SHENKAR —— שנקר
ENGINEERING. DESIGN. ART . הנדסה. עיצוב. אמנות
The Pernick Faculty of Engineering . הפקולטה להנדסה ע"ש פרניק
המחלקה להנדסת תוכנה
Software Engineering Dept.

Students: Idan Kario 300853751 Coral Rubilar 316392877

Email: idankario@gmail.com coralrubilar@gmail.com

ההנחה שלנו ל-ERD

* טבלאות של DB : לקוחות, הזמנות, ספקים, תשלומים, משלוחים, מתרגמים, לקוחות, מחברים, מוציאים לאור, מלאי ספרים ועובדים

סיסמא למנהל: 1234 בהרצה.

הזמנות\תשלומים	דיווח עבור הוצאות ורווח כלומר הזמנות יחושב בנפרד כדי שדיווחים כמו מאזן שנתי וכו יוכלו להתבצע בקלות
מלאי מחסן	מתעדכן כל יום ונמצא מעודכן ביום בדיקת הנתונים
ספרים	ספירים יכולים להיכתב ע"י כמה סופרים\מתרגמים וכמה מתרגמים\סופרים יכולים לכתוב הרבה ספרים
מחסן	מחזיק מלאי של ספר ומציג את המצב הנוכחי של הספר בלוי חדש וכו... גם של החנות וגם של המחסן כדי שיהיה נוח לעדכן בכל יום את המלאי ולשלוח לחנות מלאים בהתאם
הוצאה לאור	הוצאה לאור יוצרת את האובייקט ספר אחרי שהסופר מסיים לכתוב

	אותו ואז מחליטה האם לתרגם את הספר
ספר	כל הספרים במחסן מסווגים לשני זהויות ID של הספר והמצב שבו הוא נמצא המחיר לספר הוא גם לחנות וגם למשלוח דרך המחסן
רשימות מלאי	במכירות קיים תיעוד לגבי כל הקניות שבוצעו כך גם בקנייה של הספר ולכן אין תיעוד נוסף להעברה של הספרים
משלוח	לקוח יכול להזמין ספר אחד או כמה ספרים מאותו סוג
משלוח	לקוח יכול לפצל משלוח לכמה אזורים ולבחור איזה ספרים לשלוח במשלוח
הוצאה לאור	יכול להיות מספר הוצאות לאור לספר אחד
מנהל	מנהל יכול לבצע פעולות שונות מעובד רגיל ולגשת למידע כללי לגבי החנות
ספר	כל ספר מקבל ID יחודי בהתאם לשנת הוצאה לאור
סיסמת מנהל	1234 כמובן שלא משקף את המציאות רק לצורך נוחות העבודה
מחסן	במחסן זיהוי הספר יתבצע באמצעות מצב הספר וID שלו

פעולות SQL

```sql
/* 1 */
SELECT
    bf.title_book As TitleBook,
    concat(a.first_name,' ',a.last_name) AS AutorsName ,
    SUM(i.amount_warehouse_inventory) WarehouseInventory,
    SUM(i.amount_store_inventory) AmountStore
FROM  inventory As i
JOIN   (books_features As bf, autors As a ,autors_books_features abf)
    ON  (bf.books_features_id= abf.books_features_id
        AND a.autor_id = abf.autor_id
        AND abf.book_id = i.book_id)
GROUP BY bf.title_book
HAVING  AutorsName = "Robert Greene" AND TitleBook = "The Laws of Human Nature";
```

```sql
/*2*/
SELECT *
FROM clients
ORDER BY join_date  LIMIT 1 ;
```

```sql
/*3*/
SELECT
    bf.title_book As TitleBook, ps.purchase_date As DateInTheStore
FROM inventory As i
JOIN (books_features As bf, purchase_store As ps,autors_books_features abf)
    ON (abf.books_features_id = bf.books_features_id
        AND i.Book_id = abf.Book_id )
WHERE i.amount_store_inventory + i.amount_warehouse_inventory > 0
ORDER BY DateInTheStore  LIMIT 1;
```

```sql
/*4*/
SELECT
    bf.title_book As TitleBook,
    concat(a.first_name,' ',a.last_name) AS AutorsName,
    o.order_state,
    o.orderbooks_date as OrederDate
FROM orders As o
JOIN (books_features As bf, autors As a, autors_books_features  abf )
    ON (o.order_state = 'Ordered'
        AND o.Book_id=abf.Book_id
        AND bf.books_features_id = abf.books_features_id
        AND a.autor_id = abf.autor_id)
ORDER BY OrederDate;
```

```sql
/*5*/
SELECT
    bf.title_book As TitleBook,
    concat(a.first_name,' ',a.last_name) AS AutorsName ,
    SUM(bc.amount_of_books) Sold
FROM  purchase_client As pc
JOIN   (books_features As bf, autors As a , book_client bc)
INNER JOIN( SELECT  *
            FROM  autors_books_features
            GROUP BY book_id
           )abf
    ON  (bf.books_features_id = abf.books_features_id
        AND a.autor_id = abf.autor_id
        AND abf.Book_id = bc.book_id
        AND pc.purchase_client_id = bc.purchase_client_id)
        GROUP BY AutorsName,TitleBook
HAVING  TitleBook='The Betrayal';
```

```sql
/*6*/

SELECT
    concat(a.first_name,' ',a.last_name) AS AutorsName,
    SUM(bc.amount_of_books) Sold
FROM purchase_client As pc
JOIN (books_features As bf, autors As a ,autors_books_features abf, book_client bc)
    ON  (bf.books_features_id = abf.books_features_id
        AND a.autor_id = abf.autor_id
        AND abf.Book_id = bc.Book_id
        AND pc.purchase_client_id=bc.purchase_client_id)
WHERE pc.purchase_date BETWEEN CAST('2000-07-02'  AS DATE) AND CAST('2025-05-02' AS DATE)
GROUP BY AutorsName
ORDER BY Sold DESC LIMIT 1;
```

```sql
/*7*/

SELECT
    concat(c.first_name,' ',c.last_name) CustomerName,
    SUM(bc.amount_of_books) AmountBuyBooks
FROM purchase_client As pc
JOIN (clients As c,book_client bc)
    ON (c.client_id=pc.client_id
        AND bc.purchase_client_id=pc.purchase_client_id)
GROUP BY CustomerName
ORDER BY AmountBuyBooks DESC LIMIT 3;
```

```sql
/* 8 */
SELECT
    abf.book_id,
    bf.title_book As TitleBook,
    count(DISTINCT b.language_book) AmountLanguageTrnslate
FROM inventory i
JOIN (books AS b, autors_books_features abf,books_features As bf)
    ON (b.book_id= abf.book_id
        AND bf.books_features_id = abf.books_features_id
        AND i.amount_store_inventory + amount_warehouse_inventory>0)
GROUP BY TitleBook
ORDER BY AmountLanguageTrnslate  DESC  LIMIT 1;
```

```sql
-- /* 9 */
SELECT
    concat(c.first_name,' ',c.last_name) AS CustomerName,
    pc.purchase_client_id  InvoiceNumber,
    bf.title_book As TitleBook,
    amount_of_books AmountBuyBooks ,
    FLOOR(pc.total_price/amount_of_books)  PriceForEachBook,
    pc.total_price TotalPrice,
    purchase_date DateBuy
FROM purchase_client As pc
JOIN (books_features As bf,clients As c,book_client bc, autors_books_features abf)
    ON  (bf.books_features_id= abf.books_features_id
        AND bc.book_id=abf.book_id
        AND c.client_id=pc.client_id
        AND bc.purchase_client_id=pc.purchase_client_id)
HAVING CustomerName = 'coral rubilar';
```

```sql
/* 10 */

SELECT
    orderbooks_date DateOrder, order_state,
    bf.title_book TitleBook
from orders o
JOIN (clients c,books_features as bf,autors_books_features as abf)
    on (o.client_id = c.client_id
        AND o.Book_id=abf.Book_id
        AND bf.books_features_id = abf.books_features_id)
WHERE c.first_name = 'coral' and c.last_name = 'rubilar'
ORDER BY DateOrder;
```

```sql
/*11*/

SELECT
    dt.company Company,
    SUM(dt.price_per_kg *bf.weight + db.amount_of_books*dt.price_per_book) TotalPriceDelivery
FROM deliveries d
JOIN (delivery_type AS dt, delivery_book AS db  ,books_features AS bf,autors_books_features abf)
    ON (db.delivery_id = d.delivery_id
        AND abf.book_id=db.book_id
        AND bf.books_features_id = abf.books_features_id
        AND dt.delivery_type_id= d.delivery_type_id )
WHERE(d.delivery_id=6)
GROUP BY Company;
```

```sql
/* 12 */

SELECT
    d.delivery_id IDDelivery,
    d.purchase_client_id IDPurchase,
    d.traking_num TrakingNum,
    d.address Adress,
    d.delivary_condition DelivaryCondition,
    dt.d_type TypeOfDelivery,
    d.delivery_date DeliveryDate,
    concat(c.first_name,' ',c.last_name) CustomerName,
    c.home_num HomeNum,
    c.phone_num PhoneNum
FROM delivery_type dt
JOIN (clients AS c ,
     purchase_client pc, deliveries d)
     INNER JOIN( SELECT  *
            FROM  delivery_book dbc
            GROUP BY dbc.delivery_id
           )dbc
    ON (dt.delivery_type_id=d.delivery_type_id
        AND c.client_id = pc.client_id
        AND pc.purchase_client_id=d.purchase_client_id
        AND dbc.delivery_id=d.delivery_id)
GROUP BY IDPurchase
HAVING COUNT(distinct d.delivery_id) > 1 AND CustomerName = 'Coral rubilar'
ORDER BY IDDelivery, d.purchase_client_id;
```

```sql
/* 13 */

SELECT
    traking_num NumberTraking,
    d.delivary_condition StatusDelivery
FROM   deliveries as d
WHERE d.traking_num='AN2345';
```

```sql
/* 14*/

SELECT
    MONTH(d.delivery_date) MonthCheckDelivery,
    count(distinct d.delivery_id) NumberOfDelivery
FROM deliveries d
JOIN delivery_type dt
    on d.delivery_type_id = dt.delivery_type_id
WHERE dt.company = 'Xpress' AND YEAR(d.delivery_date)=YEAR(CURDATE()) AND MONTH(d.delivery_date) = 8 ;
```

```sql
/* 15*/

SELECT
    MONTH(p.purchase_date) MonthCheckPay,
    sum(p.total_price) AS TotalMoneyPayBybit
From purchase_client p
WHERE p.purchase_type='Bit' AND YEAR(p.purchase_date)=YEAR(CURDATE()) AND MONTH(p.purchase_date)=8 ;
```

```sql
/* 16 */

SELECT *
FROM purchase_client AS pc
Where pc.purchase_date >= CURDATE() - INTERVAL 12 MONTH
    AND (SELECT AVG(purchase_client.total_price)
        FROM purchase_client) < pc.total_price;
```

```sql
/* 17 */

SELECT
    dt.company, count(*)
from deliveries d
JOIN delivery_type dt
    ON d.delivery_type_id = dt.delivery_type_id
WHERE  d.delivery_date >= CURDATE() - INTERVAL 12 MONTH AND (dt.company='Xpress' or dt.company='Israel Post')
group by  dt.company;
```

```
/* 18 */
SELECT
    d.delivery_id IDDelivery,
    d.purchase_client_id IDPurchase,
    d.traking_num TrakingNum,
    d.address Adress,
    d.delivary_condition DelivaryCondition,
    d.delivery_date DeliveryDate,
    concat(c.first_name,' ',c.last_name) CustomerName,
    c.home_num HomeNum,
   c.phone_num PhoneNum,
   bf.title_book As TitleBook,
    COUNT(DISTINCT b.publishe_year ) AmountEditions
FROM deliveries d
JOIN (books AS b,autors AS a, clients AS c ,
     books_features AS bf ,purchase_client pc,delivery_book: db )
INNER JOIN( SELECT  *
            FROM  autors_books_features
            GROUP BY book_id
           )abf
    ON (db.book_id = b.book_id
        AND c.client_id = pc.client_id
        AND a.autor_id=abf.autor_id
        AND abf.book_id = b.book_id
        AND abf.books_features_id = bf.books_features_id
        AND pc.purchase_client_id=d.purchase_client_id
        AND db.delivery_id=d.delivery_id)
GROUP BY TitleBook
HAVING COUNT(DISTINCT b.publishe_year  ) > 1
ORDER BY IDDelivery, d.purchase_client_id;
```

```
/* 19 */

SELECT *
FROM clients c
JOIN purchase_client pc
    ON  c.client_id = pc.client_id
WHERE pc.purchase_date < CURDATE() - INTERVAL 24 MONTH;
```

```sql
/* 20 */

SELECT
    concat(c.first_name,' ',c.last_name) CustomerName,
    o.contact_date ConantDate,
    DATE_SUB(CURDATE(), INTERVAL 14 DAY) Contant14DaysAgo
FROM orders o
JOIN clients c
    ON o.client_id = c.client_id
WHERE  o.order_state='Available'
HAVING ConantDate <= Contant14DaysAgo ;
```

```sql
-- /*21 */
SELECT
    CURDATE() - INTERVAL 12 MONTH FromDate,
    CURDATE() UntilDate,Year(pc.purchase_date) YearCheck,
    MONTHNAME(pc.purchase_date ) MonthCheck,
    (i.amount_warehouse_inventory+count(bc.amount_of_books)-count(bc.amount_of_books))/DAY(LAST_DAY(pc.purchase_date)) AverageNumOfBook
FROM inventory AS i
JOIN( book_client bc,purchase_store pstore, purchase_client pc)
group by MonthCheck;
```

```sql
 /*22 a*/
SELECT

    SUM(pstore.amount_of_books) TotalBooksPurchased,
    SUM(pstore.total_price) TotalPrice
FROM purchase_store pstore
WHERE pstore.purchase_date >= '2018-05-02' AND YEAR(pstore.purchase_date) <= '2020-08-01';
```

```sql
/*22 b*/

SELECT
    MONTH(pstore.purchase_date) MonthCheck,
    YEAR(pstore.purchase_date) YearCheck,
    SUM(pc.total_price)- SUM(pstore.total_price)
TotalProfit
FROM purchase_store pstore
JOIN purchase_client pc
WHERE MONTH(pstore.purchase_date) ='05'  AND MONTH(pc.purchase_date) = '05'
      AND  YEAR(pstore.purchase_date) = '2018' AND YEAR(pc.purchase_date)= '2018' ;
```

```sql
/* 23 */

SELECT
    CURDATE() - INTERVAL 12 MONTH FromDate,
    CURDATE() UntilDate,Year(pc.purchase_date) YearCheck,
    MONTHNAME(pc.purchase_date) MonthCheck,
    count(pc.purchase_client_id)/DAY(LAST_DAY(pc.purchase_date)) AverageTransaction
FROM purchase_client pc
WHERE CURDATE() - INTERVAL 12 MONTH < pc.purchase_date  AND pc.purchase_date <  CURDATE()
GROUP BY  MonthCheck
ORDER BY pc.purchase_date;
```

```sql
/* 24 */

SELECT
    w.worker_id ,
    concat(w.first_name ,' ',w.last_name ) WorkerName,
    wh.total_hours TotalHours,
    w.hourly_wage HourlyWage,
    w.hourly_wage * wh.total_hours GrossSalary
FROM workers w
JOIN work_hours wh
    ON w.worker_id = wh.worker_id
WHERE w.worker_id=1 AND wh.work_year=YEAR(CURDATE()) AND wh.work_month = 07;
```

```sql
/* 25 */
SELECT
    p.worker_id ,
    concat(w.first_name ,' ',w.last_name ) WorkerName ,
    count(*)  NumberTranslations
FROM purchase_client p
JOIN workers w
    on w.worker_id = p.worker_id
WHERE  YEAR(p.purchase_date) = '2019' AND MONTH(p.purchase_date)='08'
GROUP BY worker_id
ORDER BY worker_id LIMIT 1;
```