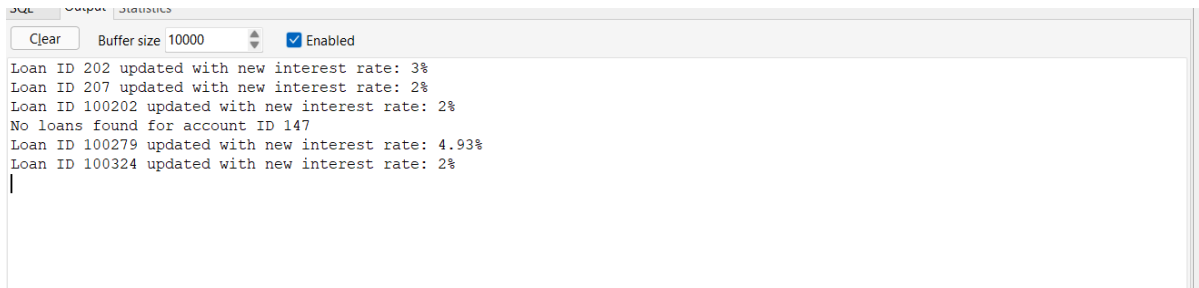
            IF v_new_interest < 2 THEN
                v_new_interest := 2;
            END IF;

            UPDATE Loans
            SET interest_rate = v_new_interest
            WHERE loan_id = v_loan_id;

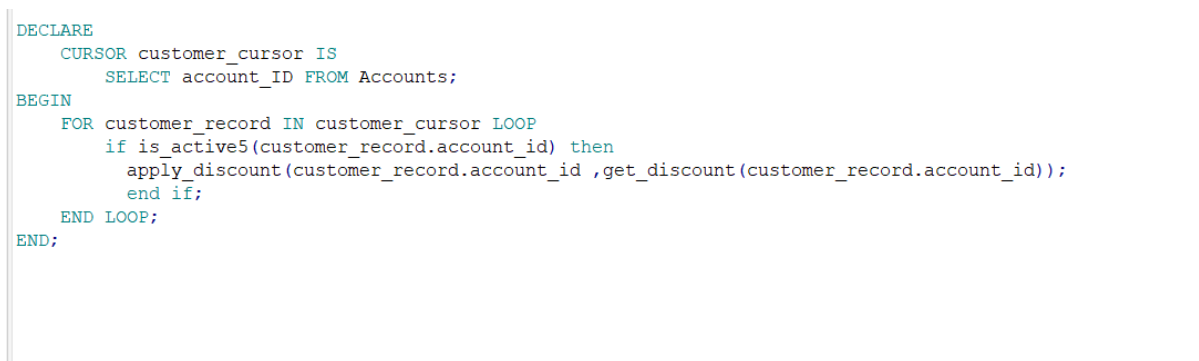
            DBMS_OUTPUT.PUT_LINE('Loan ID ' || v_loan_id || ' updated with new interest rate: ' || v_new_

        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                DBMS_OUTPUT.PUT_LINE('No loans found for account ID ' || p_account_id);
        END;
    END;
END;
```

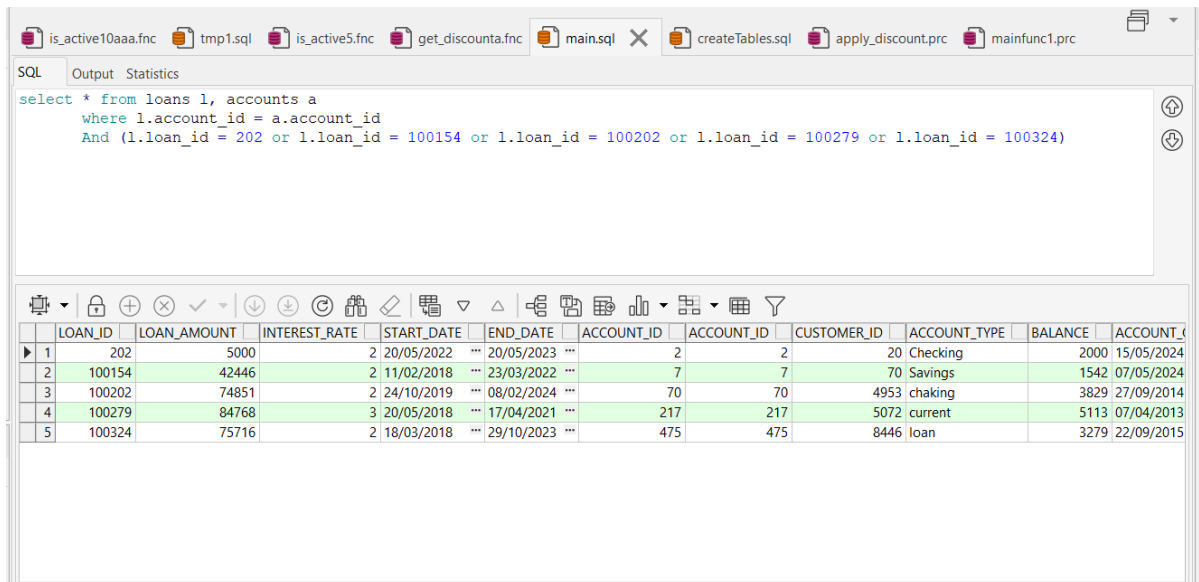
נריץ בדיקה:



נגדיר פונקציות מיין שתבדוק עבור כל לקוח האם מגיעה לו הנחה, אם כן, מה גובה ההנחה ועידכון DB!



נראה שאכן הטבלאות מתעדכנות:



וואוואלה:
תוכנית 2:

בעקבות המלחמה הבנק נכנס לתקופה קשה ועומד לפני פשיטת רגל, בשביל להציל את המצב, ולמנוע מכל לקוחות הבנק לאבד את כספם במקרה של פשיטת רגל, הוחלט בבנק על צעדים משמעותיים שיפגעו במספר לקוחות אך יצילו את הבנק מקריסה! תוכנית הפעולה של הבנק היא לגבות את כל ההלוואות שתאריך הפידיון שלהם עוד לא הגיע, בתנאי שללקוח יש מספיק כסף נזיל בחשבון. במקרה שבו ללקוח יש מספר הלוואות, הבנק יפדה את ההלוואה הכי קטנה (סכום ההלוואה הנמוך ביותר). לדוגמא מנחם הנחומי לקח הלוואה של 10 אש"ח וצריך להחזיר אותה רק בעוד שנה, אם למנחם יש יותר מ10 אש"ח בחשבון הבנק באופן אוטומטי יקח את הכסף, ויידע על כך את מנחם. יש לדבר היתכנות משפטית, וזכויות הפרט אינם נכללים בשיקול זה. אלה שיקול הכללי. ניצור פונקציה שבודקת האם ללקוח יש הלוואה וקיים מספיק כסף בחשבון בשביל לכסות אותה, ופרוצדורה שתעדכן את הבסיס נתונים בכסף שנלקח. בנוסף נכתוב פונקציה מיין שתתפעל את הפונקציות שיצרנו.

פונקציה בוליאנית שמחזירה אמת אם הלקוח יכול לכסות הלוואה באופן מיידי:

```
CREATE OR REPLACE FUNCTION can_cover(p_account_id IN NUMBER)
RETURN BOOLEAN IS
    v_loan_amount NUMBER;
    v_account_balance NUMBER;
BEGIN
    SELECT MIN(Loan_Amount)
    INTO v_loan_amount
    FROM Loans
    WHERE account_id = p_account_id
    AND End_Date > SYSDATE;

    SELECT Balance
    INTO v_account_balance
    FROM Accounts
    WHERE account_id = p_account_id;

    IF v_account_balance >= v_loan_amount THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;

    SELECT Balance
    INTO v_account_balance
    FROM Accounts
    WHERE account_id = p_account_id;

    IF v_account_balance >= v_loan_amount THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN FALSE;
END can_cover;
```

נריץ בדיקה:

```

DECLARE
    CURSOR customer_cursor IS
        SELECT account_ID FROM Accounts;
BEGIN
    FOR customer_record IN customer_cursor LOOP
        if can_cover(customer_record.account_id) then
            dbms_output.put_line(customer_record.account_id);
        end if;
    END LOOP;
END;

```

```

95
104
238
336

```

ב"ה ניתן לראות שרק ארבעה לקוחות נפלו ברשת של הבנק ויצטרכו לכסות את ההלוואה באופן מידוי.
ניצור פרוצדורה שמעדכנת את הנ"ל בבסיס נתונים.

```

1 create or replace procedure cover_loan_and_update(p_account_id in Number) is
2     v_loan_amount NUMBER;
3     v_loan_id NUMBER;
4
5 begin
6
7     update accounts a
8     set a.balance = a.balance - ( SELECT MIN(Loan_Amount)
9     FROM Loans
10    WHERE account_id = p_account_id
11        AND End_Date > SYSDATE)
12    where a.account_id = p_account_id;
13
14
15    --find loan_id of the minimal loan
16    SELECT loan_id, Loan_Amount
17    INTO v_loan_id, v_loan_amount
18    FROM Loans
19    WHERE account_id = p_account_id
20        AND End_Date > SYSDATE
21    ORDER BY Loan_Amount ASC
22    FETCH FIRST 1 ROWS ONLY;
23
24    update loans l
25    set loan_amount = 0

```

```

23
24    update loans l
25    set loan_amount = 0,
26    l.end_date = SYSDATE
27    where l.account_id = p_account_id
28    And l.loan_id = v_loan_id;
29
30    commit;
31    DBMS_OUTPUT.PUT_LINE('Loan covered and database updated successfully. ');
32    EXCEPTION
33    WHEN NO_DATA_FOUND THEN
34        DBMS_OUTPUT.PUT_LINE('No eligible loan found or insufficient balance. ');
35
36 end cover_loan_and_update;

```

בעצם אנו מעדכנים את היתרת חשבון ע"י הפחתה של סכום ההלוואה ע"י שאילתא מקוננת, ואח"כ מוצאים את מספר הלוואה בכדי לדעת איזה אחת לאפס ומאפסים את ההלוואה ומעדכנים את התאריך פדיון.


```
DECLARE
    CURSOR customer_cursor IS
        SELECT account_ID FROM Accounts;
BEGIN
    FOR customer_record IN customer_cursor LOOP
        if can_cover(customer_record.account_id) then
            cover_loan_and_update(customer_record.account_id);
        end if;
    END LOOP;
END;
```

```
Loan covered and database updated successfully.
Loan covered and database updated successfully.
Loan covered and database updated successfully.
Loan covered and database updated successfully.
```

```
SQL Output Statistics
select * from loans l
where (l.account id = 95 or l.account id = 104 or l.account id = 336 or l.account id = 238)
```


ניתן לראות שהתארכי סיום עודכנו להיום, הסכום עודכן ל0 חוץ ממספר חשבון 336 שהיו לו שתי הלוואות ורק ההלוואה הנמוכה כוסתה. ניתן לראות בוודאות שהנתונים שונים, כמו כן הנה לפני ואחרי של מצב החשבון, ניתן לראות שהסכום הנזיל השתנה בהתאם לסכום ההלוואה:
לפני:

SQL

Output

Statistics

```
select * from accounts a
  where (a.account_id = 95 or a.account_id = 104 or a.account_id = 336 or a.account_id = 238)
```




	ACCOUNT_ID	CUSTOMER_ID	ACCOUNT_TYPE	BALANCE	ACCOUNT_OPENING_DATE	
▶ 1	95	9721	loan	8222	12/04/2014	...
2	104	5693	saving	6802	15/07/2015	...
3	238	9686	saving	5479	17/03/2017	...
4	336	4644	current	4294967210	13/03/2014	...

אחרי:

SQL

Output Statistics

```
select * from accounts l
  where (l.account_id = 95 or l.account_id = 104 or l.account_id = 336 or l.account_id = 238)
```



	ACCOUNT_ID	CUSTOMER_ID	ACCOUNT_TYPE	BALANCE	ACCOUNT_OPENING_DATE	
▶	1	95	9721 loan	7642	12/04/2014	...
	2	104	5693 saving	2723	15/07/2015	...
	3	238	9686 saving	4680	17/03/2017	...
	4	336	4644 current	4294961219	13/03/2014	...

בשעה טובה, בזכות לקוחות פרטיים אלו הבנק ניצל מפשיטת רגל!

עד כאן שלב 3
דוד אוהב ציון ומרקוס צ'אמה.