מסמך 2 - פרויקט תכנות

מגישים:
ניב סויסה 307929257
ירין מזרחי 205663917

- **לכל פעולה את הפעולה באלגברת יחסים**
- **לכל פעולה את הקוד בSQL שמממש אותה.**

שאילתות מ-SQL:

1. האם ספר X נמצא במלאי ?

```
SELECT
    i.bookID, store_amount, storage_amount
FROM
    inventory i
        JOIN
    All_Books al
WHERE
    i.bookID = al.bookID AND al.bookID = 100;
```

2.מי הוא הלקוח הוותיק ביותר?

```
SELECT
    fname, lname, join_date
FROM
    Customers
WHERE
    join_date = (SELECT
            MIN(join_date)
        FROM
            Customers);
```

3. מה הספר הוותיק ביותר? הספר שנמצא הזמן הרב ביותר במלאי

```
SELECT
    al.title
FROM
    inventory i
        JOIN
    all_books al
        INNER JOIN
    store_purchase s ON i.bookID = s.bookID
WHERE
    i.store_amount > 0
        OR i.storage_amount > 0
ORDER BY s.purchase_date
LIMIT 1
```

4.רשימה הזמנות הנוכחית מסודר לפי תאריכי הזמנה:

```
SELECT
    *
FROM
    reservation r
WHERE
    res_status != 0
ORDER BY r.res_date
```

5.כמה עותקים של ספר Y נמכרו על ידי החנות

```
SELECT
    al.bookID, al.title, COUNT(p.bookID) AS Sold
FROM
    all_books al
        JOIN
    purchase_details p
WHERE
    p.bookID = al.bookID AND p.bookID = ? ;
```

6. מי הסופר הכי נקרא בתווך תאריכים X עד Y

```
SELECT
    author.fname, author.lname
FROM
    author
        INNER JOIN
    author_book ON author.authorID = author_book.authorID
        INNER JOIN
    All_Books ON All_Books.bookID = author_book.bookID
        INNER JOIN
    purchase_details ON purchase_details.bookID = All_Books.bookID
        INNER JOIN
    customer_purchase ON customer_purchase.purchaseID = purchase_details.purchaseID
WHERE
    purchase_date BETWEEN 'X' AND 'Y'
GROUP BY author.authorID
ORDER BY COUNT(author_book.bookID) DESC
LIMIT 1;
```

7.רשימת 3 הלקוחות שרכשו הכי הרבה ספרים לאורך השנים

```
SELECT
    c.fname, c.lname, COUNT(pd.customerID) AS 'bought'
FROM
    purchase_details pd
        JOIN
    Customers c
WHERE
    c.customerID = pd.customerID
GROUP BY pd.customerID
ORDER BY bought DESC
LIMIT 3;
```

8.מי הספר עם מספר התרגומים הגדול ביותר שקיים כרגע במלאי?

```
SELECT DISTINCT
    aa.title, COUNT(aa.translator) AS 'amount of translations'
FROM
    inventory i
        INNER JOIN
    All_Books aa ON aa.bookID = i.bookID
        AND (i.store_amount > 0
        OR i.storage_amount > 0)
        INNER JOIN
    All_Books ab ON aa.title = ab.title
WHERE
    aa.translator < ab.translator
GROUP BY aa.title
ORDER BY COUNT(ab.translator) DESC
LIMIT 1;
```

9.היסטוריית רכישות של לקוח x

```
SELECT
    c.fname, c.lname, cp.purchase_date,
FROM
    purchase_details pd
        INNER JOIN
    customers c ON pd.customerID = c.customerID
        INNER JOIN
    customer_purchase cp ON pd.purchaseID = cp.purchaseID
WHERE
    pd.customerID = ?
    ORDER BY purchase_date;
```

10

.לגבי אדם X אלו ספרים רכש באלו תאריכים ומה המחיר ששילם עבור כל ספר?

```
SELECT DISTINCT
    c.fname,
    c.lname,
    r.bookID,
    r.res_date AS 'reservation date',
    r.res_status AS 'reservation status'
FROM
    reservation r
        INNER JOIN
    customers c ON c.customerID = r.customerID
        AND r.customerID = 'X'
ORDER BY res_date DESC;
```

11.חישוב עלות משלוח.

```
SELECT
    purchaseID,
    SUM((weight * cost_per_kilo)) AS 'shipment taarif'
FROM
    (SELECT
        al.bookID, pd.purchaseID, al.weight, di.cost_per_kilo
    FROM
        All_books al
    INNER JOIN purchase_details pd USING (bookID)
    INNER JOIN customer_purchase cp USING (purchaseID)
    INNER JOIN deliveries d ON cp.purchaseID = d.deliveryID
    INNER JOIN delivery_info di USING (delivery_info_id)) AS temp
GROUP BY purchaseID
ORDER BY purchaseID;
```

12האם לקוח X פיצל אי פעם רכישת ספרים למספר משלוחים ?

```
SELECT
    c.customerID, c.fname, c.lname, d.address, cp.payment_method
FROM
    customer_purchase cp,
    customer_purchase cp1
        INNER JOIN
    customers c ON cp1.customerID = c.customerID
        INNER JOIN
    deliveries d ON cp1.purchaseID = d.deliveryID
WHERE
    c.fname = '?'
        AND c.lname = '?'
        AND cp.customerID = cp1.customerID
        AND cp.purchase_date = cp1.purchase_date
        AND cp.purchaseID != cp1.purchaseID;
```

13.מה הוא הסטטוס הנוכחי של משלוח מסויים?

```
SELECT
    d.tracking_num, d.status
FROM
    deliveries d
WHERE
    d.tracking_num = '?';
```

14. מה סכום המשלוחים שבוצעו על ידי חברת xpress בחודש מסויים.

```
    SELECT DISTINCT
    di.company, COUNT(d.delivery_info_id) AS 'delivery amount'
FROM
    deliveries d
        INNER JOIN
    delivery_info di USING (delivery_info_id)
        INNER JOIN
    customer_purchase cp ON cp.purchaseID = d.deliveryID
WHERE
    YEAR(cp.purchase_date) = year
        AND MONTH(cp.purchase_date) = month
        AND di.delivery_info_id BETWEEN '4' AND '5';
```

15.סך הכסף שהעובר לחשבון החנות באמצעות אפליקציית bit בחודש מסויים?

```
SELECT
    payment_method AS 'Payment method',
    SUM(al.price) AS 'total from Bit'
FROM
    customer_purchase cp
        INNER JOIN
    purchase_details pd USING (purchaseID)
        INNER JOIN
    All_Books al USING (bookID)
WHERE
    payment_method = 'Bit app'
        AND MONTH(purchase_date) = '?'
        AND YEAR(purchase_date) = '?';
```

16.מהנה עסקאות שבוצעו במהלך 12 החודשים האחרונים

```
    SELECT
    cp.purchase_date,
    cp.purchaseID,
    (al.price - sp.book_price) AS income
FROM
    store_purchase sp
        INNER JOIN
    All_Books al USING (bookID)
        INNER JOIN
    purchase_details pd USING (bookID)
        INNER JOIN
    customer_purchase cp USING (purchaseID)
WHERE
    (al.price - sp.book_price) > (SELECT
            AVG((al.price - sp.book_price)) AS average
        FROM
            store_purchase sp
                INNER JOIN
            All_Books al USING (bookID)
                INNER JOIN
            purchase_details pd USING (bookID)
                INNER JOIN
            customer_purchase cp
        WHERE
            cp.purchase_date BETWEEN DATE_SUB(CURRENT_DATE(), INTERVAL 1 YEAR) AND CURRENT_DATE())
        AND (cp.purchase_date BETWEEN DATE_SUB(CURRENT_DATE(),INTERVAL 1 YEAR) AND CURRENT_DATE());
```

17.כמה משולחים בוצעו במהלך 12 החודשים האחרונים באמצעות דואר ישראל וכמה בוצעו באמצעות חברת xpress

```
SELECT
    di.company, COUNT(d.delivery_info_id) AS 'num of deliveries'
FROM
    deliveries d
        INNER JOIN
    delivery_info di USING (delivery_info_id)
        INNER JOIN
    customer_purchase cp ON cp.purchaseID = d.deliveryID
WHERE
    cp.purchase_date BETWEEN DATE_SUB(CURRENT_DATE(),
        INTERVAL 1 YEAR) AND CURRENT_DATE()
        AND di.company != 'store'
GROUP BY company;
```

19.נתונים על כל הלקוחות שרכשו בערב לפחות ספר אחד מהחנות ולא ביצעו רכישה במהלך ה24 חדושים האחרונים

```
SELECT
    c.customerID, c.fname, c.lname, cp.purchase_date
FROM
    customer_purchase cp
        INNER JOIN
    customers c USING (customerID)
WHERE
    customerID NOT IN (SELECT
            customerid
        FROM
            customer_purchase
        WHERE
            cp.purchase_date BETWEEN DATE_SUB(CURRENT_DATE(),
                INTERVAL 2 YEAR) AND CURRENT_DATE());
```

20.רשימת הלקוחות שביצעו הזמנות,הספרים שהזימנו הגיעו לחנות, החנות יצרה איתם קשר ליידע אותם על הזמינות הקשר נוצר לפני14 ימים והלקוחות עדיין לא רכשו את הספר

```sql
SELECT
    r.reservationID,
    c.fname,
    c.lname,
    r.last_contacted,
    r.res_status
FROM
    reservation r
        INNER JOIN
    customers c USING (customerID)
WHERE
    CURRENT_DATE() > last_contacted + 14
        AND res_status != 'item purchased';
```

21.מספר הספרים במחסן בחתך חודשי

```sql
SELECT
    YEAR(purchase_date),
    MONTH(purchase_date),
    SUM(i.storage_amount)
FROM
    store_purchase sp
        INNER JOIN
    inventory i USING (bookID)
GROUP BY purchase_date;
```

22. כמה ספרים רכשה החנות בין תאריך d1 ל d2 ומה היה סך התשלום עבורם

```sql
SELECT
    SUM(i.store_amount + storage_amount) AS 'amount of books',
    SUM(sp.book_price * (i.store_amount + i.storage_amount)) AS 'total amount'
FROM
    store_purchase sp
        INNER JOIN
    inventory i USING (bookID)
WHERE
    purchase_date BETWEEN '?' AND '?';
```

23. רווח החנות ממכירות בחודש מסוים .

```
SELECT
    (SUM(al.price) - (SELECT
            SUM(sp.book_price * (i.store_amount + i.storage_amount)) AS 'total prof'
        FROM
            store_purchase sp
                INNER JOIN
            inventory i USING (bookID)
        WHERE
            YEAR(sp.purchase_date) = Y
                AND MONTH(sp.purchase_date) = M)) AS 'profit'
FROM
    All_Books al
        INNER JOIN
    purchase_details ps USING (bookID)
        INNER JOIN
    customer_purchase cp USING (purchaseID)
WHERE
    YEAR(cp.purchase_date) = ?
        AND MONTH(cp.purchase_date) = ? ;
```

24. ממוצע עסקאות שנתי בחתך חודשי

```
SELECT
    YEAR(cp.purchase_date) AS year,
    SUM(al.price) / 12 AS 'average income'
FROM
    customer_purchase cp
        INNER JOIN
    purchase_details pd USING (purchaseID)
        INNER JOIN
    All_Books al USING (bookID)
WHERE
    YEAR(purchase_date)
GROUP BY YEAR(purchase_date)
ORDER BY YEAR(purchase_date) DESC;
```

25.משכורת ברוטו של עובד z בחודש מסויים

```sql
SELECT
    wt.year, wt.month, e.fname, e.lname, wt.salary_for_month
FROM
    Employee e
        INNER JOIN
    work_time wt USING (employeeID)
WHERE
    wt.year = year AND wt.month = month
        AND e.employeeID = employeeID;
```

26. מי המוכר עם הכי הרבה עסקאות בחודש X

```sql
SELECT
    e.fname, e.lname, COUNT(employeeID)
FROM
    employee e
        INNER JOIN
    customer_purchase cp USING (employeeID)
WHERE
    YEAR(purchase_date) = ?
        AND MONTH(purchase_date) = ?
GROUP BY employeeID
ORDER BY employeeID DESC
LIMIT 1;
```

אלגברת יחסים :

1.
Π (inventory.bookID, inventory.store_amount , inventory.storage_amount (inventory)⋈ inventory.bookID = All_Books.bookID (All_Books)) AND All_Books.bookID = ?

2.
Π(customers.fname,customers.lname(Customers))Customers.join_date = (Π min(customers.join_date) (customers)

3.
Π(All_Books.title (All_Books)⋈All_Books.bookID = inventory.bookID(inventory) inventory.store_amount>0 OR inventory.storage_amount > 0)

4.
Π(reservation) (reservation.res_status != 0
Order by.reservation.res_date)

5.
Π(all_books.bookID,all_books.title,count(purchase_Details.bookID)as sold(all_books))⋈ purchase_details(purchase_Details.bookID = all_books.bookID AND Purchase_Details.bookID = 'Y')

6.
Π(author.fname , author.lname (author)⋈author.authorID = author_book.authorID (author_book) ⋈ author_book.bookID = All_Books.bookID (All_Books)⋈ All_books.bookID = purchase_details.bookID (purchase_details)⋈ purchase_details.purchaseID = customer_purchase.purchaseID (customer_purchase) customer_purchase between ? and ?)

7.
Π(customer.fname,customers,lname,count(purchase_details.customerID)ρ bought Π(purchase_Details))⋈customer(customer.customerID = purchase_Details.customerID
Group by purchase_details.customerID
Order by(bought,desc)
Limit 3)

8.
Π(all_books.title,count(all_book.translator ρ amount of translations (inventory)⋈ all books all_Books.bookID = inventory.bookID AND inventory.store_amount >0 or inventory.storage_amount >0)⋈ all books all_books.title = all_Books.title
(all_books_=.translator < all_books.translator)
Group by all_Books.title
Order by count(all_books.translator,desc) limit1)

9.
Π(customers.fname , customers.lname , customer_purchase.purchase_date(customer)⋈customers.customerID = customer_purchsae.customerID
(customer_purchase)⋈customer_purchse.purchaseID = purchase_details.purchaseID (purchaseID) purchase_details.customerID = ?)

10.
Π(customers.fname,customer.lname,reservation.bookID,reservation.res_date, reservation.res_status (reservation))⋈ customers
customers.customerID = reservation.customerID and
reservation.customer ID = ?
Order by (reservation.res_Date ,desc))

11.
Π(purchase_details.purchaseID , sum((All_Books.weight * delivery_info.cost_per_kilo)) ρ ‘shipment taarif’ Π(All_Books.bookID, purchase_details.purchaseID, All_Books.weight , delivery_info.cost_per_kilo (All_Books)⋈ All_Books.bookID = purchase_details.bookID (purchase_details) ⋈ purchase_details.purchaseID = customer_purchase.bookID (customer_purchase) ⋈ customer_purchase.purchaseID = deliveries.deliveryID (deliveries) ⋈ deliveries.delivery_info_id = delivery_info.delivery_info_id (delivery_info)) ρ temp

12.
Π(customers.customerID,customers.fname,customers.lname,deliveries.address,customer_purchae.payment_method (customer_purchase,customer_purchase)⋈ customers customers_purcahse.customerID = customers.customerID ⋈ deliveries customer_purchase.purchaseID = deliveries.deliveryID(customer.fname = ? AND customers.lname = ? AND customer_purchase.customerID = customer_purchase.customerID AND customer_purchase.purchaseID != customer_purchase.purchaseID))

13.
Π(deliveries.tracking_num,deliveries.status(deliveries)(deliveries.tracking_num =?)

14.
Π(deliveries.company,count(deliveries.delivery_info_id)(deliveries)⋈ delivery_info delivery.delivery_info_id = delivery.delivry_info_id ⋈ customer_purchase customer_purchase.purchaseID=deliveries.delivryID(year(customer_purchase.customer_date)= ?
AND month(customer_purchase.purchase_date = ? AND delivery_info.delivery_info_id between ‘4’ AND ‘5’))

15.

Π(customer_purchase.payment_method ρ ‘payment_method’ ,
sum(All_Books.price) ρ ‘total from bit’ (customer_purchase)⋈
purchase_details.purchaseID = customer_purchase.purchaseID
(purchase_details) ⋈ customer_purchase.bookID = All_Books.bookID
(All_Books) payment_method = ‘Bit app’ AND
month(purchase_details.purchase_date) = ? and
year(purchase_details.purchase_date) =?)

16.

Π(customer_purchae.purchase_Date,customer_purchase.purchaseID,
all_book.price - store_purchase.book_price ρ income(store_pruchase)⋈
all_books all_books.bookID = store_purchase.bookID ⋈ purchase_details
purchase_details.bookID = all_books.bookID⋈ customer_purchase on
customer_purchase.purchaseID =
store_purchase.purchaseID(all_Books.price-store_purchase.book_price >
(Π AVG(all_books.price-store_purchase.book_price)ρ average (
store_pruchase ⋈ all_books all_books.bookID = store_purchase.bookID
⋈purchase_details purchase_details.bookID = all_book.bookID ⋈
customer_purchase (customer_purchase.purchase_date between
DATE_SUB(CURRENT_DATE(),
AND(customer_purchasep.purchase_date BETWEEN
DATE_SUB(CURRENT_DATE(), INTERVAL 1 YEAR) AND
CURRENT_DATE());
INTERVAL 1 YEAR) AND CURRENT_DATE (

17.
Π(delivery_info.company , count(deliveries.delivery_info_id) ρ ‘num of deliveries’ (deliveries) ⋈ deliveries.delivery_info_id = delivery_info.delivery_info_id (delivery_info)⋈ delivery.deliveryID = customer_purchase.purchaseID (customer_purchase) customer_purchase.purchase_date between date_sub(current_date() , interval 1 year) and current_date() and delivery_info.company != ‘store’)

19.
Π(customers.customerID,customers.fname,customers.lname,customer_purchass.purchase_date(customer_purchase)⋈ customers customers.customerID = customer_purchase.customerID(customerID NOT in (Π(customerID(customer_purchase)(customer_purchase.purchase_Date between DATE_SUB(CURRENT_DATE(), INTERVAL 2 YEAR) AND CURRENT_DATE

20
.Π(reservation.reservationID,customers.fname,customers.lname,reservation.last_contacted,reservation.res_status(from reservation ⋈ customers customer.customerID = reservation.customerID(current_date() > last_contacted +14 AND res_status != ‘item purchased’)

21.
Π(year(store_purchase.purchase_date) , month(store_purchase.purchase_date), sum(inventory.storage_amount)(store_purchaes) ⋈ store_purchase.bookID = inventory.bookID (inventory))

22.
Π(inventory.store_amount + inventory.storage_amount) ρ amount of books (store_purchase ⋈ inventory inventory.bookID= store_purchase.bookID(purchase_date between ? AND ?)

23.
Π(sum((All_Books.price)-Π(sum(store_purchase.book_price *
(inventory.store_amount +inventory.storage_amount)) ρ ‘total prof’
(store_purchase) ⋈ store_purchase.bookID = inventory.bookID (inventory)
year(store_purchase.purchase_date) = ? and
month(store_purchase.purchase_date) = ?)ρ ‘profit’ (All_Books)⋈
All_Books.bookID = purchase_details.bookID (purchase_details) ⋈
purchase_details.purchaseID = customer_purchase.purchaseID
(customer_purchase) year(customer_purchase.purchase_date) = ? AND
month(customer_purchase.purchase_date) = ?)

24.
Π(YEAR(customer_purchase.purchase_date) ρ year SUM(all_books.price)/
12 ρ average income (customer_purchase ⋈ purchase_details
purchase_details.purchaseID = customer_purchase.purchaseID)

25.
Π(work_time.year,work_time.month, employee.fname,employee.lname
,work_time.salary_for_month(employee ⋈ work_time
work_time.employeeID = employee.employeeID (work_time.year = ? AND
work_time.month = ? AND employee.employeeID = ?)

26.
Π(employee.fname,employee.lname,count(employee.emoloyeeID)(employ
ee ⋈ customer_purchase customer_purchase.employeeID =
employee.employeeID(year(purchase_date) = ? AND
month(purchase_date) =?
Group by employeeID
Order by (employeeID ,desc)
limit1)