

## מיניפ שלב ג

### מיכל שלומי

ביצעתי את העדכונים הבאים לטבלאות שלי:

1. הוספתי לטבלת ספר את העמודה DiscountPrice המציין את מחיר הספר לאחר ההנחה במידה וללקוח מופיע בעמודה DiscountPrice אחוז ההנחה שצריך לקבל (שונה מאפס)
2. הוספתי לטבלת הלקוח את העמודה DiscountPercentage המציין את אחוז ההנחה שיש לתת ללקוח בעת הרכישה. ברירת המחדל היא 0 ובמידה והלקוח זכאי להנחה כלשהי נעדכן את השדה הזה בהתאם.

```
ALTER TABLE Book  
ADD DiscountPrice NUMBER;
```

```
ALTER TABLE Customer  
ADD DiscountPercentage NUMBER DEFAULT 0;
```

3. החלפתי את העמודה 'status' להיות בשם 'condition' המתאר את מצב בספר: חדש/משומש/משומש במצב חדש

```
ALTER TABLE bookcopyy  
RENAME COLUMN Status TO Condition VARCHAR2(20);
```

4. הוספתי עמודה בשם 'status' המתארת אם הספר במלאי או לא:  
In stock/not in stock

```
ALTER TABLE bookcopyy  
ADD Status VARCHAR2(15) NOT NULL;
```

5. עבור התוכנית השנייה יצרתי טבלה חדשה בשם "book\_orders" עבור יצירת הזמנות חדשות לספרים.

```
CREATE TABLE book_orders  
(  
    orderID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    bookID NUMBER NOT NULL,  
    availableCopies NUMBER,  
    orderQuantity NUMBER ,  
    FOREIGN KEY (bookID) REFERENCES book(bookID)  
);
```

6. הוספתי עמודה המציינת את הבונוס של עובד בטבלה worker.

```
ALTER TABLE Worker  
ADD Bonus NUMBER DEFAULT 0;
```

## פרוצדורה 1:

הפרוצדורה מקבלת מספר לקוח ובודקת אם מגיע לו הנחה וכמה, ומעדכנת בהתאם את ההנחה,

הנחה ניתנת לפי מספר הספרים המקסימלי שרכש בקניה אחת.

עבור 100 פריטים ומעלה יקבל הנחה של 70 אחוז

עבור 20 פריטים ומעלה יקבל הנחה של 40 אחוז

עבור 10 פריטים ומעלה יקבל הנחה של 20 אחוז

פרוצדורה מעדכנת את השדה DiscountPercentage בהתאם.

## הקוד:

```
1 create or replace procedure update_customer_discount(customer_id IN NUMBER,
2                                     discount_info OUT SYS_REFCURSOR)
3 is
4     type purchase_record_type is record (
5         purchase_id number,
6         total_items number
7     );
8     purchase_record purchase_record_type;
9
10    cursor purchase_cursor is
11        select p.purchaseid, count(bs.bookid) as total_items
12        from purchase p
13        join booksales bs on p.purchaseid = bs.purchaseid
14        where p.customerid = customer_id
15        group by p.purchaseid;
16
17    max_items number := 0;
18    discount_percentage number := 0;
19
20 begin
21     -- open cursor
22     open purchase_cursor;
23
24     loop
25         fetch purchase_cursor into purchase_record;
26         exit when purchase_cursor%notfound;
27
28         -- check if the current purchase has more items than the previous maximum
```

```
28         -- check if the current purchase has more items than the previous maximum
29         if purchase_record.total_items > max_items then
30             max_items := purchase_record.total_items;
31         end if;
32     end loop;
33
34     close purchase_cursor;
35
36     -- determine discount percentage based on max items
37     case
38     when max_items >= 100 then
39         discount_percentage := 70;
40     when max_items >= 20 then
41         discount_percentage := 40;
42     when max_items >= 10 then
43         discount_percentage := 20;
44     else
45         discount_percentage := 0;
46     end case;
47
48     -- update customer's discount percentage
49     update customer
50     set discountpercentage = discount_percentage
51     where customerid = customer_id;
52
53     -- open ref cursor to return discount info
54     open discount_info for
55         select customerid, discountpercentage
```

```

53      -- open ref cursor to return discount info
54      open discount_info for
55          select customerid, discountpercentage
56          from customer
57          where customerid = customer_id;
58
59      -- print message based on discount applied
60      if discount_percentage > 0 then
61          dbms_output.put_line('customer ' || customer_id || ' is eligible for a ' || discount_percentage ||
62              '% discount. discount updated.' || ' he purchased ' || max_items || ' items');
63      else
64          dbms_output.put_line('customer ' || customer_id ||
65              ' is not eligible for any discount based on their purchase history. ');
66      end if;
67
68  exception
69      when no_data_found then
70          dbms_output.put_line('no purchase records found for customer ' || customer_id);
71      when others then
72          dbms_output.put_line('error updating discount for customer ' || customer_id);
73  end update_customer_discount;

```

**ההרצה:**

```

1  begin
2      -- Call the procedure
3      update_customer_discount(customer_id => :customer_id,
4                              discount_info => :discount_info);
5  end;

```

Variable	Type	Value
customer_id	Float	10
discount_info	Cursor	<Cursor>

6:1    michal@XE    [10:27:02] Executed in 0.002 seconds

Test script   DBMS Output   Statistics   Profiler   Trace

Clear   Buffer size 10000   ☒ Enabled

customer 10 is eligible for a 20% discount. discount updated. he purchased 10 items

**בדיקת פירטי הלקוח לאחר ההרצה:**

```
select * from customer where customer.customerid=10;
```

	CUSTOMERID	CUSTOMERFNAME	CUSTOMERLNAME	CUSTOMERPHONE	CUSTOMEREMAIL	DISCOUNTPERCENTAGE
1	10	Hannah	Rodriguez	1234567809	hannah.rodriguez@example.com	20

154:35    michal@XE    [10:49:54] 1 row selected in 0.030 seconds

## פונקציה 1 חדשה:

(לפי נתוני הטבלאות שלי כל אחד מהעובדים מכר בין 1 ל 5 פריטים סה"כ ולכן בפונקציה נתייחס לנתון הזה).

נכתוב פונקציה שרוצה לצ'פר עובדים שביצעו מספר מכירות גבוה ביחס לאחרים ולכן עובדים ש:

מכרו מעל 3 פריטים יזכו בבונוס של 50 ₪,

מכרו מעל 4 פריטים יזכו בבונוס של 100 ₪,

מכרו מעל 5 פריטים יזכו בבונוס של 200 ₪.

הקוד:

```
1 create or replace function update_bonus_for_worker(p_workerid in number)
2 return number
3 is
4     v_items_sold number := 0;
5     v_bonus_amount number := 0;
6 begin
7     -- Declare cursor for fetching worker details
8     for r_worker in (select workerid from worker where workerid = p_workerid) loop
9         -- Get the count of items sold by the worker
10        select count(*)
11        into v_items_sold
12        from purchase p
13        where p.workerid = r_worker.workerid;
14
15        -- Determine bonus amount based on items sold
16        case
17            when v_items_sold > 5 then
18                v_bonus_amount := 200;
19            when v_items_sold > 4 then
20                v_bonus_amount := 100;
21            when v_items_sold > 3 then
22                v_bonus_amount := 50;
23            else
24                v_bonus_amount := 0; -- No bonus if less than or equal to 3 items sold
25        end case;
26
27        -- Update the bonus for the worker if applicable
28        if v_bonus_amount > 0 then
29            update worker
30            set bonus = bonus + v_bonus_amount
31            where workerid = r_worker.workerid;
32        end if;
33
34        -- Return the bonus amount awarded
35        return v_bonus_amount;
36    end loop;
37
38    -- If no data found, return 0
39    return 0;
40
41 exception
42 when no_data_found then
43     return 0; -- Return 0 if no data found for worker ID
44 when others then
45     return -1; -- Return -1 for any other error
46 end update_bonus_for_worker;
```

לפני ההרצה עשיתי בדיקה של כמות הפריטים שמכרו העובדים:

ID: 123456789, Items Sold: 5  
ID: 123568974, Items Sold: 1  
ID: 323564871, Items Sold: 3  
ID: 323135648, Items Sold: 5  
ID: 315556965, Items Sold: 1  
ID: 226755554, Items Sold: 1  
ID: 743502274, Items Sold: 2  
ID: 718109763, Items Sold: 2  
ID: 335673276, Items Sold: 1  
ID: 901683660, Items Sold: 3  
ID: 137920011, Items Sold: 1  
ID: 397989285, Items Sold: 3  
ID: 294357585, Items Sold: 4  
ID: 822958626, Items Sold: 3

פירטי לקוח 123456789 לפני הרצה של הפונקציה:

```
select * from worker w where w.workerid=123456789;
```

	WORKERID	WORKERFNAME	WORKERLNAME	WORKERPHONE	POSITION	HIREDATE	BONUS
▶ 1	123456789	noam	levi	504865985	Manager	01/01/2020 ***	0

פירטי לקוח 123456789 לפני הרצה של הפונקציה:

```
select * from worker w where w.workerid=123456789;
```

	WORKERID	WORKERFNAME	WORKERLNAME	WORKERPHONE	POSITION	HIREDATE	BONUS
▶ 1	123456789	noam	levi	504865985	Manager	01/01/2020 ***	100

## הרצה של הפונקציה:

Test script
DBMS Output
Statistics
Profiler
Trace

```

1  begin
2      -- Call the function
3      :result := update_bonus_for_worker(p_workerid => :p_workerid);
4  end;

```

Variable	Type	Value
result	Float	100
p_workerid	Float	123456789

1:1

michal@XE

[12:16:33] Executed in 0.006 seconds

## תוכנית ראשית 1:

אנו מעוניינים לדאוג לשימור לקוחות רווחיים וכן לשימור עובדים שמכניסים יותר כסף לחברה ולכן ללקוחות הרווחיים אנו ניתן הנחות שאין לכולם, ולעובדים רווחיים ניתן מענקים לפי הרווח שהם מכניסים לחברה.

הפרוצדורה update customer discount יטפל בעדכוני ההנחות ללקוחות ואת העובדים נשלח לפונקציה update bonus for worker.

## הקוד:

SQL	Output	Statistics
-----	--------	------------

```
declare
  cursor c_customers is
    select customerid
    from customer;

  cursor c_workers is
    select workerid
    from worker;

  v_customer_id customer.customerid%type;
  v_worker_id worker.workerid%type;
  v_bonus_awarded number;
  discount_info SYS_REFCURSOR;
begin
  -- Loop over customers and update discount
  open c_customers;
  loop
    fetch c_customers into v_customer_id;
    exit when c_customers%notfound;

    -- Call procedure to update customer discount
    update_customer_discount(v_customer_id, discount_info);
  end loop;
  close c_customers;

  -- Loop over workers and update bonus
  open c_workers;
  loop
    fetch c_workers into v_worker_id;
    exit when c_workers%notfound;

    -- Call function to update worker bonus
    v_bonus_awarded := update_bonus_for_worker(v_worker_id);

    -- Print bonus information for the worker
    --dbms_output.put_line('Worker ID ' || v_worker_id || ': Bonus awarded: ' || v_bonus_awarded);
  end loop;
  close c_workers;
end;
```

## הטבלאות customer, worker לפני הרצת התוכנית:

```
select * from customer;
```

	CUSTOMERID	CUSTOMERFNAME	CUSTOMERLNAME	CUSTOMERPHONE	CUSTOMEREMAIL	DISCOUNTPERCENTAGE
4	123456789	John	Doe	551234567	john.doe@example.com	0
5	544125895	Jane	Smith	557654321	jane.smith@example.com	0
6	263933120	Annette	Nolte	508529887	annette.nolte@d	0
7	421988038	Harris	Redgrave	500843066	harris.redgrave	0
8	143136513	Saul	Pollack	503668483	s.pollack@mls.i	0
9	906571901	Lari	Matarazzo	501731458	lari.matarazzo@	0
10	428081030	Etta	Pierce	503456002	etta.p@midwestm	0
11	766023468	Rolando	Kravitz	506398566	rolando.kravitz	0
12	938819453	Meryl	Parsons	506848355	meryl.parsons@c	0
13	945180637	Rose	Judd	508135056	rjudd@unica.uk	0
14	374349078	Ned	Giraldo	507794914	ned.giraldo@sds	0

```
select * from worker;
```

	WORKERID	WORKERFNAME	WORKERLNAME	WORKERPHONE	POSITION	HIREDATE	BONUS
1	123456789	noam	levi	504865985	Manager	01/01/2020	100
2	123568974	chen	naki	526486564	Employee	04/07/2020	0
3	323564871	shai	gamliel	584571323	Manager	04/05/2018	0
4	323135648	noga	okashi	544696812	Employee	06/02/2021	0
5	315556965	avi	shlomi	524493683	Manager	01/01/2018	0
6	226755554	Cesar	Wincott	515565640	Employee	12/06/2017	0
7	743502274	Hikaru	Stevenson	511360250	Manager	17/09/2023	0
8	718109763	Emilio	Pollak	500886993	Manager	19/04/2022	0
9	335673276	Marlon	Moraz	513212401	Employee	11/11/2020	0
10	901683660	Marianne	Shatner	517220436	Manager	29/01/2019	0
11	137920011	Andy	Fender	500238138	Employee	02/02/2022	0

## הטבלאות customer, worker אחרי הרצת התוכנית:

```
select * from customer;
```

	CUSTOMERID	CUSTOMERFNAME	CUSTOMERLNAME	CUSTOMERPHONE	CUSTOMEREMAIL	DISCOUNTPERCENTAGE
1	325625541	tov	avni	586645247	tom@gmail.com	0
2	245556898	dov	barak	524960993	dov@gmail.com	70
3	987564231	nir	jank	507884545	nir@g.com	0
4	123456789	John	Doe	551234567	john.doe@example.com	0
5	544125895	Jane	Smith	557654321	jane.smith@example.com	70
6	263933120	Annette	Nolte	508529887	annette.nolte@d	40
7	421988038	Harris	Redgrave	500843066	harris.redgrave	40
8	143136513	Saul	Pollack	503668483	s.pollack@mls.i	0
9	906571901	Lari	Matarazzo	501731458	lari.matarazzo@	0
10	428081030	Etta	Pierce	503456002	etta.p@midwestm	0
11	766023468	Rolando	Kravitz	506398566	rolando.kravitz	40

```
select * from worker;
```

	WORKERID	WORKERFNAME	WORKERLNAME	WORKERPHONE	POSITION	HIREDATE	BONUS
209	266398616	Red	Parker	514056240	Employee	23/01/2020	0
210	11	Alice	Smith	1234567890	Manager	15/01/2020	100
211	12	Bob	Johnson	-1949288395	Engineer	22/03/2019	50
212	13	Charlie	Williams	-838178284	Technician	30/07/2021	0
213	14	David	Jones	272922827	Clerk	12/11/2018	100
214	15	Eva	Brown	1383933938	Analyst	06/05/2017	200
215	16	Frank	Davis	-1800922247	Supervisor	18/09/2022	100
216	17	Grace	Miller	-699811136	Operator	25/02/2016	50
217	18	Hannah	Wilson	311299975	Coordinator	11/04/2023	0
218	19	Ivy	Moore	422411086	Administrator	19/08/2015	0



## פרוצדורה 2: (מכאן מתחיל החלק השני. פרוצדורה 2, פונקציה 2, תוכנית ראשית 2.)

הפרוצדורה מקבלת מספר הזמנה וID של ספר ויוצרת הזמנה חדשה לטבלת ההזמנות לפי כמות העותקים הזמינים הקיימים במלאי. אנו מעוניינים שתמיד ישאר בסביבות ה-30 עותקים זמינים ולכן נעדכן בהתאם:

אם נשאר פחות מ-5 עותקים זמינים ניצור הזמנה של 30 עותקים חדשים,

אם נשאר פחות מ-10 עותקים זמינים ניצור הזמנה של 20 עותקים חדשים,

אם נשאר פחות מ-20 עותקים זמינים ניצור הזמנה של 10 עותקים חדשים,

אם נשאר פחות מ-30 עותקים זמינים ניצור הזמנה של 0 עותקים חדשים, ובעצם עודכן שלא צריך עוד ספרים.

### הקוד:

```
1 create or replace procedure create_book_order(p_order_id in number, p_book_id in number)
2 is
3     cursor book_cursor is
4         select count(*) as total_available_copies
5         from bookcopy
6         where bookid = p_book_id
7         and status = 'in stock' ;
8
9     v_available_copies number;
10    v_order_quantity number;
11 begin
12     -- Open the cursor
13     open book_cursor;
14     fetch book_cursor into v_available_copies;
15
16     if book_cursor%notfound then
17         -- If no available copies found
18         dbms_output.put_line('no available copies found for book id ' || p_book_id);
19         close book_cursor;
20         return;
21     end if;
22
23     -- Use CASE to determine order quantity
24     v_order_quantity := case
25         when v_available_copies < 5 then 30
26         when v_available_copies < 10 then 20
27         when v_available_copies < 20 then 10
28         when v_available_copies < 30 then 0
29         else 0
30     end;
31
32     -- Create a new order
33     if v_order_quantity > 0 then
34         insert into book_orders (orderid, bookid, availablecopies, orderquantity)
35         values (p_order_id, p_book_id, v_available_copies, v_order_quantity);
36     end if;
37     -- Close the cursor
38     close book_cursor;
39
40     -- Print message about the new order created
41     dbms_output.put_line('new order created: order id = ' || p_order_id || ', book id = ' ||
42         p_book_id || ', available copies = ' || v_available_copies ||
43         ', order quantity = ' || v_order_quantity);
44
45 exception
46     when no_data_found then
47         dbms_output.put_line('no available copies found for book id ' || p_book_id);
48     when others then
49         dbms_output.put_line('error occurred: ' || sqlerrm);
50 end create_book_order;
```

## הרצה של הפרוצדורה עבור הזמנה מס 0 ומס' ספר 112459:

Test script DBMS Output Statistics Profiler Trace

```
1 begin
2   -- Call the procedure
3   create_book_order(p_order_id => :p_order_id,
4                     p_book_id => :p_book_id);
5 end;
```

Variable	Type	Value
p_order_id	Float	1
p_book_id	Float	112463

Test script DBMS Output Statistics Profiler Trace

Clear Buffer size 10000 ☒ Enabled

new order created: order id = 1, book id = 112463, available copies = 2, order quantity = 30

## ניראה את העדכון בטבלה:

```
select * from book_orders;
```

	ORDERID	BOOKID	AVAILABLECOPIES	ORDERQUANTITY
1	9737	111771	0	0
2	0	112459	2	30
3	1	112463	2	30

## פונקציה 2:

פונקציה שמקבלת בכל פעם IDBOOK ומספר סידורי ראשון שממנו תתחיל.

מה שהפונקציה תעשה זה ליצור מספר עותקים חדשים בטבלת העותקים של ספר, לפי המספר המעודכן בטבלת ההזמנות בעמודה, order Quantity בפקודת הכנסה לטבלת העותקים תכניס את ה IDBOOK ובסטטוס תעדכן 'in stock' ובמצב הספר תעדכן 'new' ובכל עותק תגדיל את מספר העותק ב1.

לאחר מכן תעדכן את ההזמנה של אותו הספר בהתאם כך שתאפס את שדה של כמות ההזמנות ותוסיף אותו למספר העותקים הקיים

availableCopies= availableCopies+orderQuantity , ואח"כ 0+orderQuantity ותחזיר את מספר העותקים הקיים במלאי. כתוב את כל הקוד באותיות קטנות

### הקוד:

```
1 create or replace function create_book_copies(p_idbook in number, p_start_serialnumber in number)
2 return number
3 is
4     cursor c_order is
5         select bookid, orderquantity, availablecopies
6         from book_orders
7         where bookid = p_idbook
8         for update;
9
10    r_order c_order%rowtype;
11    v_available_copies number;
12    v_serialnumber number := p_start_serialnumber;
13 begin
14     -- Open the cursor
15     open c_order;
16     fetch c_order into r_order;
17
18     if c_order%notfound then
19         -- If no order found
20         close c_order;
21         raise_application_error(-20001, 'No order found for the given book ID');
22     end if;
23
24     -- Create new copies in the BookCopy table
25     for i in 1..r_order.orderquantity loop
26         insert into bookcopy (serialnumber, status, condition, bookid)
27         values (v_serialnumber, 'in stock', 'new', r_order.bookid);
28         v_serialnumber := v_serialnumber + 1;
29     end loop;
30
31     -- Update availableCopies and reset orderQuantity in book_orders table
32     update book_orders
33     set availablecopies = r_order.availablecopies + r_order.orderquantity,
34     orderquantity = 0
35     where bookid = r_order.bookid;
36
37     -- Close the cursor
38     close c_order;
39
40     -- Fetch the total available copies from the BookCopy table
41     select count(*)
42     into v_available_copies
43     from bookcopy
44     where bookid = p_idbook;
45
46     -- Return the total available copies
47     return v_available_copies;
48 exception
49     when no_data_found then
50         raise_application_error(-20001, 'No order found for the given book ID');
51     when others then
52         raise_application_error(-20002, 'An error occurred: ' || sqlerrm);
53 end create_book_copies;
```

## הרצה עבור BOOKID=112459 ומס סידורי מתחיל מ20000:

Test script DBMS Output Statistics Profiler Trace

```
1 begin
2   -- Call the function
3   :result := create_book_copies(p_idbook => :p_idbook,
4                               p_start_serialnumber => :p_start_serialnumber);
5 end;
```

Variable	Type	Value
result	Float	32
p_idbook	Float	112459
p_start_serialnumber	Float	20000

## הטבלאות לפני ההרצה:

```
select * from book_orders where bookid=112459;
```

	ORDERID	BOOKID	AVAILABLECOPIES	ORDERQUANTITY
1	0	112459	2	30

```
select * from bookcopyy where bookid=112459;
```

	SERIALNUMBER	CONDITION	BOOKID	STATUS
1	12502	new	112459	in stock ***
2	211975	new-used	112459	in stock ***

## הטבלאות לאחר ההרצה:

```
select * from book_orders where bookid=112459;
```

	ORDERID	BOOKID	AVAILABLECOPIES	ORDERQUANTITY
1	0	112459	32	0

```
select * from bookcopyy where bookid=112459;
```

	SERIALNUMBER	CONDITION	BOOKID	STATUS
1	12502	new	112459	in stock ***
2	211975	new-used	112459	in stock ***
3	20000	new	112459	in stock ***
4	20001	new	112459	in stock ***
5	20002	new	112459	in stock ***
6	20003	new	112459	in stock ***

## **תוכנית ראשית 2:**

עבור התוכנית יצרנו טבלה חדשה להזמנות של ספרים. התוכנית עוברת על כל הספרים מעדכנת להם את ההזמנות שיש לבצע ואז יוצרת עותקים נוספים לכל ספר לפי ההזמנה שביצענו.

התוכנית עוברת על כל הספרים ושולחת בכל פעם ספר אחד לפרוצדורה להזמנה ומעדכנת את ההזמנה בהתאם לקריטריונים שפירטנו ולאחר מכן קוראת לפונקציה שתבצע את ההזמנה בפועל ומוסיפה עותקים חדשים ומחזירה את סכום העותקים שיש לכל ספר לאחר העדכון.

שם הקובץ הוא: declareNewOrders.sql

SQL	Output	Statistics
-----	--------	------------

```

declare
  cursor c_books is
    select bookid
    from book;

  r_book c_books%rowtype;
  v_order_id number := 60000; -- starting order ID
  v_start_serialnumber number := 20000; -- starting serial number
  v_total_copies number;
  v_order_quantity number;
  type t_order_record is record (
    orderquantity number
  );
  r_order t_order_record;
  cursor c_orders(p_bookid number) is
    select orderquantity
    from book_orders
    where bookid = p_bookid
    for update;
begin
  -- Open the cursor for books
  open c_books;

  loop
    fetch c_books into r_book;
    exit when c_books%notfound; -- exit if no more books

    -- Call the procedure to create a new book order
    create_book_order(v_order_id, r_book.bookid);

    -- Initialize order quantity
    v_order_quantity := 0;

    -- Open the cursor for orders
    open c_orders(r_book.bookid);
    fetch c_orders into r_order;

    if c_orders%found then
      v_order_quantity := r_order.orderquantity;
    end if;

    close c_orders;

    -- Call the function to create book copies
    if v_order_quantity > 0 then
      v_total_copies := create_book_copies(r_book.bookid, v_start_serialnumber);

      -- Increment order ID and starting serial number for the next iteration
      v_order_id := v_order_id + 1;
      v_start_serialnumber := v_start_serialnumber + v_order_quantity;

    end if;
  end loop;

  -- Close the cursor
  close c_books;
exception
  when others then
    dbms_output.put_line('Error: ' || sqlerrm);
end;
```

## תוכנית נלוות בשם countBaookCopies

עשיתי בדיקה למצוא כמות העותקים יש לכל ספר לפני הרצת התוכנית:

הקוד:

```
declare
  cursor c_books is
    select bookid, bookname
    from book;

  r_book c_books%rowtype;
  v_copy_count number;
begin
  open c_books;

  loop
    fetch c_books into r_book;
    exit when c_books%notfound;

    -- ספירת העותקים עבור הספר הנוכחי
    select count(*)
    into v_copy_count
    from bookcopy
    where bookid = r_book.bookid;

    -- הדפסת כמות העותקים של הספר הנוכחי
    dbms_output.put_line('Book ID: ' || r_book.bookid || ' : ' || v_copy_count);
  end loop;

  close c_books;
end;
```

ההרצה של הקוד:

SQL	Output	Statistics
	Clear	Buffer size 78000 <input checked="" type="checkbox"/> Enabled
Book ID: 112459 : 29		
Book ID: 112463 : 27		
Book ID: 112467 : 16		
Book ID: 112471 : 9		
Book ID: 112475 : 5		
Book ID: 112479 : 9		
Book ID: 112483 : 26		
Book ID: 112487 : 30		
Book ID: 112491 : 9		
Book ID: 112495 : 22		
Book ID: 112499 : 29		
Book ID: 112503 : 23		
Book ID: 112507 : 6		
Book ID: 112511 : 5		
Book ID: 112515 : 6		
Book ID: 112519 : 19		
Book ID: 112523 : 20		
Book ID: 112527 : 14		
Book ID: 112531 : 26		
Book ID: 112535 : 29		
Book ID: 112539 : 1		
Book ID: 112543 : 2		
Book ID: 112547 : 21		
Book ID: 112551 : 7		
Book ID: 112555 : 23		
Book ID: 112559 : 7		
Book ID: 112563 : 13		
Book ID: 112567 : 26		

000 409:1 micah@XE [09:30:24] Done in 0.077 seconds

## הטבלה של הזמנות אחרי הרצה של התוכנית:

```
select * from book_orders;
```

	ORDERID	BOOKID	AVAILABLECOPIES	ORDERQUANTITY
1	60000	22	32	0
2	60001	23	34	0
3	60002	24	21	0
4	60003	27	29	0
5	60004	29	32	0
6	60005	111115	23	0
7	60006	111123	26	0
8	60007	111127	26	0
9	60008	111135	24	0
10	60009	111139	22	0

67:27    michal@XE    [11:05:16] 253 rows selected in 0.432 seconds