

מיני פרויקט בבסיסי נתונים

מערכת לניהול חנות מקוונת



רז אברג'ל-313575185

אבי יאלאו-316606094

שלב 1

תיאור המערכת:

המערכת מנהלת מסד נתונים עבור חנות מסחר מקוונת. המערכת מנהלת מידע של מוכרים, קונים, מוצרים, קטגוריות, הזמנות, פרטי הזמנה, ביקורות והקשרים בין הישויות הללו. המערכת מאפשרת מעקב אחר מכירת מוצרים, ביצוע הזמנות, ביקורות לקוחות ופרטי מוכרים.

הישויות בבסיס הנתונים:

1. Sellers - מוכרים

seller_id

seller_name

phone

address

2. Categories - קטגוריות של מוצרים

category_id

category_name

information

3. OrderDetails - פרטי הזמנה

Order_id

Tracking_number

Order_status

delivery_method

בס"ד

4. Buyers - קונים

buyer_id

buyer_name

email

phone

address

5. Products - מוצרים

product_id

product_name

status

price

category_id

stock

available_date

6. Orders - הזמנות

order_id

order_date

product_id

7. Reviews - ביקורות

review_id

rating

comment_text

review_date

product_id

buyer_id

8. sell - מכירה

seller_id

product_id

9. include products - מוצר בהזמנה

order_id

product_id

quantity

קשרים בין הישויות השונות:

1. Categories and Products (קטגוריות ומוצרים):

- מערכת יחסים של אחד לרבים (one-to-many)

- קטגוריה יכולה לכלול מספר מוצרים, אך כל מוצר שייך לקטגוריה אחת בלבד

- מיושם על ידי מפתח זר category_id בטבלת המוצרים המתייחסת לקטגוריות.

2. Buyers and Orders (קונים והזמנות):

- מערכת יחסים של אחד לרבים (one-to-many).

- קונה יכול לבצע מספר הזמנות, אך כל הזמנה משויכת לקונה אחד בלבד בדיוק (חץ עגול)

- מיושם על ידי מפתח זר buyer_id בטבלת ההזמנות המתייחסת לקונים

3. OrderDetails and Orders (פרטי הזמנה והזמנות):

- OrderDetails ישות חלשה של orders

- מערכת יחסים של אחד לאחד (one-to-one).

- כל פרט הזמנה משויך להזמנה אחת

- מיושם על ידי מפתח הזר order_id שהוא מפתח ראשי של orderDetails

4. Products and Reviews (מוצרים וסקירות):

- מערכת יחסים של אחד לרבים (one-to-many) .

- למוצר יכולים להיות מספר ביקורות, אך כל ביקורת היא למוצר אחד בלבד בדיוק (חץ עגול)

- מיושם על ידי מפתח זר product_id בטבלת ביקורות המתייחסת למוצרים

5. Buyers and Reviews (קונים וסקירות):

- מערכת יחסים של אחד לרבים (one-to-many).
- קונה יכול לכתוב מספר ביקורות, אך כל ביקורת נכתבת על ידי קונה אחד בלבד בדיוק (חץ עגול)
- מיושם על ידי מפתח זר buyer_id בטבלת ביקורות המתייחסת לקונים

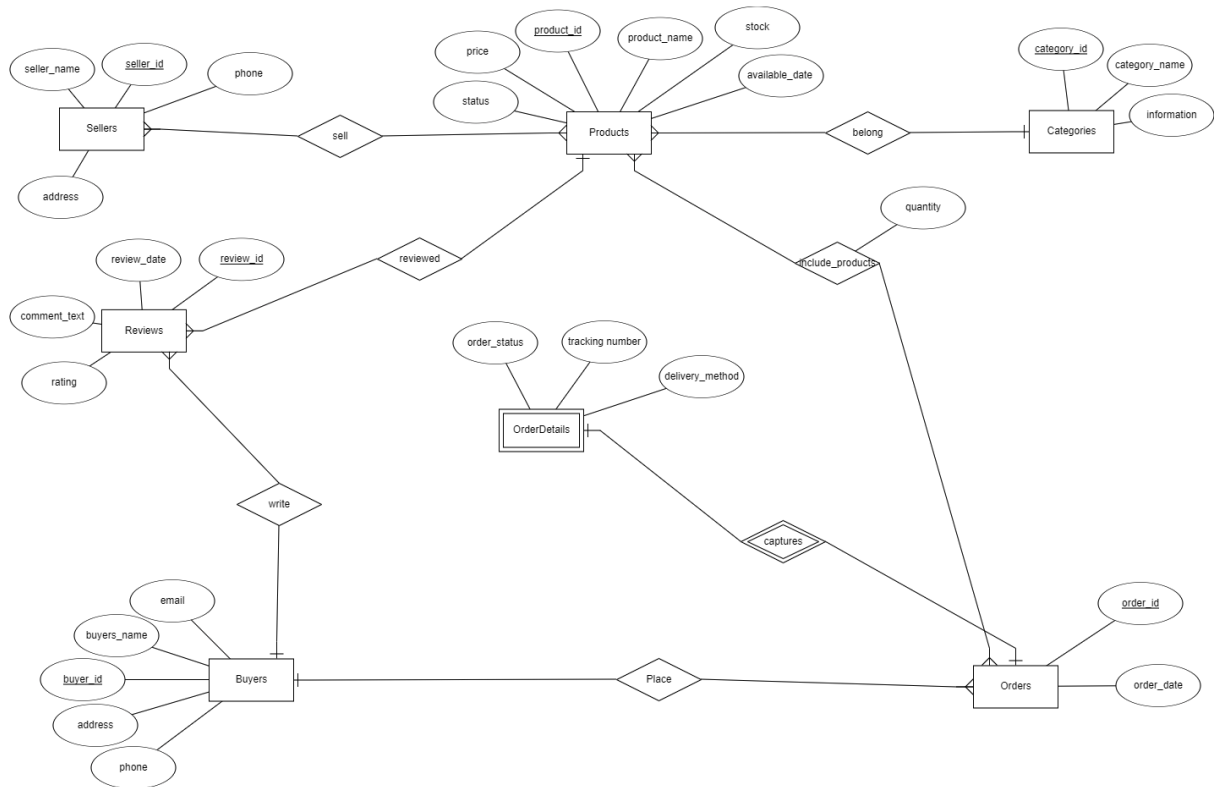
6. Sellers and Products (מוכרים ומוצרים דרך טבלת sell):

- מערכת יחסים של רבים לרבים (many-to-many).
- מוכר יכול למכור מספר מוצרים, ומוצר יכול להימכר על ידי מספר מוכרים
- מיושם על ידי טבלת הצומת-יחס 'מכירה-sell' עם מפתחות זרים הן למוכרים והן למוצרים כמפתח ראשי ליחס

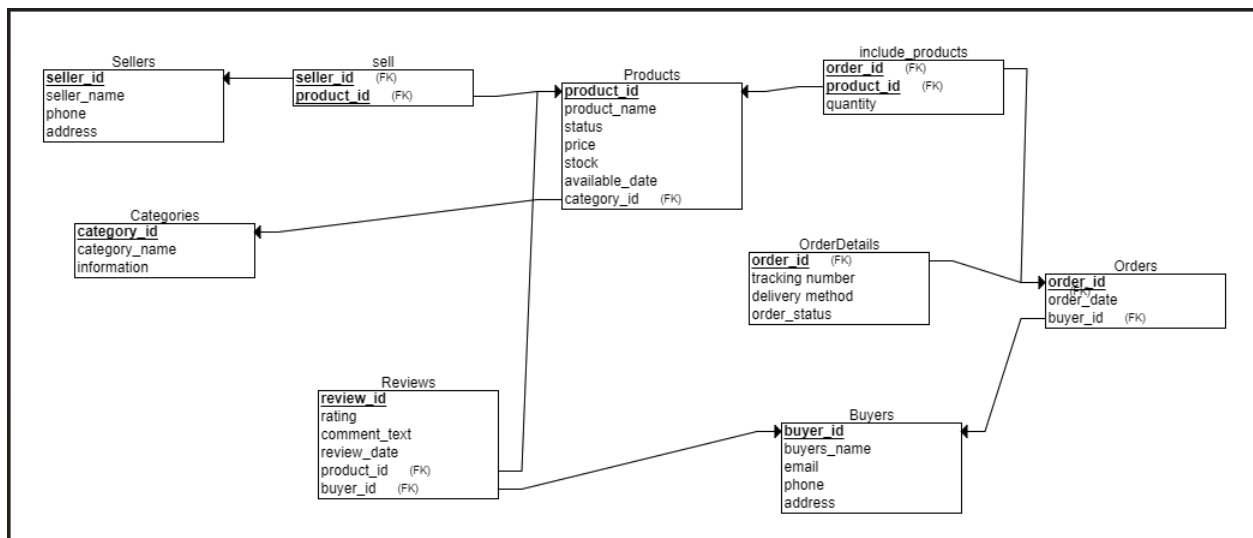
7. Orders and Products (הזמנות ומוצרים דרך טבלת include_products):

- מערכת יחסים של רבים לרבים (many-to-many).
- הזמנה יכולה לכלול מספר מוצרים, ומוצר יכול להיות במספר הזמנות
- מיושם על ידי טבלת הצומת 'include_products' עם מפתחות זרים הן להזמנות והן למוצרים בנוסף לכמות הזמנות מכל מוצר. לכל הזמנה יכולות להיות כמה רשומות בinclude_products

להלן ERD של בסיס הנתונים:



כמו כן הנה DSD של בסיס הנתונים:



יצירת הטבלאות:

```

CREATE TABLE Sellers (
    seller_name VARCHAR2(20) NOT NULL,
    seller_id INT NOT NULL,
    phone VARCHAR2(20) NOT NULL,
    address VARCHAR2(30) NOT NULL,
    PRIMARY KEY (seller_id)
);

CREATE TABLE Categories (
    category_id INT NOT NULL,
    category_name VARCHAR2(35) NOT NULL,
    information VARCHAR2(50),
    PRIMARY KEY (category_id)
);

CREATE TABLE OrderDetails (
    order_id INT NOT NULL,
    delivery_method VARCHAR2(20) NOT NULL,
    tracking_number VARCHAR2(20) NOT NULL,
    PRIMARY KEY (order_id),
    FOREIGN KEY (order_id) REFERENCES Orders(order_id)
);

CREATE TABLE Buyers (
    buyer_id INT NOT NULL,
    buyer_name VARCHAR2(200) NOT NULL,
    email VARCHAR2(255) NOT NULL,
    phone VARCHAR2(20) NOT NULL,
    address VARCHAR2(200) NOT NULL,
    PRIMARY KEY (buyer_id)
);

CREATE TABLE Products (
    product_id INT NOT NULL,
    product_name VARCHAR2(200) NOT NULL,
    status VARCHAR2(20),
    price NUMBER(10,2) NOT NULL, -- Use NUMBER for price with precision (10) and scale
    (2)
    category_id INT NOT NULL,
    stock INT,
    available_date date,
    PRIMARY KEY (product_id),
    FOREIGN KEY (category_id) REFERENCES Categories(category_id)
);

CREATE TABLE Orders (
    order_id INT NOT NULL,
    order_date DATE NOT NULL,
    buyer_id INT NOT NULL,
    PRIMARY KEY (order_id),
    FOREIGN KEY (buyer_id) REFERENCES Buyers(buyer_id)
);

```

```

CREATE TABLE Reviews (
    review_id INT NOT NULL,
    rating NUMBER(10,1), -- Use NUMBER for rating with precision (10) and scale (1)
    comment_text VARCHAR2(2000),
    review_date DATE NOT NULL,
    product_id INT NOT NULL,
    buyer_id INT NOT NULL,
    PRIMARY KEY (review_id),
    FOREIGN KEY (product_id) REFERENCES Products(product_id),
    FOREIGN KEY (buyer_id) REFERENCES Buyers(buyer_id)
);

```

```

CREATE TABLE sell (
    seller_id INT NOT NULL,
    product_id INT NOT NULL,
    PRIMARY KEY (seller_id, product_id),
    FOREIGN KEY (seller_id) REFERENCES Sellers(seller_id),
    FOREIGN KEY (product_id) REFERENCES Products(product_id)
);

```

```

CREATE TABLE include_products (
    order_id INT NOT NULL,
    product_id INT NOT NULL,
    quantity INT not null,
    PRIMARY KEY (order_id, product_id),
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);

```

```
);
```


מחיקת הטבלאות:

```
-- dropTables.sql

-- Drop the tables in the reverse order of their creation to ensure all
dependencies are handled properly

DROP TABLE include_products CASCADE CONSTRAINTS;
DROP TABLE sell CASCADE CONSTRAINTS;
DROP TABLE Reviews CASCADE CONSTRAINTS;
DROP TABLE Orders CASCADE CONSTRAINTS;
DROP TABLE Products CASCADE CONSTRAINTS;
DROP TABLE Buyers CASCADE CONSTRAINTS;
DROP TABLE OrderDetails CASCADE CONSTRAINTS;
DROP TABLE Categories CASCADE CONSTRAINTS;
DROP TABLE Sellers CASCADE CONSTRAINTS;
```

הכנסת נתונים לבסיס הנתונים::הכנסה ע"י Insert

```
-- insertTables.sql

-- Insert data into Sellers table
INSERT INTO Sellers (seller_name, seller_id, phone, address) VALUES ('Seller One', 1, '123-456-7890', '123 Seller St');
INSERT INTO Sellers (seller_name, seller_id, phone, address) VALUES ('Seller Two', 2, '987-654-3210', '456 Seller Ave');

-- Insert data into Categories table
INSERT INTO Categories (category_id, category_name, information) VALUES (1, 'Electronics', 'Electronic items and gadgets');
INSERT INTO Categories (category_id, category_name, information) VALUES (2, 'Books', 'Various books and literature');

-- Insert data into Products table
INSERT INTO Products (product_id, product_name, information, price, category_id) VALUES (1, 'Laptop', 'High performance laptop', 999.99, 1);
INSERT INTO Products (product_id, product_name, information, price, category_id) VALUES (2, 'Smartphone', 'Latest model smartphone', 699.99, 1);
INSERT INTO Products (product_id, product_name, information, price, category_id) VALUES (3, 'Novel', 'Best-selling novel', 19.99, 2);

-- Insert data into OrderDetails table
INSERT INTO OrderDetails (product_id, quantity, delivery_method) VALUES (1, 10, 'Standard');
INSERT INTO OrderDetails (product_id, quantity, delivery_method) VALUES (2, 15, 'Express');
INSERT INTO OrderDetails (product_id, quantity, delivery_method) VALUES (3, 5, 'Standard');

-- Insert data into Buyers table
INSERT INTO Buyers (buyer_id, buyer_name, email, phone, address) VALUES (1, 'Buyer One', 'buyer1@example.com', '111-222-3333', '123 Buyer Rd');
INSERT INTO Buyers (buyer_id, buyer_name, email, phone, address) VALUES (2, 'Buyer Two', 'buyer2@example.com', '444-555-6666', '456 Buyer Ln');

-- Insert data into Orders table
INSERT INTO Orders (order_id, order_date, tracking_number, product_id, buyer_id) VALUES (1, TO_DATE('2024-01-01', 'YYYY-MM-DD'), 'TRACK123', 1, 1);
INSERT INTO Orders (order_id, order_date, tracking_number, product_id, buyer_id) VALUES (2, TO_DATE('2024-01-02', 'YYYY-MM-DD'), 'TRACK456', 2, 2);
INSERT INTO Orders (order_id, order_date, tracking_number, product_id, buyer_id) VALUES (3, TO_DATE('2024-01-03', 'YYYY-MM-DD'), 'TRACK789', 3, 1);

-- Insert data into Reviews table
INSERT INTO Reviews (review_id, rating, comment_text, review_date, product_id, buyer_id) VALUES (1, 4.5, 'Great product!', TO_DATE('2024-01-10', 'YYYY-MM-DD'), 1, 1);
INSERT INTO Reviews (review_id, rating, comment_text, review_date, product_id, buyer_id) VALUES (2, 3.5, 'Good value for money', TO_DATE('2024-01-11', 'YYYY-MM-DD'), 2, 2);
INSERT INTO Reviews (review_id, rating, comment_text, review_date, product_id, buyer_id) VALUES (3, 5.0, 'Excellent read!', TO_DATE('2024-01-12', 'YYYY-MM-DD'), 3, 1);

-- Insert data into sell table
INSERT INTO sell (seller_id, product_id) VALUES (1, 1);
INSERT INTO sell (seller_id, product_id) VALUES (1, 3);
INSERT INTO sell (seller_id, product_id) VALUES (2, 2);

-- Insert data into include table
INSERT INTO include_products (order_id, product_id) VALUES (1, 1);
INSERT INTO include_products (order_id, product_id) VALUES (2, 2);
INSERT INTO include_products (order_id, product_id) VALUES (3, 3);

-- End of insertTables.sql

select * from sellers
|
```

הכנסה ע"י data generate :

PRODUCTS				
< Owner		Table	Number of records	
> C##RAZ		PRODUCTS	600.800	
...				
Name	Type	Size	Data	Master
PRODUCT_ID	NUMBER		Random(111,9)	...
PRODUCT_NAME	VARCHAR2	200	List('Wireless Mouse', 'Bluetooth Speaker', 'Laptop Stand', 'USB-C Hub', 'Gaming Keyboard', 'Noise Cancelling Headphones', 'Portable Charger', 'Smart Watch', 'Digital Camera', 'LED Desk Lamp', 'External Hard Drive', 'Smartphone Case',	...
INFORMATION	VARCHAR2	50	Components.Description	...
PRICE	NUMBER	10,2	Components.Price	...
CATEGORY_ID	NUMBER		Random(1,10)	...
...				

```

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (2484781687, 'Bluetooth Speaker', 'GeForceFX 5200 128MB DDR (Light Version) DVI + TV-', 79.00, 2);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (1498622636, 'Smart Watch', 'Combo Intel P4 3.06ghz CPU + Fan(BOx)with Hyper Th', 519.00, 6);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (213653289, 'Smartphone Gimbal', 'Gigabyte GA-7VAXP AMD Motherboard. VIA KT400 RAID ', 119.00, 5);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (3024872317, 'Saw', '313B Black 300W P4/AMD Fan&USB .', 23.00, 9);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (1698245755, 'Lawn Mower', '512Mb DDR PC2100 Kingston', 85.00, 8);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (3835088224, 'Blood Pressure Monitor', 'Western Digital 20GB 5400RPM ATA100', 57.50, 6);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (2217730790, 'Wallet', 'Intel D845PEBT2 M/B onboard Raid and S-ATA .', 129.00, 8);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (396609712, 'Outdoor Grill', 'Boka 406 Sub Woofer 4 Channel Speaker system', 35.00, 4);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (499260042, 'Car Vacuum', 'Cooler Master Copper HEAT PIPE Cooler. HHC-001', 34.00, 7);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (2049356926, 'External Hard Drive', '312B ATX Case 250W P4/AMD Ready Power Supply.', 14.00, 1);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (1528148711, 'Outdoor Grill', 'Asus P4C800 Deluxe intel 875 Chip 800FSB DDR SATA ', 618.00, 1);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (664236441, 'Pulse Oximeter', 'Cyber Power 1250VA Back Up.', 135.00, 10);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (2446625864, 'Fire Pit', 'GA-7VA VIA KT400 chipset 8X AGP DDR400 onboard Sou', 69.00, 3);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (699522961, 'Fitness Tracker', 'AMD XP 1800+ & Gigabyte GA-7DXE MB w/ 313M Case W/', 135.00, 4);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (195856121, 'Travel Pillow', 'Seagate 40GB 5400RPM ATA100', 56.00, 3);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (2281037408, 'Car Charger', '5ft Cat5 network cable w/RJ45 connectors', 5.00, 9);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (3721363991, 'Workbench', 'AMD XP 2200+ & Gigabyte 7DXE w/ 313M case . W/ Fre', 159.00, 2);

insert into C##RAZ.PRODUCTS (PRODUCT_ID, PRODUCT_NAME, INFORMATION, PRICE, CATEGORY_ID)
values (1730629222, 'Wristwatch', 'Boka 1050 Sub Woofer and Speakers', 9.00, 5);

```

Definition Options Result

▼ C##RAZ@Xe -[11:27:17] 662 records generated in 0.234 seconds

category

SQL Output Statistics

```

INSERT INTO categories (category_id, category_name, information) VALUES (10, 'Lighting', 'Lights for home and office use.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (11, 'Data Storage', 'Store and manage data: hard drives.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (12, 'Health & Fitness', 'Fitness trackers, smart scales.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (13, 'Smart Home Devices', 'Thermostats, security cameras.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (14, 'Small Kitchen Appliances', 'Blenders, coffee makers.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (15, 'Home Cleaning', 'Robot vacuums, pressure washers.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (16, 'Personal Care', 'Toothbrushes, shavers for grooming.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (17, 'Medical Devices', 'Blood pressure devices for home.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (18, 'Exercise Equipment', 'Yoga mats, treadmills for fitness.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (19, 'Television & Home Theater', 'Smart TVs, projectors for entertainment.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (20, 'Streaming & Media', 'Media streaming with sticks.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (21, 'Outdoor & Garden', 'Garden tools, patio furniture.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (22, 'Tools & Hardware', 'Drills, saws, toolkits for home.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (23, 'Automotive Accessories', 'Car chargers, GPS, dash cams.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (24, 'Travel Accessories', 'Travel pillows, suitcases, bags.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (25, 'Fashion Accessories', 'Sunglasses, wallets for fashion.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (26, 'Luggage & Bags', 'Camera bags, handbags, luggage.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (27, 'Camera Accessories', 'Tripods, memory cards for photography.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (28, 'Virtual Reality', 'VR headsets for virtual experiences.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (29, 'Drones', 'Drones for photography, entertainment.');
```

```

INSERT INTO categories (category_id, category_name, information) VALUES (30, '3D Printing', '3D printers for creating objects.');
```

```

COMMIT;
```

הכנסה ע"י קובץ טקסט:

products

Data from Tefitile | Data to Oracle

File

55, Leaf Blower, Refurbished, 99.99, 10
60, Pressure Washer, New, 149.99, 10
61, Tool Kit, New, 29.99, 10
62, Cordless Drill, Refurbished, 79.99, 10
63, Saw, New, 49.99, 10
64, Workbench, New, 99.99, 10
65, Safety Goggles, Refurbished, 9.99, 10
66, Work Gloves, New, 14.99, 10
67, Paint Sprayer, New, 69.99, 10
68, Air Compressor, New, 149.99, 10
69, Wrench Set, Refurbished, 19.99, 10
70, Screwdriver Set, New, 14.99, 10
71, Socket Set, New, 39.99, 10
72, Measuring Tape, Refurbished, 9.99, 10
73, Car Vacuum, New, 29.99, 10
74, Car Charger, Refurbished, 19.99, 10
75, GPS Navigator, New, 99.99, 10
76, Dash Cam, New, 59.99, 10
77, Car Seat Cover, New, 49.99, 10
78, Steering Wheel Cover, Refurbished, 14.99, 10
79, Car Air Purifier, New, 39.99, 10
80, Tire Pressure Gauge, Refurbished, 9.99, 10
81, Jump Starter, New, 69.99, 10
82, Toolbox, New, 49.99, 10
83, Sunglasses, Refurbished, 19.99, 10
84, Watch, New, 39.99, 10
85, Wallet, New, 29.99, 10
86, Backpack, Refurbished, 49.99, 10
87, Handbag, New, 24.99, 10

Configuration

General: Field names: Quiet character: Field1 (+,-,*) Field2 (+,-,*) Field3 (+,-,*) End of line end: Comment line: Import lines: 1

Field Start: Relative position Absolute position Character Field End: Length Character

Filter:

Result Preview

#	1	2	3	4	5
97	Action Camera	New	199.99	19	10
98	3D Printer	New	399.99	19	10
99	Electric Scooter	Refurbished	299.99	19	10
100	Wireless Mouse	New	29.99	25	10

Ignore Import to Script Copy CMMR@2M 12/24/2023 484 records imported in 44.68 seconds (7 failed)

Data from Tefitile | Data to Oracle

General

Owner: Table: PRODUCTS
Commit every: 100
☒ Overwrite duplicates ☐ Delete records
☐ Ignore duplicates ☐ Truncate table

Fields

Field1 -> PRODUCT_ID (NUMBER)
Field2 -> PRODUCT_NAME (VARCHAR2)
Field3 -> STATUS (VARCHAR2)
Field4 -> CATEGORY_ID (NUMBER)
Field5 -> STOCK_QUANTITY

Field STOCK (NUMBER)

FieldType: Number
Create SQL

SQL function:
additional Oracle processing for example substr(?, 1, 20)

הכנסה ע"י Mockaroo:

buyers

SCHEMAS DATASETS MOCK APIS SCENARIOS PROJECTS FUNCTIONS
?
SIGN IN
UPGRADE NOW

Field Name	Type	Options
buyer_id	Sequence	start at: 9999 step: 1 repeat: 1 restart at: blank: 0%
buyer_name	Full Name	blank: 0%
email	Email Address	blank: 0%
phone	Phone	format: ###-###-#### blank: 0%
address	Street Address	blank: 0%

+ ADD ANOTHER FIELD
GENERATE FIELDS USING AI...

Rows: 700
Format: SQL
Table Name: Buyers
☐ include CREATE TABLE

[Follow @mockaroo](#)

```

dropTables.sql selectAll.sql createTables.sql category generation insertTables.sql Buyers_mockaroo.sql
SQL Output Statistics
INSERT INTO Buyers (buyer_id, buyer_name, email, phone, address) VALUES (9999, 'Omero Hackett', 'ohackett0@globo.com', '768-751-2311', '70 Eastlawn Pass');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10000, 'Ronald Kennerley', 'rkennerley1@latimes.com', '348-486-0041', '238 Del Sol Court');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10001, 'Janek Hundell', 'jhundell12@google.com.br', '118-350-3220', '21697 Straubel Pass');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10002, 'Danell Jeness', 'djeness3@barnesandnoble.com', '635-177-1340', '2 Eagan Trail');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10003, 'Monika Scallon', 'mscallon4@exblog.jp', '280-711-3391', '4 Paget Avenue');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10004, 'Mathilda Nyssens', 'mnyssens5@511a', '106-717-1994', '22593 Melby Alley');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10005, 'Paul Quincee', 'pquincee6@eventbrite.com', '369-141-6106', '14 Northport Parkway');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10006, 'Carter Casconi', 'ccasconi7@privacy.gov.au', '263-638-7327', '3600 Wayridge Lane');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10007, 'Xever Spadelli', 'xspadelli8@simplemachines.org', '304-428-1475', '37 Surrey Plaza');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10008, 'Traver Cosser', 'tcosser9@washington.edu', '290-555-8727', '54355 Eagle Crest Aven');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10009, 'Aurthur Ennew', 'aennewa@over-blog.com', '242-769-4745', '380 Buhler Lane');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10010, 'Kissee Suttaby', 'ksuttabyb@ca.gov', '552-424-8256', '3873 Mayfield Point');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10011, 'Friedrick Mack', 'fmackc@google.pl', '636-610-5533', '578 Sutteridge Road');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10012, 'Louissette Po', 'lpod@stumbleupon.com', '921-120-1240', '1 Mallory Point');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10013, 'Godfree Gavey', 'ggaveye@cargocollective.com', '836-623-2801', '0807 Gale Parkway');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10014, 'Bjorn Loran', 'bloramf@spotify.com', '829-617-2737', '444 Emmet Lane');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10015, 'Reg Gosart', 'rgosartg@narod.ru', '575-550-5426', '12117 Coolidge Park');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10016, 'Clim Cassely', 'ccasselyh@hugobdomains.com', '626-217-9523', '564 Bluejay Terrace');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10017, 'Pebrook Cultcheth', 'pcultchethi@google.co.jp', '691-368-1749', '44 Harbort Park');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10018, 'Pepito Elener', 'pelenerj@typepad.com', '607-101-9410', '32129 Gina Plaza');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10019, 'Corly Peaseman', 'cpeasemank@bloglines.com', '603-862-8127', '2084 Banning Street');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10020, 'Tedman Hagwood', 'thagwoodl@opensource.org', '747-951-7941', '0 Arkansas Court');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10021, 'Jeramie Trundler', 'jtrundlerm@imgur.com', '105-417-4031', '74 Utah Way');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10022, 'Wolfgang Burnell', 'wburnelln@columbia.edu', '360-678-6279', '14157 Schiller Circl');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10023, 'Jamesy Hought', 'jhoughto@google.com.au', '603-785-4970', '72067 Ridge Oak Point');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10024, 'Avivah Stoggles', 'astogglesp@opensource.org', '766-402-6097', '27 Oriole Street');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10025, 'Wally Elrick', 'welricky@blogtalkradio.com', '860-389-8723', '5 Commercial Crossin');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10026, 'Lara Sallinger', 'lsallinger@xing.com', '504-577-4248', '0 Derek Point');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10027, 'Bear Farrier', 'bfarriers@icio.us', '762-971-6897', '9131 Sutteridge Circle');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10028, 'Letisha Cuppitt', 'lcuppitt@unicef.org', '867-659-1341', '6 Randy Point');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10029, 'Byrom Casbourne', 'bcasbourneu@eepurl.com', '287-942-1181', '3945 David Center');
insert into Buyers (buyer_id, buyer_name, email, phone, address) values (10030, 'Cesya Slaymaker', 'cslaymakerv@instagram.com', '691-896-6898', '09521 Eastlawn Ave');

```

OrderDetails

ORDERDETAILS				
Owner		Table	Number of records	
C##RAZ		ORDERDETAILS	10..20	
Name	Type	Size	Data	Master
ORDER_ID	NUMBER		Sequence(9999, [10])	...
DELIVERY_METHOD VARCHAR2	20		List("home delivery", "pick up spots", "pick up from store")	...
TRACKING_NUM VARCHAR2	20		Components.Code	...
				...

include_products

General

Owner

Table

Number of records

C##RAZ

INCLUDE_PRODUCTS

10..20

Commit every...

100

Overwrite duplicates

Ignore duplicates

Delete records

Truncate table

Initializing Script

Finalizing Script

Fields

Field1 -> ORDER_ID (NUMBER)

Field2 -> PRODUCT_ID (NUMBER)

Field: PRODUCT_ID (NUMBER)

Fieldtype: Number

Create SQL

SQL function

additional Oracle processing, for example: substr(%, 1, 20)

Result Preview

1	2
10344	305
10687	415
10729	106
11309	114
11768	111
11815	181
12217	405
12313	292
12509	23
12649	123
13024	221
13886	66
13990	296
14278	152

Import

Import to Script

Close

C##RAZ@Xe

[0:58.33] 400 records imported in 0.265 seconds

Help

Orders

ORDERS

< Owner

C##RAZ

Table

ORDERS

Number of records

10..20

Name	Type	Size	Data	Master
ORDER_ID	NUMBER	*	Sequence(9999, [15])	...
ORDER_DATE	DATE	*	Random(1-1-2023, 1-1-2025)	...
BUYER_ID	NUMBER	*	List(select buyer_id from buyers)	...

יצירת BACKUP לשלב 1:

Name	Type	Compiled
BUYERS	TABLE	07/07/2024 21:22:56
CATEGORIES	TABLE	07/07/2024 22:07:44
INCLUDE_PRODUCTS	TABLE	07/07/2024 21:22:56
ORDERDETAILS	TABLE	07/07/2024 21:22:56
ORDERS	TABLE	07/07/2024 21:22:56
PRODUCTS	TABLE	07/07/2024 22:07:02
REVIEWS	TABLE	07/07/2024 21:22:56
SELL	TABLE	07/07/2024 21:22:56
SELLERS	TABLE	07/07/2024 21:22:55

User: <CURRENT USER>

Oracle Export SQL Inserts PL/SQL Developer Log

9 Table(s) exported
4271 Record(s) exported
Export finished on 17/07/2024 09:41:49

C#RAZ@Xe Exporting... Done

חלק 2

שאלות - Queries

שאלות select :שאלתה ראשונה:

צוות השיווק של החנות מתכנן קמפיין חדש להדגשת קטגוריות המוצרים בעלות הביצועים הטובים ביותר. באתר. הם רוצים לבסס את ההחלטות שלהם על משוב ומעורבות של לקוחות. מנהלת השיווק מעוניינת בדו"ח שמראה את הביצועים של כל קטגוריה על סמך ביקורות של לקוחות. נרצה לראות את הדירוג הממוצע לכל קטגוריה וכמה ביקורות הוא קיבל. המנהלת טוענת שזה יעזור לחנות להחליט אילו קטגוריות להציג במסע הפרסום הקרוב שלנו 'מועדפים של לקוחות'.

השאלתה הבאה מחזירה את ממוצע של דירוגים של מוצרים תחת קטגוריה מסויימת וכן את מספר הביקורת שנעשו עבור על קטגוריה.

```
SELECT
  C.category_name,
  AVG(R.rating) AS average_rating,
  COUNT(R.review_id) AS review_count
FROM
  Categories C
JOIN
  Products P ON C.category_id = P.category_id
JOIN
  Reviews R ON P.product_id = R.product_id
GROUP BY
  C.category_name
ORDER BY
  average_rating DESC;
```

	CATEGORY_NAME	AVERAGE_RATING	REVIEW_COUNT
3	Networking	6.02857142857143	7
4	Outdoor & Garden	5.90491803278689	61
5	Gaming Gear	5.85	20
6	Headphones & Earbuds	5.8375	8
7	Television & Home Theater	5.68743455497382	191
8	Virtual Reality	5.6	8
9	Exercise Equipment	5.516	25
10	Cameras & Photography	5.4125	8
11	Streaming & Media	5.36666666666667	27
12	Drones	5.1	1
13	Health & Fitness	4.98333333333333	6
14	Audio Equipment	4.86666666666667	9
15	Lighting	4.5	9
16	Computer Accessories	4.23	10
17	Wearable Technology	3.2	1

שאלתה שנייה:

החנות רוצה לנתח את ביצועי המוצרים בחנות. לצורך כך החנות רוצה לזהות מוצרים פופולריים ומוצרים פחות פופולריים בהתאם לתוצאות לקבל החלטה על צורך בקידום מוצרים או הסרתם מהקטלוג וגם לגבי ניהול מלאי ואסטרטגיות שיווק.

השאלתה הבאה מחזירה רשימה של מוצרים עם ספירת ההזמנות שלהם, כולל מוצרים שלא הוזמנו. ממזין לפי מוצרים הנמכרים ביותר.

```
SELECT
    P.product_id,
    P.product_name,
    COUNT(IP.product_id) AS order_count
FROM
    Products P
LEFT JOIN
    include_products IP ON P.product_id = IP.product_id
GROUP BY
    P.product_id, P.product_name
ORDER BY
    order_count DESC;
```

	PRODUCT_ID	PRODUCT_NAME	ORDER_COUNT
1	184	Wristwatch	4
2	221	Rowing Machine	4
3	421	Electric Pressure Washer	4
4	399	Massage Roller	4
5	409	Video Doorbell Pro	4
6	369	Virtual Reality Headset	4
7	7	USB Flash Drive	3
8	477	3D Scanner	3
9	58	Lawn Mower	3
10	84	Wristwatch	3
11	91	Tripod	3
12	111	Smartphone Case	3
13	161	Tool Kit	3
14	172	Measuring Tape	3
15	177	Car Seat Cover	3
16	407	Headphones	3

שאלתה שלישית:

החנות רוצה שתהיה לה בקרה על ביצועי המוכרים ועל כמות המוצרים הכוללת שמכרו, כדי לזהות מוכרים מובילים, ומוכרים פחות טובים. התוכנית של חנות היא לתת הטבות למוכרים מובילים כדי לדברן אותם להמשיך למכור אצלנו בחנות וכן לתת עזרה למוכרים פחות טובים כדי שימכרו יותר מוצרים.

השאלתה הבאה מחשבת את המספר הכולל של מוצרים שנמכרו ואת סך ההכנסות עבור כל מוכר, ממוינת לפי הכנסה בסדר יורד.

```
SELECT
    S.seller_id,
    S.seller_name,
    COUNT(SL.product_id) AS total_products_sold,
    SUM(P.price * IP.quantity) AS total_revenue
FROM
    Sellers S
JOIN
    sell SL ON S.seller_id = SL.seller_id
JOIN
    Products P ON SL.product_id = P.product_id
JOIN
    include_products IP ON P.product_id = IP.product_id
GROUP BY
    S.seller_id, S.seller_name
ORDER BY
    total_revenue DESC;--

--most popular products by counting the number of orders each product
-- has received and then ordering the results by the number of orders in desce
```

	SELLER_ID	SELLER_NAME	TOTAL_PRODUCTS_SOLD	TOTAL_REVENUE
1	14738	Joanna Henstridge	5	45049.39
2	11705	Martin Fariq	3	44799.48
3	15989	Marc Taylor	4	44799.44
4	13811	Rita Schwarzenegger	4	44799.44
5	13352	Nora von Sydow	4	44799.44
6	15512	Kirsten Neuwirth	2	44199.66
7	12470	Shawn Dukakis	4	35079.52
8	16763	Vin Chambers	2	33999.66
9	16169	Jane Hynde	2	33999.66
10	13325	Jonathan LaMond	2	32999.78
11	16925	Arnold Bragg	5	31649.62
12	11885	Mandy Paltrow	6	30264.13
13	15350	Giancarlo Paltrow	4	28759.57
14	16646	Jean-Claude Lemmon	3	28699.59
15	14774	Ashton Russo	6	24349.27
16	11381	Anna Sample	3	22249.65
17	14117	Maria Keener	2	22099.83
18	11615	Bradley Shorter	7	21349.16
19	16547	Emma Diddley	3	19999.76
20	11786	Kurt Thewlis	4	19239.51

שאלת רביעית :

החנות רוצה להשיק תוכנית נאמנות ללקוחותיה הטובים ביותר. עם זאת, עם אלפי קונים, היא זקוקה לנתונים כדי לקבל החלטות מושכלות. החנות משתמשת בשאלתה הזו כדי לזהות את לקוחותיה הטובים ביותר.

השאלתה הבאה מחזירה את שלושת הקונים המובילים בהתבסס על הסכום הכולל שהוציא, ומציגה את ה id, השם, מספר הכולל של ההזמנות והסכום הכולל שהוציא.

```
--SELECT QUERIES
```

```
--Query to find the top 3 buyers who have placed the highest number
-- of orders along with the total amount spent by each buyer.
```

```
SELECT
    B.buyer_id,
    B.buyer_name,
    COUNT(O.order_id) AS total_orders,
    SUM(P.price * IP.quantity) AS total_spent
FROM
    Buyers B
JOIN
    Orders O ON B.buyer_id = O.buyer_id
JOIN
    include_products IP ON O.order_id = IP.order_id
JOIN
    Products P ON IP.product_id = P.product_id
GROUP BY
    B.buyer_id, B.buyer_name
ORDER BY
    total_spent DESC
FETCH FIRST 3 ROWS ONLY;
```

	BUYER_ID	BUYER_NAME	TOTAL_ORDERS	TOTAL_SPENT
1	10510	Nadia Giordano	2	26999.76
2	10019	Corly Peaseman	2	24679.67
3	10457	Shanta O'Sharkey	1	22099.83

שאלות DELETE:**שאלת ראשונה:**

החנות שמה לב שבסיס הנתונים שלה גדל באופן משמעותי, האט את ביצועי המערכת וניפח את עלויות השיווק. החנות מעוניינת לנקות את בסיסי הנתונים ומחליטה להסיר חשבונות של משתמשים שנרשמו אך מעולם לא ביצעו רכישה או השאירו ביקורת, מתוך מחשבה שכנראה חשבונות אלו נטושים או חשבונות ספאם.

השאלתה הבאה מוחקת קונים לא פעילים שלא ביצעו הזמנות ולא השאירו ביקורות.

```
--deletes buyers who have neither placed any orders nor written any reviews.
DELETE FROM Buyers
WHERE buyer_id IN (
    SELECT B.buyer_id
    FROM Buyers B
    WHERE B.buyer_id NOT IN (
        SELECT O.buyer_id
        FROM Orders O
    )
    AND B.buyer_id NOT IN (
        SELECT R.buyer_id
        FROM Reviews R
    )
);
```

113:26 C##RAZ@Xe [11:08:12] 217 rows deleted in 0.116 seconds

בס"ד

לפני:

	BUYER_ID	BUYER_NAME	EMAIL	PHONE	ADDRESS
1	9999	Omero Hackett	ohackett0@globo.com	768-751-2311	70 Eastlawn Pass
2	10000	Ronald Kennerley	rkennerley1@latimes.com	348-486-0041	238 Del Sol Court
3	10001	JaneK Hundell	jhundell2@google.com.br	118-350-3220	21697 Straubel Pass
4	10002	Danell Jeness	djeness3@barnesandnoble.com	635-177-1340	2 Eagan Trail
5	10003	Monika Scallon	mscalton4@exblog.jp	280-711-3391	4 Paget Avenue
6	10004	Mathilda Nyssens	mnyssens5@51.la	106-717-1994	22593 Melby Alley
7	10005	Paul Quincee	pquincee6@eventbrite.com	369-141-6106	14 Northport Parkway
8	10006	Carter Cesconi	ccesconi7@privacy.gov.au	263-638-7327	3600 Wayridge Lane
9	10007	Xever Spadelli	xspadelli8@simplemachines.org	304-428-1475	37 Surrey Plaza
10	10008	Traver Cosser	tcosser9@washington.edu	290-555-8727	54355 Eagle Crest Avenue
11	10009	Aurthur Ennew	aennewa@over-blog.com	242-769-4745	380 Buhler Lane
12	10010	Kissee Suttaby	ksuttabyb@ca.gov	552-424-8256	3873 Mayfield Point
13	10011	Friedrick Mack	fmackc@google.pl	636-610-5533	578 Sutteridge Road
14	10012	Louissette Po	lpod@stumbleupon.com	921-120-1240	1 Mallory Point
15	10013	Godfree Gavey	ggaveye@cargocollective.com	836-623-2801	0807 Gale Parkway
16	10014	Bjorn Loram	bloramf@spotify.com	829-617-2737	444 Emmet Lane
17	10015	Reg Gosart	rgosartg@narod.ru	575-550-5426	12117 Coolidge Park

אחרי:

Buyer_id 10008 נמחק

	BUYER_ID	BUYER_NAME	EMAIL	PHONE	ADDRESS
1	9999	Omero Hackett	ohackett0@globo.com	768-751-2311	70 Eastlawn Pass
2	10000	Ronald Kennerley	rkennerley1@latimes.com	348-486-0041	238 Del Sol Court
3	10001	JaneK Hundell	jhundell2@google.com.br	118-350-3220	21697 Straubel Pass
4	10002	Danell Jeness	djeness3@barnesandnoble.com	635-177-1340	2 Eagan Trail
5	10003	Monika Scallon	mscalton4@exblog.jp	280-711-3391	4 Paget Avenue
6	10004	Mathilda Nyssens	mnyssens5@51.la	106-717-1994	22593 Melby Alley
7	10005	Paul Quincee	pquincee6@eventbrite.com	369-141-6106	14 Northport Parkway
8	10006	Carter Cesconi	ccesconi7@privacy.gov.au	263-638-7327	3600 Wayridge Lane
9	10007	Xever Spadelli	xspadelli8@simplemachines.org	304-428-1475	37 Surrey Plaza
10	10009	Aurthur Ennew	aennewa@over-blog.com	242-769-4745	380 Buhler Lane
11	10010	Kissee Suttaby	ksuttabyb@ca.gov	552-424-8256	3873 Mayfield Point
12	10011	Friedrick Mack	fmackc@google.pl	636-610-5533	578 Sutteridge Road
13	10012	Louissette Po	lpod@stumbleupon.com	921-120-1240	1 Mallory Point
14	10013	Godfree Gavey	ggaveye@cargocollective.com	836-623-2801	0807 Gale Parkway
15	10014	Bjorn Loram	bloramf@spotify.com	829-617-2737	444 Emmet Lane
16	10015	Reg Gosart	rgosartg@narod.ru	575-550-5426	12117 Coolidge Park
17	10016	Clim Cassely	ccasselyh@hugedomains.com	626-217-9523	564 Bluejay Terrace

שאלתה שנייה:

במהלך שנות פעילות החנות, האתר צבר רשימה ארוכה של קטגוריות מוצרים, שרבים מהן כבר לא היו רלוונטיות. רשימת הקטגוריות העמוסה מבלבלת משתמשים חדשים ומסבכת את חיפושי המוצרים. החנות החליטה לייעל את הקטגוריות אך רצתה להבטיח שהיא לא תסיר את הקטגוריות שעדיין בשימוש.

שאלתה זו מסירה קטגוריות שאין בהן מוצרים הנמכרים כעת.

```
--delete categories that have not had any products sold
DELETE FROM Categories
WHERE category_id NOT IN (
    SELECT DISTINCT p.category_id
    FROM Products p
    JOIN sell s ON p.product_id = s.product_id
);
```

לפני:

	CATEGORY_ID	CATEGORY_NAME	INFORMATION
▶ 1	1	Computer Accessories	Enhance computer functionality with peripherals.
2	2	Audio Equipment	Speakers and headphones for great sound.
3	3	Office Supplies	Stands, lamps, and organizers for office.
4	4	Networking	Routers and hubs for network management.
5	5	Gaming Gear	Keyboards, consoles, VR headsets.
6	6	Headphones & Earbuds	Personal audio: music, podcasts.
7	7	Mobile Accessories	Cases, chargers, gimbals for mobiles.
8	8	Wearable Technology	Smart watches, fitness trackers.
9	9	Cameras & Photography	Capture photos, videos with cameras.
10	10	Lighting	Lights for home and office use.
11	11	Data Storage	Store and manage data: hard drives.
12	12	Health & Fitness	Fitness trackers, smart scales.
13	13	Smart Home Devices	Thermostats, security cameras.
14	14	Small Kitchen Appliances	Blenders, coffee makers.

	CATEGORY_ID	CATEGORY_NAME	INFORMATION
14	14	Small Kitchen Appliances	Blenders, coffee makers.
15	15	Home Cleaning	Robot vacuums, pressure washers.
16	16	Personal Care	Toothbrushes, shavers for grooming.
17	17	Medical Devices	Blood pressure devices for home.
18	18	Exercise Equipment	Yoga mats, treadmills for fitness.
19	19	Television & Home Theater	Smart TVs, projectors for entertainment.
20	20	Streaming & Media	Media streaming with sticks.
21	21	Outdoor & Garden	Garden tools, patio furniture.
22	22	Tools & Hardware	Drills, saws, toolkits for home.
23	23	Automotive Accessories	Car chargers, GPS, dash cams.
24	24	Travel Accessories	Travel pillows, suitcases, bags.
25	25	Fashion Accessories	Sunglasses, wallets for fashion.
26	26	Luggage & Bags	Camera bags, handbags, luggage.
27	27	Camera Accessories	Tripods, memory cards for photography.
28	28	Virtual Reality	VR headsets for virtual experiences.
29	29	Drones	Drones for photography, entertainment.
30	30	3D Printing	3D printers for creating objects.

אחרי: אפשר לראות שקטגוריה 3 נמחקה בעקבות אי מכירה של מוצרים

	CATEGORY_ID	CATEGORY_NAME	INFORMATION
1	1	Computer Accessories	Enhance computer functionality with peripherals.
2	2	Audio Equipment	Speakers and headphones for great sound.
3	4	Networking	Routers and hubs for network management.
4	5	Gaming Gear	Keyboards, consoles, VR headsets.
5	6	Headphones & Earbuds	Personal audio: music, podcasts.
6	8	Wearable Technology	Smart watches, fitness trackers.
7	9	Cameras & Photography	Capture photos, videos with cameras.
8	10	Lighting	Lights for home and office use.
9	11	Data Storage	Store and manage data: hard drives.
10	12	Health & Fitness	Fitness trackers, smart scales.
11	17	Medical Devices	Blood pressure devices for home.
12	18	Exercise Equipment	Yoga mats, treadmills for fitness.
13	19	Television & Home Theater	Smart TVs, projectors for entertainment.
14	20	Streaming & Media	Media streaming with sticks.
15	21	Outdoor & Garden	Garden tools, patio furniture.
16	28	Virtual Reality	VR headsets for virtual experiences.
17	29	Drones	Drones for photography, entertainment.

שאלות עידכון:

שאלת ראשונה:

החנות מגלה שעקב מחסור עולמי בשבבים, עלות אביזרי המחשב תגדל משמעותית. כדי לשמור על רווחיות מבלי לזעזע את הלקוחות בעליית מחירים גדולה פתאומית, החנות החליטה ליישם עלייה הדרגתית של 10% על כל אביזרי המחשב. החנות רוצה לעדכן ביעילות מחירים בכל מלאי אביזרי המחשב שלה, החל מעכברים ומקלדות ועד למצלמות אינטרנט ורכזות USB. החנות רוצה להבטיח שרק מוצרים בקטגוריה ספציפית זו יושפעו מהעליה, ולהותיר את המחירים של שאר המוצרים האחרים ללא שינוי.

שאלתה זו מעלה את המחיר של כל המוצרים בקטגוריית 'אביזרים למחשב' ב-10%.

--updates the price of all products in a specific category by a certain percentage.

```
UPDATE
Products
SET
price = price * 1.10
WHERE
category_id =
(SELECT category_id
FROM Categories
WHERE category_name = 'Computer Accessories');
```

לפני:

	PRODUCT_ID	PRODUCT_NAME	STATUS	PRICE	CATEGORY_ID	STOCK	AVAILABLE_DATE
1	1	Laptop	New	899.99	1	1	26/04/2023
2	2	Desktop Computer	Refurbished	699.99	1	22	20/10/2023
3	3	Monitor	New	199.99	1	0	03/05/2023
4	4	Keyboard	New	49.99	1	11	30/06/2023
5	5	Mouse	New	19.99	1	16	30/06/2023
6	6	External Hard Drive	Refurbished	79.99	11	18	23/12/2023
7	7	USB Flash Drive	New	14.99	11	20	24/10/2023
8	8	Printer	New	149.99	2	15	05/06/2023
9	9	Wireless Keyboard	Refurbished	29.99	1	25	12/03/2023
10	10	Bluetooth Speaker	New	79.99	2	21	26/04/2023
11	11	Gaming Mouse	New	49.99	5	28	05/04/2023
12	12	Mechanical Keyboard	Refurbished	89.99	5	16	27/07/2023
13	13	Wireless Headphones	New	129.99	6	11	30/01/2023
14	14	Earbuds	Refurbished	39.99	6	24	18/09/2023
15	15	Smartwatch	New	199.99	8	27	21/02/2023
16	16	Fitness Tracker	New	69.99	12	28	19/11/2023
17	17	Digital Camera	Refurbished	149.99	9	27	06/09/2023

אחרי:

	PRODUCT_ID	PRODUCT_NAME	STATUS	PRICE	CATEGORY_ID	STOCK	AVAILABLE_DATE
1	1	Laptop	New	989.99	1	1	26/04/2023
2	2	Desktop Computer	Refurbished	769.99	1	22	20/10/2023
3	3	Monitor	New	219.99	1	0	03/05/2023
4	4	Keyboard	New	54.99	1	11	30/06/2023
5	5	Mouse	New	21.99	1	16	30/06/2023
6	6	External Hard Drive	Refurbished	79.99	11	18	23/12/2023
7	7	USB Flash Drive	New	14.99	11	20	24/10/2023
8	8	Printer	New	149.99	2	15	05/06/2023
9	9	Wireless Keyboard	Refurbished	32.99	1	25	12/03/2023
10	10	Bluetooth Speaker	New	79.99	2	21	26/04/2023
11	11	Gaming Mouse	New	49.99	5	28	05/04/2023
12	12	Mechanical Keyboard	Refurbished	89.99	5	16	27/07/2023
13	13	Wireless Headphones	New	129.99	6	11	30/01/2023
14	14	Earbuds	Refurbished	39.99	6	24	18/09/2023
15	15	Smartwatch	New	199.99	8	27	21/02/2023
16	16	Fitness Tracker	New	69.99	12	28	19/11/2023
17	17	Digital Camera	Refurbished	149.99	9	27	06/09/2023

שאלתה שנייה:

קונים שביצעו הזמנה בחודש האחרון והזמינו מוצר יותר מארבע פעמים יקבלו שדרוג למשלוח עד הבית
שאלתה זו מעדכנת את שיטת המשלוח ל'משלוח הביתה' עבור כל ההזמנות שבוצעו במהלך החודש האחרון
ואחד מהמוצרים בהזמנה הוזמן יותר מארבע פעמים.

```
--updates the delivery method to 'home delivery' for all orders placed
in the last month and bought more than 4 items for one of the order
products
UPDATE
    OrderDetails OD
SET
    delivery_method = 'home delivery'
WHERE

    OD.order_id IN (
        SELECT order_id
        FROM Orders o
        join include_products ip ON ip.order_id = o.order_id
        WHERE o.order_date >= TRUNC(SYSDATE, 'MM') - INTERVAL '1'
MONTH AND ip.quantity > 4  --trunc cut the date for a specific format
of date here first day of the curr month
-- interval = jump back one month

    );
```

לפני:

הזמנה שנוספה למערכת זה עתה

	ORDER_ID	DELIVERY_METHOD	TRACKING_NUMBER	ORDER_STATUS	ORDER_ID	ORDER_DATE	BUYER_ID
1	313575	pick up spots	TRACK-66666	pending	313575	11/08/2024 22:09:13	9999

אחרי :

	ORDER_ID	DELIVERY_METHOD	TRACKING_NUMBER	ORDER_STATUS	ORDER_ID	ORDER_DATE	BUYER_ID
1	313575	home delivery	TRACK-66666	pending	313575	11/08/2024 22:09:13	9999

שאלות עם פרמטרים

1. שאלתה מחזירה מוצרים לפי קטגוריה מסוימת עם list של שמות על הקטגוריות כאשר קטגוריה היא פרמטר שנלקח מרשימת הקטגוריות

```
SELECT p.product_name, p.price, c.category_name
FROM Products p
```

```
JOIN Categories c ON p.category_id = c.category_id
```

```
WHERE c.category_name = &<name = "cat" type = "string" list = "select category_name from Categories" default = "Computer Accessories" >
```

	PRODUCT_NAME	PRICE	CATEGORY_NAME
1	Laptop	989.99	Computer Accessories
2	Desktop Computer	769.99	Computer Accessories
3	Monitor	219.99	Computer Accessories
4	Keyboard	54.99	Computer Accessories
5	Mouse	21.99	Computer Accessories

2. שאלתה מחזירה את כל ההזמנות שבוצעו בטווח תאריכים מסוים כאשר התאריכים הם פרמטרים

```
-- 2. Query with a date parameter to find orders within a date range
SELECT o.order_id, o.order_date, b.buyer_name, p.product_name
FROM Orders o
JOIN Buyers b ON o.buyer_id = b.buyer_id
JOIN include_products ip ON ip.order_id = o.order_id
JOIN Products p ON ip.product_id = p.product_id
WHERE o.order_date BETWEEN &<name = "start_date" type = "date" hint = "the store exists form 1/1/23"> AND &<name = "end_date" type = "date" hint = "the store info updated until 1/1/23">
```

	ORDER_ID	ORDER_DATE	BUYER_NAME	PRODUCT_NAME
1	45503	15/12/2023	Ronald Kennerley	Lawn Mower
2	45503	15/12/2023	Ronald Kennerley	Wireless Earphones
3	58535	28/07/2023	Danell Jeness	Basic Tool Set
4	93917	22/02/2023	Aurthur Ennew	Portable Photo Printer
5	93917	22/02/2023	Aurthur Ennew	Beard Trimmer
6	52636	04/10/2023	Aurthur Ennew	Baby Monitor
7	25176	12/09/2023	Godfree Gavey	Fitness Tracker
8	66073	18/06/2023	Godfree Gavey	Earbuds
9	91096	03/05/2023	Clim Cassely	Digital Tire Pressure Gauge
10	55042	11/03/2023	Clim Cassely	Fire Table
11	76509	18/01/2023	Corly Peaseman	Travel Pillow
12	76509	18/01/2023	Corly Peaseman	Photo Printer
13	76509	18/01/2023	Corly Peaseman	Mesh WiFi System
14	59083	04/05/2023	Corly Peaseman	Screwdriver Set
15	72406	30/03/2023	Lara Sallinger	Smart Plug Pack
16	72406	30/03/2023	Lara Sallinger	Full Body Workout Machine
17	40685	27/09/2023	Cass Bum	Car Seat Cover Set

3. חיפוש מוכר לפי תבנית מסוימת , שתתקבל תהיה של כל מוכר שעומד בתבנית .

הפרמטר הוא תבנית

```
SELECT s.seller_name,
       s.phone,
       s.address

FROM Sellers s

WHERE LOWER(s.seller_name)

LIKE LOWER(&<name = "name_pattern" hint = "write between %_%" default = "%sam%" type = "string">) ;
```

	SELLER_NAME	PHONE	ADDRESS
1	Sam Evett	512345678	45 Rich Blvd
2	Anna Sample	512346128	411 Luis Street
3	Sam Kershaw	512346563	63rd Street
4	Solomon Sossamon	512348603	298 Jarvis Road
5	Samuel Atkins	512348888	7 Jimmy Blvd
6	Samantha Schwarzeneg	512349083	32nd Street
7	Sammy Sanchez	512349278	49 Beck Drive
8	Sammy Camp	512351333	33 Sepulveda Ave

4. שאילתה שמחזירה ביקורות לפי קטגוריה מסוימת החל מדירוג מסוים והחל מתאריך מסוים.

הפרמטרים הם דירוג, קטגוריה, תאריך

```
SELECT r.review_id,
       r.rating,
       r.comment_text,
       p.product_name

FROM Reviews r

JOIN Products p ON r.product_id = p.product_id

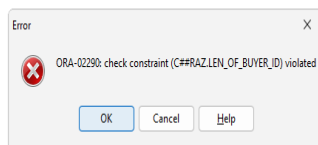
WHERE r.rating >= &<name = "min_rating" type = "INTEGER" hint = "number between 1-10" default = "8">
AND r.review_date > &<name = "review_after date" type = "date" hint = "1/1/2023 - 1/1/2025" default = "5/5/2024">
AND p.category_id = &<name = "category_id" type = "INTEGER" hint = "number between 1-30" default = "1">
```

	REVIEW_ID	RATING	COMMENT_TEXT	PRODUCT_NAME
1	43	8.7	Exceeded expectations	Ergonomic Mouse
2	252	9.5	Not as expected	Keyboard

אילוציםאילוצ ראשון – הגבלת אורך העמודה של buyer_id בטבלה buyers למינימום של ארבע ספרות

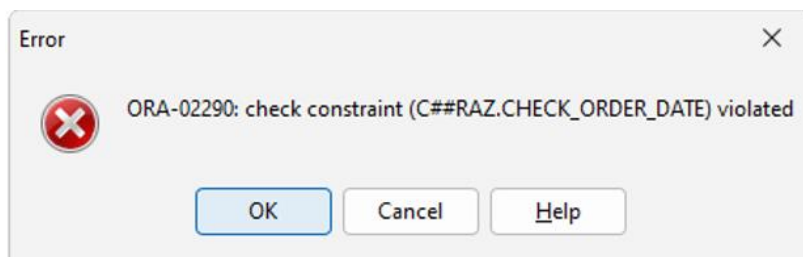
```
ALTER TABLE buyers
add constraint len_of_buyer_id
CHECK (length(buyer_id) >= 4);

INSERT INTO Buyers (buyer_id, buyer_name, email, phone, address) VALUES (999, 'avi', 'ohackett0@glabo.com', '054-343-2341', '70 herzel ofakim');
```

אילוצ שני – אין אפשרות להוסיף הזמנה עם תאריך הזמנה שקדם לתאריך הקמת החנות

```
--2
ALTER TABLE Orders
ADD CONSTRAINT check_order_date_since_2023
CHECK (order_date >= DATE '2023-01-01');

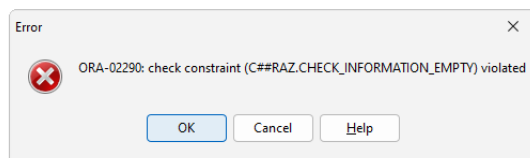
INSERT INTO Orders (order_id, order_date, buyer_id) VALUES (1233456, DATE '2021-07-25', 9999);
```



אילוצ שלישי – אין אפשרות להוסיף קטגוריה חדשה של מוצרים ללא תיאור הקטגוריה information

```
--3
ALTER TABLE Categories
ADD CONSTRAINT check_information_empty
CHECK (information IS NOT NULL);

INSERT INTO Categories (category_id,category_name,information ) VALUES (31,'tashmishi kdoshy','');
```

חלק שלישי

PL/SQL תכנות

פרוצדורה ראשונה

הפרוצדורה תרוץ על רשימת עם הקונים עם CURSER החל מתאריך מסויים שינתן כפרמטר

לכל קונה שרכש מוצר יותר מחמש פעמים יקבל הנחה של עשרה אחוזים

לאחר מכן התוכנית תדפיס את שמות הקונים ואת המחיר שישלמו לפני ההנחה אם ניתנה להם ואחרי ההנחה

```
--checks if a buyer has ordered more than 5 items, applies a 10% discount,
-- and then prints the buyer's name along with the original and discounted
prices
-- explicit cursor , loop ,excepetion , conditions
```

```
CREATE OR REPLACE PROCEDURE ApplyDiscountAndPrint (
    p_order_date IN DATE
) IS
    CURSOR order_cursor IS
        SELECT b.buyer_id, b.buyer_name, o.order_id, p.product_id, ip.quantity,
p.price
        FROM Orders o
        JOIN include_products ip ON o.order_id = ip.order_id
        JOIN Buyers b ON o.buyer_id = b.buyer_id
        JOIN Products p ON ip.product_id = p.product_id
        WHERE o.order_date >= p_order_date;

    v_buyer_name Buyers.buyer_name%TYPE;
    v_buyer_id Buyers.buyer_id%TYPE;
    v_order_id Orders.order_id%TYPE;
    v_product_id Products.product_id%TYPE;
```

```

v_quantity INT;
v_price NUMBER(10, 2);
v_total_price NUMBER(10, 2);
v_discounted_price NUMBER(10, 2);
v_discount_rate NUMBER := 0.1; -- 10% discount
v_empty_cursor BOOLEAN := TRUE;
BEGIN

    IF p_order_date < TO_DATE('01/01/2023', 'DD/MM/YYYY') THEN
        RAISE_APPLICATION_ERROR(-20001, 'invalid date ' || TO_CHAR(p_order_date,
'DD/MM/YYYY'));
    END IF;

    OPEN order_cursor;
    LOOP
        FETCH order_cursor INTO v_buyer_id, v_buyer_name, v_order_id,
v_product_id, v_quantity, v_price;
        EXIT WHEN order_cursor%NOTFOUND;

        v_empty_cursor := FALSE;-- if we got here the curser is not empty

        -- Calculate total price for the order
        v_total_price := v_quantity * v_price;

        -- Apply discount if quantity is greater than 5
        IF v_quantity > 5 THEN
            v_discounted_price := v_total_price * (1 - v_discount_rate);
        ELSE
            v_discounted_price := v_total_price; -- No discount
        END IF;

        -- Print buyer's name and price details
        DBMS_OUTPUT.PUT_LINE('Buyer: ' || v_buyer_name);
        DBMS_OUTPUT.PUT_LINE('Original Price: ' || v_total_price);
        DBMS_OUTPUT.PUT_LINE('Discounted Price: ' || v_discounted_price);
        DBMS_OUTPUT.PUT_LINE('-----');
    END LOOP;

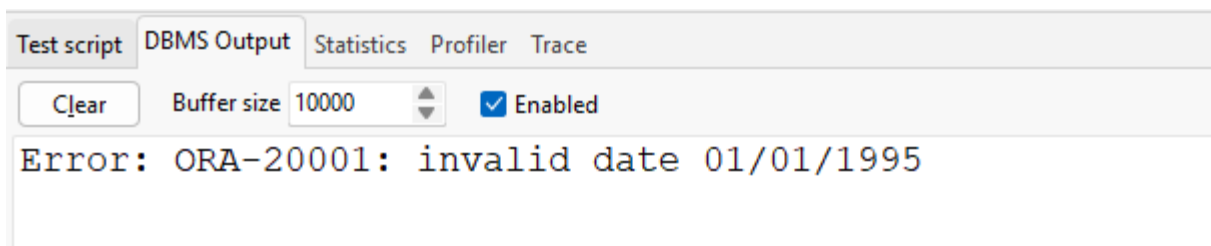
    -- Check if cursor was empty and raise an error if true
    IF v_empty_cursor THEN
        RAISE_APPLICATION_ERROR(-20002, 'No orders found for the given date. ');
    END IF;

    CLOSE order_cursor;

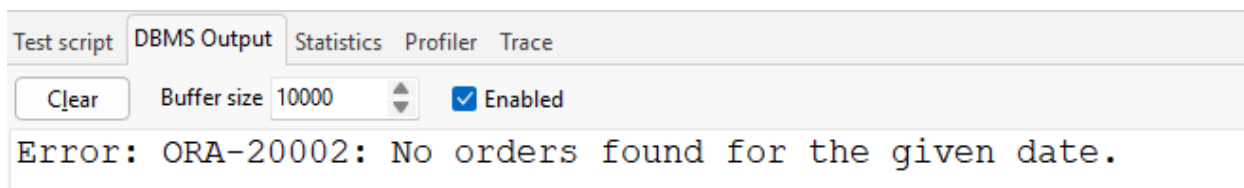
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END ApplyDiscountAndPrint;

```


אם הכנסנו כפרמטר תאריך שהוא לא תקין נקבל חריגה



אם הקורסור לא מצא שום שורה תואמת תופיע חריגה



פונקציה ראשונה

הפונקציה תקבל דירוג כפרמטר וטווח תאריכים ותחזיר את רשימת המוכרים שקיבלו על המוצרים שלהם ביקורות עם דירוג נמוך מתחת לדירוג שניתן כפרמטר בטווח תאריכים מסוים

```
--function that receives a rating as a parameter and returns a list of
sellers
-- who have products with ratings below the given rating between the
range of the param dates
-- refcursor ,exception
```

```
CREATE OR REPLACE FUNCTION GetSellersWithLowRatedProducts(
  p_rating IN NUMBER,
  p_start_date IN DATE,
  p_end_date IN DATE
) RETURN SYS_REFCURSOR IS
  v_sellers SYS_REFCURSOR;
BEGIN
  IF p_rating < 1 OR p_rating > 5 THEN
    RAISE_APPLICATION_ERROR(-20002, 'Invalid rating. Rating must be
between 1 and 5.');
```

```
    ELSIF p_start_date > p_end_date THEN
      RAISE_APPLICATION_ERROR(-20003, 'Invalid date range. Start date
must be before end date.');
```

```
    END IF;
```

```

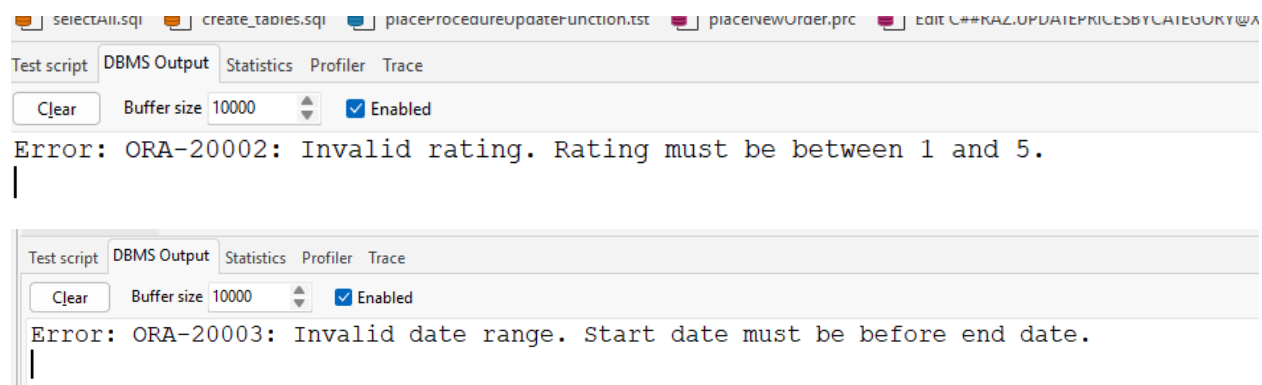
OPEN v_sellers FOR
  SELECT DISTINCT s.seller_id, se.seller_name, AVG(r.rating) OVER
(PARTITION BY p.product_id) AS avg_rating
  FROM Reviews r
  JOIN Products p ON r.product_id = p.product_id
  JOIN sell s ON p.product_id = s.product_id
  JOIN sellers se ON s.seller_id = se.seller_id
  WHERE r.rating < p_rating
        AND r.review_date BETWEEN p_start_date AND p_end_date;

RETURN v_sellers;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
    RETURN NULL;
END GetSellersWithLowRatedProducts;

```

אם המשתמש ישלח בתור פרמטר של דירוג מספר קטן מ1 או גדול מ5 הוא יקבל חריגה וכן תאריכים מוקדם לפני מאוחר

דוגמה



תוכנית ראשית שתקרא לפרוצדורה הראשונה ApplyDiscountAndPrint ולפונקציה הראשונה

GetSellersWithLowRatedProducts

התוכנית תשתמש בREFCURSER כדי להדפיס את כל שמות הקונים עם ביקורות נמוכות

שחזרו מהפונקציה

```
-- main to apply discount and print buyer information, and get sellers
with low-rated products
```

```
-- Declare local variables and cursors
```

```
DECLARE
```

```
    v_sellers SYS_REFCURSOR; -- Cursor for fetching sellers with low-
rated products
    v_seller_id INT;
    v_seller_name VARCHAR2(50);
    v_avg_rating NUMBER(10, 2);
    v_empty_cursor BOOLEAN := TRUE; -- Flag to check if cursor is empty
```

```
BEGIN
```

```
    -- Enable DBMS_OUTPUT with a buffer size of 1,000,000 bytes
    DBMS_OUTPUT.ENABLE(1000000);
```

```
    ApplyDiscountAndPrint(p_order_date => :p_order_date);
```

```
    DBMS_OUTPUT.PUT_LINE('*****
*****');
```

```
    -- Call the GetSellersWithLowRatedProducts function to get sellers
with products rated below 3
    -- with reviews within the date range from '2023-01-01' to '2025-01-
01'
```

```
    v_sellers := GetSellersWithLowRatedProducts(3, TO_DATE('2023-01-01',
'YYYY-MM-DD'), TO_DATE('2025-01-01', 'YYYY-MM-DD'));
```

```
    LOOP
```

```
        FETCH v_sellers INTO v_seller_id, v_seller_name, v_avg_rating;
```

```
        EXIT WHEN v_sellers%NOTFOUND;
```

```
        -- If we reach here, the cursor is not empty
```

```
        v_empty_cursor := FALSE;
```

```
        DBMS_OUTPUT.PUT_LINE('Seller ID: ' || v_seller_id || ', Name: ' ||
v_seller_name || ', Average Rating: ' || v_avg_rating);
```

```
    END LOOP;
```

```

IF v_empty_cursor THEN
    DBMS_OUTPUT.PUT_LINE('No sellers found with low-rated products. ');
END IF;

CLOSE v_sellers;

EXCEPTION
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);

END;
```

דוגמאות ריצה

	Variable	Type	Value
<input type="checkbox"/>	result	Cursor	<Cursor>
<input type="checkbox"/>	p_rating	Float	
<input type="checkbox"/>	p_start_date	Date	01/01/2023
<input type="checkbox"/>	p_end_date	Date	01/01/2025
<input checked="" type="checkbox"/>	p_order_date	Date	01/01/2024
*			

000 4:1 0:03 C##RAZ@Xe [15:17:10] Executed in 3.096 seconds

חלק מרשימת הקונים שחלקם קיבלו הנחה

Buyer: Omero Hackett
Original Price: 3959.96
Discounted Price: 3959.96

Buyer: Janek Hundell
Original Price: 164.94
Discounted Price: 164.94

Buyer: Janek Hundell
Original Price: 569.81
Discounted Price: 512.83

Buyer: Xever Spadelli
Original Price: 104.93
Discounted Price: 104.93

Buyer: Bjorn Loram
Original Price: 1429.89
Discounted Price: 1286.9

Buyer: Clim Cassely
Original Price: 799.9
Discounted Price: 799.9

Buyer: Corly Peaseman
Original Price: 679.83
Discounted Price: 611.85

Buyer: Wolfgang Burnell
Original Price: 2549.83
Discounted Price: 2294.85

Buyer: Avivah Stoggles
Original Price: 7999.8
Discounted Price: 7199.82

Buyer: Lara Sallinger
Original Price: 314.91
Discounted Price: 314.91

Buyer: Byrom Casbourne
Original Price: 599.8
Discounted Price: 539.82

וחלק מרשימת המוכרים שקיבלו דירוג נמוך מ3

Seller ID: 14621, Name: Lari Olin, Average Rating: 2.2
Seller ID: 16070, Name: Owen Belle, Average Rating: 2.2
Seller ID: 15629, Name: Praga Cruz, Average Rating: 1.6
Seller ID: 15062, Name: Allan Imperioli, Average Rating: 2.4
Seller ID: 11777, Name: Martha Bates, Average Rating: 2.4
Seller ID: 11786, Name: Kurt Thewlis, Average Rating: 2.4
Seller ID: 12632, Name: Buffy Lachey, Average Rating: 2.4
Seller ID: 11588, Name: Hikaru Hynde, Average Rating: 1.7
Seller ID: 14657, Name: Howard Rucker, Average Rating: 1.7
Seller ID: 14216, Name: Charles Weber, Average Rating: 1.7
Seller ID: 16061, Name: Shirley Schneider, Average Rating: 1.8
Seller ID: 11804, Name: Chloe Farina, Average Rating: 1.8
Seller ID: 15296, Name: Leonardo McClinton, Average Rating: 1.8
Seller ID: 14666, Name: Ernie Chinlund, Average Rating: 1.8
Seller ID: 16691, Name: Christian Pastore, Average Rating: 1.6
Seller ID: 16223, Name: Rob Danes, Average Rating: 1.6
Seller ID: 13676, Name: Jeroen Bradford, Average Rating: 2.3
Seller ID: 13172, Name: Carrie-Anne Paul, Average Rating: 2.3
Seller ID: 16268, Name: Chet Connery, Average Rating: 1.3
Seller ID: 11642, Name: Sam Kershaw, Average Rating: 1.3
Seller ID: 13829, Name: Petula Wopat, Average Rating: 1.8
Seller ID: 16232, Name: Aida Salt, Average Rating: 1.6
Seller ID: 13433, Name: Lili Elliott, Average Rating: 2.8
Seller ID: 15359, Name: Matt Witherspoon, Average Rating: 2.8
Seller ID: 11597, Name: Lee Balk, Average Rating: 2
Seller ID: 13865, Name: Aidan Weiland, Average Rating: 2
Seller ID: 13604, Name: Toshiro Garcia, Average Rating: 2
Seller ID: 15863, Name: Debbie Howard, Average Rating: 2
Seller ID: 14864, Name: Dave Santana, Average Rating: 2.6
Seller ID: 14432, Name: Miriam Joli, Average Rating: 2.6
Seller ID: 13874, Name: Ahmad Phillippe, Average Rating: 2.6
Seller ID: 11750, Name: Tobey Morales, Average Rating: 2.6
Seller ID: 16691, Name: Christian Pastore, Average Rating: 2.6
Seller ID: 14576, Name: Frank Rubinek, Average Rating: 2.9
Seller ID: 12704, Name: Nastassja Schwarzene, Average Rating: 1.2
Seller ID: 15746, Name: Meredith Rickman, Average Rating: 2.8
Seller ID: 15314, Name: Hal Viterelli, Average Rating: 2.8
Seller ID: 15584, Name: Billy Karyo, Average Rating: 2.8

פרוצדורה שנייה

מבצעת הזמנה חדשה עם רשימה של מספר מוצרים ורשימה נוספת של כמות הזמנות מכל מוצר אם המלאי של המוצר לא תואם את ההזמנה חריגה תופיע אחרת תתוסף הזמנה חדשה שבכל איטרציה של הלולאה תתוסף להזמנה המוצר הנוסף שנמצא ברשימת ההזמנות והמלאי של כל מוצר שהוזמן יתעדכן ואם לא תהיה שום בעיה יתבצע COMMIT למסד הנתונים

אחרת נבצע rollback למצב הקודם של בסיס הנתונים שנשמר ב savepoint

```
CREATE OR REPLACE PROCEDURE PlaceNewOrder (
  buyerId IN INT,
  orderId IN INT,
  orderDate IN DATE,
  trackingNumber IN VARCHAR2,
  products IN SYS.ODCINUMBERLIST, -- Input parameter: List of product
IDs (collection of numbers)
  quantities IN SYS.ODCINUMBERLIST -- Input parameter: List of
quantities (collection of numbers)
) IS
  v_product_id INT;           -- Local variable
  v_quantity INT;
  v_stock INT;
BEGIN
  -- Check if the buyer exists using EXISTS
  BEGIN
    SELECT 1
    INTO v_product_id
    FROM Buyers
    WHERE buyer_id = buyerId;

  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      RAISE_APPLICATION_ERROR(-20002, 'Buyer with ID ' || buyerId || '
does not exist.');
```

```
    END;

    SAVEPOINT sp_before_order; -- Savepoint to allow rollback in case
of errors

    INSERT INTO OrderDetails (order_id, delivery_method,
tracking_number, Order_Status)
    VALUES (orderId, 'home delivery', trackingNumber, 'pending');

    INSERT INTO OrderDetails (order_id, delivery_method,
tracking_number)
    VALUES (orderId, 'home delivery', trackingNumber);

    -- Loop through each product in the products list
    FOR i IN 1 .. products.COUNT LOOP
```

```

v_product_id := products(i); -- Get the current product ID from
the list
v_quantity := quantities(i); -- Get the current quantity from the
list

-- Check stock
SELECT p.stock
INTO v_stock
FROM Products p
WHERE p.product_id = v_product_id; -- Query to get the stock of
the current product

-- If there is not enough stock, raise an error
IF v_stock < v_quantity THEN
    RAISE_APPLICATION_ERROR(-20001, 'Insufficient stock for product
' || v_product_id);
END IF;

INSERT INTO include_products (order_id, product_id, quantity)
VALUES (orderId, v_product_id, v_quantity);

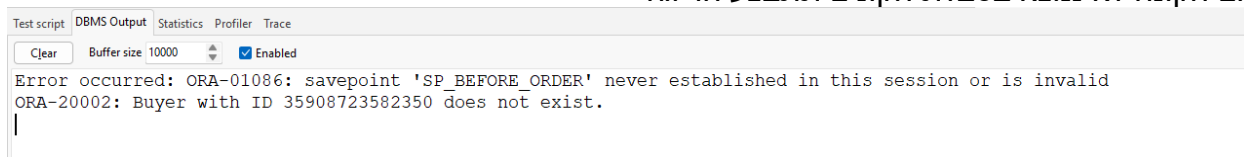
UPDATE Products
SET stock = stock - v_quantity
WHERE product_id = v_product_id;
END LOOP;

-- Commit transaction
COMMIT; -- Commit the transaction if all operations succeed

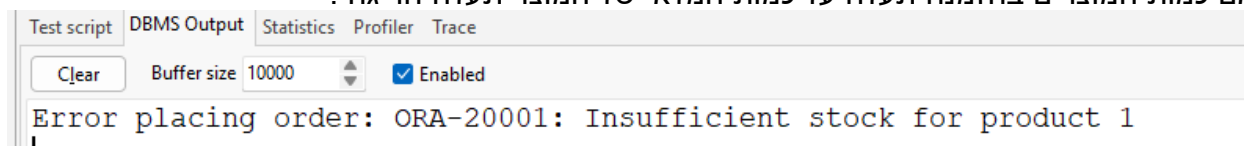
EXCEPTION
WHEN OTHERS THEN
    -- Rollback to savepoint in case of error
    ROLLBACK TO sp_before_order; -- Rollback to the savepoint if any
error occurs
    RAISE; -- Re-raise the exception to propagate the error
END PlaceNewOrder;

```

אם הקונה לא נמצא בטבלת הקונים תתבצע חריגה



אם כמות המוצרים בהזמנה תעלה על כמות המלאי של המוצר תעלה חריגה :



דוגמת ריצה

```

DECLARE
v_buyer_id INT := 9999999999; -- Assume buyer with ID 1
v_order_id INT := 2; -- New order ID
v_order_date DATE := SYSDATE; -- Current date
v_tracking_number VARCHAR2(20) := 'TRACK12345'; -- Tracking number for the order

-- Lists of product IDs and quantities
v_products SYS.ODCINUMBERLIST := SYS.ODCINUMBERLIST(1, 2, 3); -- Product IDs
v_quantities SYS.ODCINUMBERLIST := SYS.ODCINUMBERLIST(2, 1, 3); -- Quantities for the products
result VARCHAR2(50); -- for the UpdatePricesByCategory

BEGIN
-- Call the PlaceNewOrder procedure
PlaceNewOrder(
    buyerId => v_buyer_id,
    orderId => v_order_id,
    orderDate => v_order_date,
    trackingNumber => v_tracking_number,
    products => v_products,
    quantities => v_quantities
);

```

```

select *
from include_products
where order_id = 9999999999;

```

	ORDER_ID	PRODUCT_ID	QUANTITY
1	9999999999	1	2
2	9999999999	2	1
3	9999999999	3	3

פונקציה שנייה

בעקבות עליית מחירים בשוק נצטרך להעלות את המחיר של מוצרים ממחלקה מסוימת שנכנסו לחנות החל מתאריך מסויים שינתן לפונקציה ביחד עם הקטגוריה, הפונקציה תעלה את המחיר של כל המוצרים בקטגוריה בחמישה אחוזים ותדווח לנו את כמות המוצרים שעודכנו

```
CREATE OR REPLACE FUNCTION UpdatePricesByCategory (
  startDate IN DATE,
  categoryId IN INT
) RETURN VARCHAR2 IS
  CURSOR productCursor IS
    SELECT product_id, price
    FROM Products
    WHERE category_id = categoryId AND available_date >= startDate;

  productRecord productCursor%ROWTYPE;
  updatedCount INT := 0;

BEGIN
  -- Check if any products were updated
  IF categoryId < 0 THEN
    RAISE_APPLICATION_ERROR(-20004, 'invalid category');
  END IF;

  -- Open the cursor
  OPEN productCursor;

  -- Loop through each product
  LOOP
    FETCH productCursor INTO productRecord;
    EXIT WHEN productCursor%NOTFOUND;

    -- Update the product price by 5%
    UPDATE Products
    SET price = price * 1.05
    WHERE product_id = productRecord.product_id;

    -- Increment the count of updated products
    updatedCount := updatedCount + 1;
  END LOOP;

  -- Close the cursor
  CLOSE productCursor;

  -- Check if any products were updated
  IF updatedCount = 0 THEN
    RAISE_APPLICATION_ERROR(-20004, 'No products found for the given
category and date range.');
```

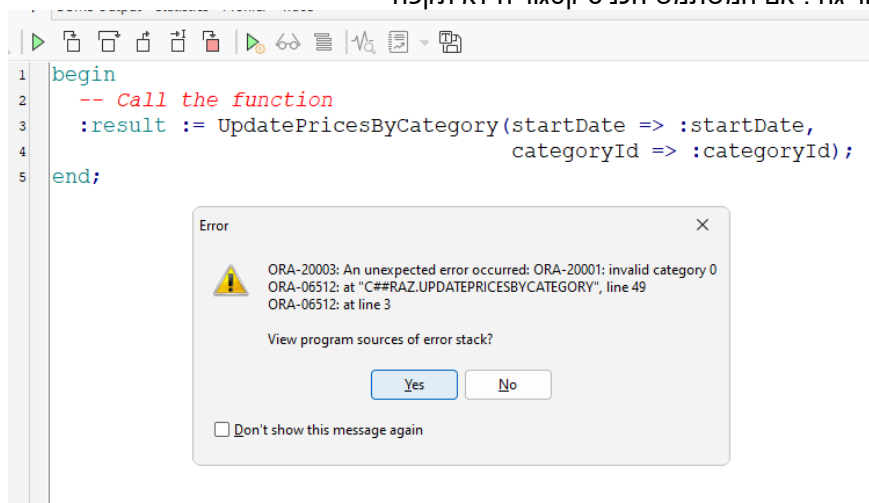
category and date range.');

```
  END IF;

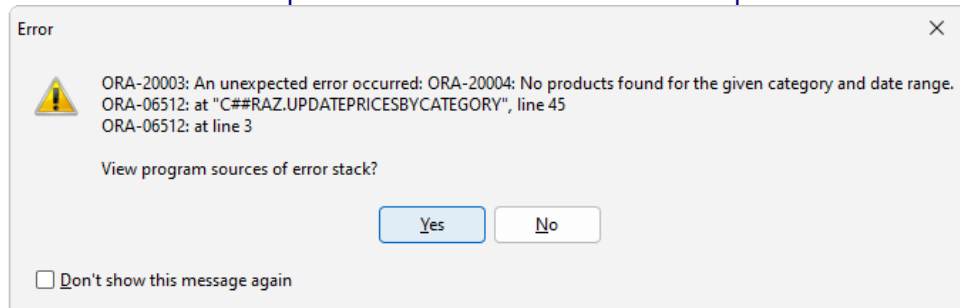
  -- Return a success message with the number of updated products
  RETURN 'Updated ' || updatedCount || ' products successfully.';
```

```
EXCEPTION
  WHEN OTHERS THEN
    -- Handle other unexpected errors
    RAISE_APPLICATION_ERROR(-20003, 'An unexpected error occurred: '
|| SQLERRM);
END UpdatePricesByCategory;
```

חריגה : אם המשתמש הכניס קטגוריה לא תקפה



חריגה שנייה : אם לא קיימים מוצרים בטגוריה מסויימת החל מתאריך מסויים



דוגמת הרצה:

1	begin
2	-- Call the function
3	:result := UpdatePricesByCategory(startDate => :startDate,
4	categoryId => :categoryId);
5	end;

Variable	Type	Value
result	String	Updated 13 products successfully.
startDate	Date	15/05/2023
categoryId	Float	1

לפני העדכון :

```
-- Select data from Products table
SELECT * FROM Products;
```

PRODUCT_ID	PRODUCT_NAME	INFORMATION	PRICE	CATEGORY_ID	STOCK	AVAILABLE_DATE
1	Laptop	New	989.99	1	24	24/10/2024
2	Desktop Computer	Refurbished	769.99	1	14	16/12/2024
3	Monitor	New	219.99	1	24	15/05/2023
4	Keyboard	New	54.99	1	15	13/12/2023
5	Mouse	New	21.99	1	2	09/01/2024
6	External Hard Drive	Refurbished	79.99	11	12	27/04/2024
7	USB Flash Drive	New	14.99	11	9	12/03/2024
8	Printer	New	149.99	2	29	26/06/2023
9	Wireless Keyboard	Refurbished	32.99	1	29	11/09/2024
10	Bluetooth Speaker	New	79.99	2	5	08/05/2023
11	Gaming Mouse	New	49.99	5	22	08/03/2023
12	Mechanical Keyboard	Refurbished	89.99	5	2	06/04/2024
13	Wireless Headphones	New	129.99	6	4	08/02/2023
14	Earbuds	Refurbished	39.99	6	22	21/08/2024
15	Smartwatch	New	199.99	8	28	28/03/2023
16	Fitness Tracker	New	69.99	12	8	28/04/2024
17	Digital Camera	Refurbished	149.99	9	23	09/04/2023
18	DSLR Camera	New	699.99	9	9	30/06/2023
19	LED Desk Lamp	New	34.99	10	26	29/01/2023
20	Smart Bulbs	Refurbished	19.99	10	16	23/05/2023
21	WiFi Router	New	79.99	4	18	12/03/2024
22	Network Switch	New	49.99	4	15	05/03/2024
23	Gaming Console	New	399.99	5	22	09/06/2024
24	VR Headset	Refurbished	299.99	28	18	16/05/2024
25	Action Camera	New	199.99	28	19	23/04/2024
26	3D Printer	New	399.99	29	21	17/03/2023
27	Filament Refill	New	29.99	29	27	15/12/2023

אחרי העדכון :

-- Select data from Products table

SELECT * FROM Products;

	PRODUCT_ID	PRODUCT_NAME	INFORMATION	PRICE	CATEGORY_ID	STOCK	AVAILABLE_DATE
1	1	Laptop	New	1039.49	1	24	24/10/2024
2	2	Desktop Computer	Refurbished	808.49	1	14	16/12/2024
3	3	Monitor	New	230.99	1	24	15/05/2023
4	4	Keyboard	New	57.74	1	15	13/12/2023
5	5	Mouse	New	23.09	1	2	09/01/2024
6	6	External Hard Drive	Refurbished	79.99	11	12	27/04/2024
7	7	USB Flash Drive	New	14.99	11	9	12/03/2024
8	8	Printer	New	149.99	2	29	26/06/2023
9	9	Wireless Keyboard	Refurbished	34.64	1	29	11/09/2024
10	10	Bluetooth Speaker	New	79.99	2	5	08/05/2023
11	11	Gaming Mouse	New	49.99	5	22	08/03/2023
12	12	Mechanical Keyboard	Refurbished	89.99	5	2	06/04/2024
13	13	Wireless Headphones	New	129.99	6	4	08/02/2023
14	14	Earbuds	Refurbished	39.99	6	22	21/08/2024
15	15	Smartwatch	New	199.99	8	28	28/03/2023
16	16	Fitness Tracker	New	69.99	12	8	28/04/2024
17	17	Digital Camera	Refurbished	149.99	9	23	09/04/2023

תוכנית ראשית שקוראת לפרוצדורה PlaceNewOrder ו UpdatePricesByCategory 1

```

DECLARE
  v_buyer_id INT := 35908723582350; -- Assume buyer with ID 1
  v_order_id INT := 99999999999; -- New order ID
  v_order_date DATE := SYSDATE; -- Current date
  v_tracking_number VARCHAR2(20) := 'TRACK12345'; -- Tracking number for the
order

  -- Lists of product IDs and quantities
  v_products SYS.ODCINUMBERLIST := SYS.ODCINUMBERLIST(1, 2, 3); -- Product
IDs
  v_quantities SYS.ODCINUMBERLIST := SYS.ODCINUMBERLIST(2, 1, 3); --
Quantities for the products
  result VARCHAR2(50); -- for the UpdatePricesByCategory

BEGIN
  -- Call the PlaceNewOrder procedure
  PlaceNewOrder(
    buyerId => v_buyer_id,
    orderId => v_order_id,
    orderDate => v_order_date,
    trackingNumber => v_tracking_number,
    products => v_products,
    quantities => v_quantities
  );

  -- Output success message
  DBMS_OUTPUT.PUT_LINE('Order placed successfully.');
```

```

  :result := UpdatePricesByCategory(startDate => :startDate,
                                     categoryId => :categoryId);

  DBMS_OUTPUT.PUT_LINE(result);

END;
```