

SQL EKSĀMENS

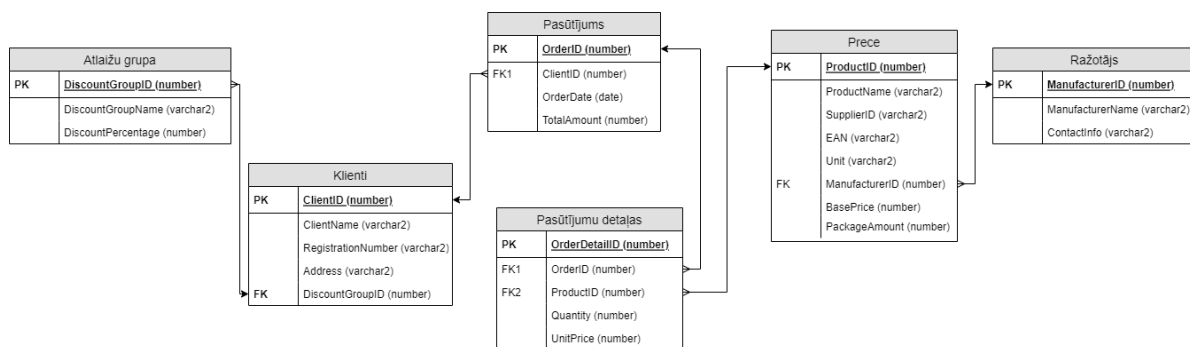
BIZNESA LOĢIKAS

Uzņēmums RAD SIA – Būvmateriālu un elektroinstalācijas materiālu tirgotājs. Uzņēmums plāno implementēt ERP (*angl. Enterprise Resource Planning*) sistēmu, kurā tiks veikt visa uzņēmuma darbība. Uzņēmuma galvenās funkcijas ir preču pirkšana, pārdošana, glabāšana.

Uzņēmuma piedāvātas preču klāsts ir no vairākiem ražotājiem un preču grupām. Visām precēm ir pieejama pamata, jeb tarifa, cena. Katram klientam ir piesaistīta arī atlaižu grupa, kura norāda – atlaižu grupu katram klientam.

Uzņēmums ir vairumtirogotājs, strādājošs B2B segmentā – tāpēc uzņēmuma klienti ir juridiskas personas personas.

ENTĪTIJU-ATTIECĪBU (ER) DIAGRAMMA



ENTĪTIJAS UN TO ATRIBŪTES

1 Entitātes un to atribūti:

Ražotājs (Manufacturer)

- **Atribūti:** ManufacturerName, ContactInfo
- Šī entītija apraksta preču ražotājus. Katram ražotājam ir unikāls identifikators (ManufacturerID), kā arī kontakta informācija un nosaukums.

Preces (Products)

- **Atribūti:** ProductName, EAN, Unit, BasePrice, PackageAmount.
- Entītija, kas apraksta preces ar to unikālo ID (ProductID). Ietver detalizētu informāciju, piemēram, vienības mērvienību, cenu, iepakojuma daudzumu un svītrkodu (EAN).

Atlaidu grupa (DiscountGroups)

- **Atribūti:** DiscountGroupName, DiscountPercentage.
- Atlaides grupa klasificē klientus, piešķirot viņiem konkrētu atlaides procentu. Katram ierakstam ir unikāls ID (DiscountGroupID).

Klienti (Clients)

- **Atribūti:** ClientName, RegistrationNumber, Address.
- Apraksta klientus ar unikālu reģistrācijas numuru un adresi. Klienti pieder vienai atlaides grupai.

Pasūtījumi (Orders)

- **Atribūti:** OrderDate, TotalAmount.
- Apraksta pasūtījumus, kas veikti no klientiem. Pasūtījuma datums un kopējā summa ir galvenie atribūti.

Pasūtījuma detaļas (OrderDetails)

- **Atribūti:** Quantity, UnitPrice.
- Apvieno pasūtījumus ar precēm, norādot daudzumu un cenu katrai precei konkrētajā pasūtījumā.

2. Saistību analīze:

Klienti Atlaidu grupa (M:1)

- Katrs klients pieder vienai atlaides grupai, bet viena atlaides grupa var būt saistīta ar vairākiem klientiem.

Klienti Pasūtījumi (1:M)

- Viens klients var veikt vairākus pasūtījumus. Katram pasūtījumam ir unikāls ID, kas identificē klienta aktivitāti.

Pasūtījumi Pasūtījuma detaļas (1:M)

- Katrs pasūtījums var saturēt vairākus produktus. Šīs attiecības tiek detalizētas tabulā OrderDetails.

Preces Ražotājs (M:1)

- Vienu preci var ražot tikai viens ražotājs, bet viens ražotājs var ražot vairākas preces.

Preces Pasūtījuma detaļas (M:M)

- Preces var būt iekļautas vairākos pasūtījumos, savukārt katrs pasūtījums var ietvert vairākas preces.

DATUBĀZES RELĀCIJAS MODELIS



Figure 1: Entītuju diagramma

TABULU STRUKTŪRA

Prece

Lauks	Apraksts	Datu tips	Izmērs	Statuss	Ierobežojumi
ProductID	Preces identifikācijas numurs	NUMBER	Long Integer	Primary Key	NOT NULL
ProductName	Preces nosaukums	VARCHAR2	100		NOT NULL
SupplierID	Ražotāja kods	VARCHAR2	50		NOT NULL
EAN	Preces svītrukods	VARCHAR2	11		NOT NULL
Unit	Mērvienība	VARCHAR2	20		NOT NULL

Prece						
ManufacturerID	Ražotāja identifikācijas numurs	NUMBER	Long Integer	Foreign Key		NOT NULL
BasePrice	Preces cena	NUMBER	10,2			NOT NULL
PackageAmount	Preces daudzums iepakojumā	NUMBER				NOT NULL

Ražotājs						
Lauks	Apraksts	Datu tips	Izmērs	Statuss		Ierobežojumi
ManufacturerID	Ražotāja identifikācijas numurs	NUMBER	Long Integer	Primary Key		NOT NULL
ManufacturerName	Ražotāja nosaukums	VARCHAR2	100			NOT NULL
ContactInfo	Ražotāja kontaktinformācija	VARCHAR2	255			NOT NULL

Klienti						
Lauks	Apraksts	Datu tips	Izmērs	Statuss		Ierobežojumi

Klienti					
ClientID	Klienta iden- ti- fikā- cijas nu- murs	NUMBER		Primary Key	NOT NULL
ClientName	Klienta nosaukums	VARCHAR2	100		NOT NULL
RegistrationNumber	Klienta reģistrā- cijas nu- murs	VARCHAR2	11		NOT NULL
Address	Klienta ju- ridiskā adrese	VARCHAR2	255		NOT NULL
DiscountGroupID	Atlaides gru- pas iden- ti- fikā- cijas nu- murs	NUMBER		Foreign Key	NOT NULL
Atlaižu grupa					
Lauks	Apraksts	Datu tips	Izmērs	Statuss	Ierobežojumi

Atlaižu grupa						
DiscountGroupID	Atlaižu grupas identifikācijas numurs	NUMBER	Long Integer	Primary Key		NOT NULL
DiscountGroupName	Atlaižu grupas nosaukums	VARCHAR2	100			NOT NULL
DiscountPercentage	Pages atlaide % no pamata cenas	NUMBER	5,2			CHECK

Pasūtījums						
Lauks	Apraksts	Datu tips	Izmērs	Statuss	Ierobežojumi	
OrderID	Pasūtījums identifikācijas numurs	NUMBER		Primary Key		NOT NULL

Pasūtījums				
ClientID	Klienta iden- ti- fikā- cijas nu- murs	NUMBER		Foreign Key NOT NULL
OrderDate	Pasūtījuma da- tums	DATE		DEFAULT NULL
TotalAmount	Pasūtījuma summa	NUMBER	10,2	NOT NULL

Pasūtījumu detaļas

Lauks

Apraksts

Datu tips

Izmērs

Statuss

Ierobežojumi

OrderDetailID

Pasūtījums detaļu identifikācijas numurs

NUMBER

Long Integer

Primary Key

NOT NULL

OrderID

Pasūtījuma identifikācijas numurs

NUMBER

Long Integer

Foreign Key

NOT NULL

ProductID

prece identifikācijas numurs

NUMBER

Long Integer

Foreign Key

NOT NULL

Quantity

Pārdotais daudzums

NUMBER

5

NOT NULL

UnitPrice

Pārdošanas cena

UnitPrice

10,2

NOT NULL

SQL KODS

SQL datubāzes veidošana R vidē

```
library(DBI)
library(RSQLite)
library(ggplot2)
con <- dbConnect(RSQLite::SQLite(), ":memory:")
```

```
dbExecute(con, "
CREATE TABLE Manufacturers (
  ManufacturerID INTEGER PRIMARY KEY,
  ManufacturerName TEXT NOT NULL,
  ContactInfo TEXT NOT NULL
)
")
```

```
dbExecute(con, "
CREATE TABLE DiscountGroups (
  DiscountGroupID INTEGER PRIMARY KEY,
  DiscountGroupName TEXT,
  DiscountPercentage REAL
)
")
```

```
dbExecute(con, "
CREATE TABLE Products (
  ProductID INTEGER PRIMARY KEY,
  ProductName TEXT NOT NULL,
  SupplierID TEXT NOT NULL,
  EAN TEXT NOT NULL UNIQUE,
  Unit TEXT NOT NULL,
  ManufacturerID INTEGER NOT NULL,
  BasePrice REAL NOT NULL,
  PackageAmount INTEGER NOT NULL,
  FOREIGN KEY (ManufacturerID) REFERENCES Manufacturers(ManufacturerID)
)
")
```

```
dbExecute(con, "
CREATE TABLE Clients (
```

```

        ClientID INTEGER PRIMARY KEY,
        ClientName TEXT NOT NULL,
        RegistrationNumber TEXT NOT NULL UNIQUE,
        Address TEXT NOT NULL,
        DiscountGroupID INTEGER NOT NULL,
        FOREIGN KEY (DiscountGroupID) REFERENCES DiscountGroups(DiscountGroupID)
    )
")

```

```

dbExecute(con, "
CREATE TABLE Orders (
    OrderID INTEGER PRIMARY KEY,
    ClientID INTEGER NOT NULL,
    OrderDate TEXT NOT NULL,
    TotalAmount REAL NOT NULL,
    FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)
)
")

```

```

dbExecute(con, "
CREATE TABLE OrderDetails (
    OrderDetailID INTEGER PRIMARY KEY,
    OrderID INTEGER NOT NULL,
    ProductID INTEGER NOT NULL,
    Quantity INTEGER NOT NULL,
    UnitPrice REAL NOT NULL,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
)
")

```

```

# Insert data into Manufacturers table
dbExecute(con, "
    INSERT INTO Manufacturers (ManufacturerID, ManufacturerName, ContactInfo) VALUES
    (101, 'Elektronika Ltd.', 'info@elektronika.com'),
    (102, 'Kabeļu Risinājumi', 'contact@kabris.lv'),
    (103, 'Drošības Sistēmas', 'support@drosiba.lv'),
    (104, 'BalticTech Solutions', 'sales@baltictech.lv'),
    (105, 'ElektroSmart', 'info@elektrosmart.com'),
    (106, 'SafeWire Systems', 'contact@safewire.lv');
")

```

```

# Insert data into DiscountGroups table
dbExecute(con, "
    INSERT INTO DiscountGroups (DiscountGroupID, DiscountGroupName, DiscountPercentage) VALUES
    (1, 'Klients C', 5.00),
    (2, 'Klients B', 10.00),
    (3, 'Klients A', 15.00),
    (4, 'Klients D', 3.00),
    (5, 'Klients E', 7.50);
")

# Insert data into Clients table
dbExecute(con, "
    INSERT INTO Clients (ClientID, ClientName, RegistrationNumber, Address, DiscountGroupID) VALUES
    (1, 'Tesco SIA', '12345678910', 'Pilsētas iela 10', 1),
    (2, 'Walmarts SIA', '23456789012', 'Dārza iela 15', 2),
    (3, 'Kaprs SIA', '34567890123', 'Rīgas iela 50', 3),
    (4, 'RISBEA SIA', '09876543211', 'Ostas iela 25', 2),
    (5, 'LULA SIA', '01928374651', 'Mazā iela 5', 1),
    (6, 'Maxima SIA', '45678901234', 'Liepājas prospekts 7', 4),
    (7, 'Elko Grupa', '56789012345', 'Ganību dambis 22', 5),
    (8, 'Rimi Latvia', '67890123456', 'Ulmaņa gatve 101', 3),
    (9, 'Prisma SIA', '78901234567', 'Brīvības bulvāris 12', 2),
    (10, 'Elektro Serviss', '89012345678', 'Maskavas iela 246', 5);
")

# Insert data into Products table
dbExecute(con, "
    INSERT INTO Products (ProductID, ProductName, SupplierID, EAN, Unit, ManufacturerID, BasePrice) VALUES
    (1, 'Elektrības kabelis 3x1.5mm', 'CAB101', '12345678901', 'm', 101, 0.85, 100),
    (2, 'Elektrības kabelis 3x2.5mm', 'CAB102', '23456789012', 'm', 101, 1.20, 100),
    (3, 'Sadalne kārba 100x100mm', 'FE10', '34567890123', 'gb', 102, 3.50, 1),
    (4, 'PVC truba 16mm', 'PIPE301', 'RLM16p', 'm', 102, 0.75, 50),
    (5, 'Vadu savienojuma skava', 'u-16', '56789012345', 'gb', 103, 0.10, 100),
    (6, 'Drošinātājs 16A', '16A2', '67890123456', 'gb', 103, 1.25, 10),
    (7, 'Vadu marķieris', 'a123', '78901234567', 'gb', 103, 0.50, 50);
")

# Insert data into Orders table
dbExecute(con, "
    INSERT INTO Orders (OrderID, ClientID, OrderDate, TotalAmount) VALUES
    (1001, 1, '2024-11-01', 125.00),
    (1002, 1, '2024-11-02', 75.00),

```

```

(1003, 3, '2024-11-03', 250.00),
(1004, 5, '2024-11-04', 90.00),
(1005, 2, '2024-11-05', 180.00),
(1006, 6, '2024-11-06', 220.00),
(1007, 8, '2024-11-07', 95.50),
(1008, 9, '2024-11-08', 150.75),
(1009, 7, '2024-11-09', 300.00),
(1010, 10, '2024-11-10', 120.50);
")

# Insert data into OrderDetails table
dbExecute(con, "
  INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity, UnitPrice) VALUES
  (2001, 1001, 1, 100, 0.85),
  (2002, 1001, 3, 5, 3.50),
  (2003, 1002, 2, 50, 1.20),
  (2004, 1003, 4, 100, 0.75),
  (2005, 1004, 6, 10, 1.25),
  (2006, 1004, 7, 20, 0.50),
  (2007, 1005, 2, 100, 1.20),
  (2008, 1005, 5, 50, 0.10),
  (2009, 1006, 9, 20, 1.75),
  (2010, 1006, 10, 15, 2.25),
  (2011, 1007, 8, 30, 2.50),
  (2012, 1008, 11, 10, 3.00),
  (2013, 1008, 12, 50, 1.10),
  (2014, 1009, 1, 150, 0.85),
  (2015, 1009, 3, 10, 3.50),
  (2016, 1009, 5, 75, 0.10),
  (2017, 1010, 6, 15, 1.25),
  (2018, 1010, 7, 25, 0.50),
  (2019, 1010, 9, 10, 1.75),
  (2020, 1010, 12, 40, 1.10);
")

```

Pamata vaicājuma izpilde R vidē

```

dbGetQuery(con, "
  SELECT * FROM OrderDetails
")

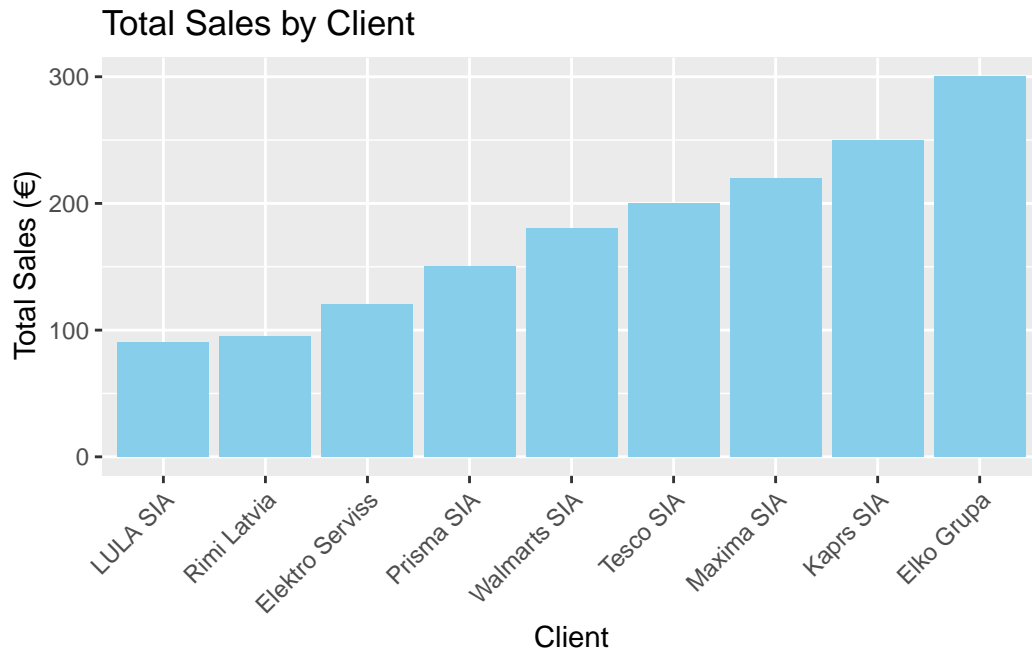
```

	OrderDetailID	OrderID	ProductID	Quantity	UnitPrice
1		2001	1001	1	100
2		2002	1001	3	5
3		2003	1002	2	50
4		2004	1003	4	100
5		2005	1004	6	10
6		2006	1004	7	20
7		2007	1005	2	100
8		2008	1005	5	50
9		2009	1006	9	20
10		2010	1006	10	15
11		2011	1007	8	30
12		2012	1008	11	10
13		2013	1008	12	50
14		2014	1009	1	150
15		2015	1009	3	10
16		2016	1009	5	75
17		2017	1010	6	15
18		2018	1010	7	25
19		2019	1010	9	10
20		2020	1010	12	40

Grafika veidošana, pamatojoties uz datubāzes datiem

```
# Izveidojam vaicājumu
query <- "
  SELECT c.ClientName, SUM(o.TotalAmount) AS TotalSales
  FROM Orders o
  JOIN Clients c ON o.ClientID = c.ClientID
  GROUP BY c.ClientName
  ORDER BY TotalSales DESC;
"

# Ievietojam vaicājuma rezultātu data frame
sales_data <- dbGetQuery(con, query)
# Atslēdzamies no datubāzes
dbDisconnect(con)
# Izveidojam peimēra grafiku
ggplot(sales_data, aes(x = reorder(ClientName, TotalSales), y = TotalSales)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Total Sales by Client", x = "Client", y = "Total Sales (€)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Darba ar Oracle SQL

Tabulu izveidošana

```
CREATE TABLE Manufacturers (
    ManufacturerID NUMBER PRIMARY KEY,
    ManufacturerName VARCHAR2(100) NOT NULL,
    ContactInfo VARCHAR2(255) NOT NULL
);

CREATE TABLE DiscountGroups (
    DiscountGroupID NUMBER PRIMARY KEY,
    DiscountGroupName VARCHAR2(100),
    DiscountPercentage NUMBER(5, 2)
);

CREATE TABLE Products (
    ProductID NUMBER PRIMARY KEY,
    ProductName VARCHAR2(100) NOT NULL,
    SupplierID VARCHAR2(50) NOT NULL,
    EAN VARCHAR2(13) NOT NULL UNIQUE,
```

```

Unit VARCHAR2(20) NOT NULL,
ManufacturerID NUMBER NOT NULL,
BasePrice NUMBER(10, 2) NOT NULL,
PackageAmount NUMBER NOT NULL,
FOREIGN KEY (ManufacturerID) REFERENCES Manufacturers(ManufacturerID)
);

CREATE TABLE Clients (
  ClientID NUMBER PRIMARY KEY,
  ClientName VARCHAR2(100) NOT NULL,
  RegistrationNumber VARCHAR2(11) NOT NULL UNIQUE,
  Address VARCHAR2(255) NOT NULL,
  DiscountGroupID NUMBER NOT NULL,
  FOREIGN KEY (DiscountGroupID) REFERENCES DiscountGroups(DiscountGroupID)
);

CREATE TABLE Orders (
  OrderID NUMBER PRIMARY KEY,
  ClientID NUMBER NOT NULL,
  OrderDate DATE NOT NULL,
  TotalAmount NUMBER(10, 2) NOT NULL,
  FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)
);

CREATE TABLE OrderDetails (
  OrderDetailID NUMBER PRIMARY KEY,
  OrderID NUMBER NOT NULL,
  ProductID NUMBER NOT NULL,
  Quantity NUMBER NOT NULL,
  UnitPrice NUMBER(10, 2) NOT NULL,
  FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
  FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

```

Tabulu aizpildīšana

```

INSERT INTO Manufacturers VALUES (101, 'Elektronika Ltd.', 'info@elektronika.com');
INSERT INTO Manufacturers VALUES (102, 'Kabeļu Risinājumi', 'contact@kabris.lv');

```



```

INSERT INTO Manufacturers VALUES (103, 'Drošības Sistēmas', 'support@drosiba.lv');
INSERT INTO Manufacturers VALUES (104, 'BalticTech Solutions', 'sales@baltictech.lv');
INSERT INTO Manufacturers VALUES (105, 'ElektroSmart', 'info@elektrosmart.com');
INSERT INTO Manufacturers VALUES (106, 'SafeWire Systems', 'contact@safewire.lv');

INSERT INTO DiscountGroups VALUES (1, 'Klienti C', 5.00);
INSERT INTO DiscountGroups VALUES (2, 'Klienti B', 10.00);
INSERT INTO DiscountGroups VALUES (3, 'Klienti A', 15.00);
INSERT INTO DiscountGroups VALUES (4, 'Klienti D', 3.00);
INSERT INTO DiscountGroups VALUES (5, 'Klienti E', 7.50);

INSERT INTO Clients VALUES (1, 'Tesco SIA', '12345678910', 'Pilsētas iela 10', 1);
INSERT INTO Clients VALUES (2, 'Walmarts SIA', '23456789012', 'Dārza iela 15', 2);
INSERT INTO Clients VALUES (3, 'Kaprs SIA', '34567890123', 'Rīgas iela 50', 3);
INSERT INTO Clients VALUES (4, 'RISBEA SIA', '09876543211', 'Ostas iela 25', 2);
INSERT INTO Clients VALUES (5, 'LULA SIA', '01928374651', 'Mazā iela 5', 1);
INSERT INTO Clients VALUES (6, 'Maxima SIA', '45678901234', 'Liepājas prospekts 7', 4);
INSERT INTO Clients VALUES (7, 'Elko Grupa', '56789012345', 'Ganību dambis 22', 5);
INSERT INTO Clients VALUES (8, 'Rimi Latvia', '67890123456', 'Ulmaņa gatve 101', 3);
INSERT INTO Clients VALUES (9, 'Prisma SIA', '78901234567', 'Brīvības bulvāris 12', 2);
INSERT INTO Clients VALUES (10, 'Elektro Serviss', '89012345678', 'Maskavas iela 246', 5);

INSERT INTO Products VALUES (1, 'Elektrības kabelis 3x1.5mm', 'CAB101', '12345678901', 'm', 102, 0.75, 50);
INSERT INTO Products VALUES (2, 'Elektrības kabelis 3x2.5mm', 'CAB102', '23456789012', 'm', 102, 0.75, 50);
INSERT INTO Products VALUES (3, 'Sadalne kārba 100x100mm', 'FE10', '34567890123', 'gb', 102, 0.75, 50);
INSERT INTO Products VALUES (4, 'PVC truba 16mm', 'PIPE301', 'RLM16p', 'm', 102, 0.75, 50);
INSERT INTO Products VALUES (5, 'Vadu savienojuma skava', 'u-16', '56789012345', 'gb', 103, 0.75, 50);
INSERT INTO Products VALUES (6, 'Drošinātājs 16A', '16A2', '67890123456', 'gb', 103, 1.25, 100);
INSERT INTO Products VALUES (7, 'Vadu marķieris', 'a123', '78901234567', 'gb', 103, 0.50, 50);

INSERT INTO Orders VALUES (1001, 1, TO_DATE('2024-11-01', 'YYYY-MM-DD'), 125.00);
INSERT INTO Orders VALUES (1002, 1, TO_DATE('2024-11-02', 'YYYY-MM-DD'), 75.00);
INSERT INTO Orders VALUES (1003, 3, TO_DATE('2024-11-03', 'YYYY-MM-DD'), 250.00);
INSERT INTO Orders VALUES (1004, 5, TO_DATE('2024-11-04', 'YYYY-MM-DD'), 90.00);
INSERT INTO Orders VALUES (1005, 2, TO_DATE('2024-11-05', 'YYYY-MM-DD'), 180.00);
INSERT INTO Orders VALUES (1006, 6, TO_DATE('2024-11-06', 'YYYY-MM-DD'), 220.00);
INSERT INTO Orders VALUES (1007, 8, TO_DATE('2024-11-07', 'YYYY-MM-DD'), 95.50);
INSERT INTO Orders VALUES (1008, 9, TO_DATE('2024-11-08', 'YYYY-MM-DD'), 150.75);
INSERT INTO Orders VALUES (1009, 7, TO_DATE('2024-11-09', 'YYYY-MM-DD'), 300.00);
INSERT INTO Orders VALUES (1010, 10, TO_DATE('2024-11-10', 'YYYY-MM-DD'), 120.50);

INSERT INTO OrderDetails VALUES (2001, 1001, 1, 100, 0.85);

```

```

INSERT INTO OrderDetails VALUES (2002, 1001, 3, 5, 3.50);
INSERT INTO OrderDetails VALUES (2003, 1002, 2, 50, 1.20);
INSERT INTO OrderDetails VALUES (2004, 1003, 4, 100, 0.75);
INSERT INTO OrderDetails VALUES (2005, 1004, 6, 10, 1.25);
INSERT INTO OrderDetails VALUES (2006, 1004, 7, 20, 0.50);
INSERT INTO OrderDetails VALUES (2007, 1005, 2, 100, 1.20);
INSERT INTO OrderDetails VALUES (2008, 1005, 5, 50, 0.10);
INSERT INTO OrderDetails VALUES (2009, 1006, 9, 20, 1.75);
INSERT INTO OrderDetails VALUES (2010, 1006, 10, 15, 2.25);
INSERT INTO OrderDetails VALUES (2011, 1007, 8, 30, 2.50);
INSERT INTO OrderDetails VALUES (2012, 1008, 11, 10, 3.00);
INSERT INTO OrderDetails VALUES (2013, 1008, 12, 50, 1.10);
INSERT INTO OrderDetails VALUES (2014, 1009, 1, 150, 0.85);
INSERT INTO OrderDetails VALUES (2015, 1009, 3, 10, 3.50);
INSERT INTO OrderDetails VALUES (2016, 1009, 5, 75, 0.10);
INSERT INTO OrderDetails VALUES (2017, 1010, 6, 15, 1.25);
INSERT INTO OrderDetails VALUES (2018, 1010, 7, 25, 0.50);
INSERT INTO OrderDetails VALUES (2019, 1010, 9, 10, 1.75);
INSERT INTO OrderDetails VALUES (2020, 1010, 12, 40, 1.10);

```

Pamata vaicājumi

```

-- Atjaunina produkta cenu
UPDATE Products
SET BasePrice = 2.25
WHERE ProductID = 4;

-- Pievieno jaunu klientu
INSERT INTO Clients VALUES (11, 'Baltijas Energo', '55544433322', 'Brīvības gatve 77', 2);

-- Dzēš klientu no tabulas Clients
DELETE FROM Clients
WHERE ClientID = 11;

-- Atlasa visas rindas no tabulas "OrderDetails", kurām ProductID ir 6
SELECT *
FROM OrderDetails
WHERE ProductID = 6;

```

```
-- Atlasa produkta ar ID 11 bāzes cenu no tabulas "Products"
SELECT BasePrice
FROM Products
WHERE ProductID = 5;

-- Pievieno jaunu kolonnu
ALTER TABLE OrderDetails
ADD DiscountAmount NUMBER(10, 2);

-- Noņem kolonnu "DiscountAmount" no tabulas "OrderDetails"
ALTER TABLE OrderDetails
DROP COLUMN DiscountAmount;
```

Vienkārša vaicājumi

1. Atlasa ražotāju ID, nosaukumu un kontaktinformāciju no tabulas “Manufacturers”

```
SELECT ManufacturerID, ManufacturerName, ContactInfo
FROM Manufacturers;
```

MANUFACTURERID	MANUFACTURERNAME	CONTACTINFO
101	Elektronika Ltd.	info@elektronika.com
102	Kabeļu Risinājumi	contact@kabris.lv
103	Drošības Sistēmas	support@drosiba.lv
104	BalticTech Solutions	sales@baltictech.lv
105	ElektroSmart	info@elektrosmart.com
106	SafeWire Systems	contact@safewire.lv

2. Atlasa klientu ID, vārdu un reģistrācijas numuru no tabulas “Clients”, kuri pieder atlaižu grupai ar ID 2

```
SELECT ClientID, ClientName, RegistrationNumber
FROM Clients
WHERE DiscountGroupID = 2;
```

CLIENTID	CLIENTNAME	REGISTRATIONNUMBER
2	Walmarts SIA	23456789012
4	RISBEA SIA	09876543211
9	Prisma SIA	78901234567

3. Atlasa produktu nosaukumu, bāzes cenu un mērvienību no tabulas “Products” un sakārto rezultātus pēc cenas dilstošā secībā

```
SELECT ProductName, BasePrice, Unit
FROM Products
ORDER BY BasePrice DESC;
```

PRODUCTNAME	BASEPRICE	UNIT
Sadalne kārba 100x100mm	3.5	gb
PVC truba 16mm	2.25	m
Elektrības kabelis 3x2.5mm	1.5	m
Drošinātājs 16A	1.25	gb
Elektrības kabelis 3x1.5mm	.85	m
Vadu marķieris	.5	gb
Vadu savienojuma skava	.1	gb

4. Atlasa klientu vārdus no tabulas “Clients”, kuriem vārda garums (simbolu skaits) ir tieši 10

```
SELECT ClientName
FROM Clients
WHERE LENGTH(ClientName) = 10;
```

CLIENTNAME
RISBEA SIA
Maxima SIA
Elko Grupa
Prisma SIA

5. Aprēķina visu pasūtījumu kopējo summu

```
SELECT SUM(TotalAmount) AS KopējaSumma
FROM Orders;
```

KOPĒJASUMMA
1606.75

6. Izvada pasūtījumus, kas veikti 2024. gada novembrī

```
SELECT *
FROM Orders
WHERE OrderDate BETWEEN TO_DATE('2024-11-01', 'YYYY-MM-DD') AND TO_DATE('2024-11-30', 'YYYY-MM-DD');
```

ORDERID	CLIENTID	ORDERDATE	TOTALAMOUNT
1001	1	01-NOV-24	125
1002	1	02-NOV-24	75
1003	3	03-NOV-24	250
1004	5	04-NOV-24	90
1005	2	05-NOV-24	180
1006	6	06-NOV-24	220
1007	8	07-NOV-24	95.5
1008	9	08-NOV-24	150.75
1009	7	09-NOV-24	300
1010	10	10-NOV-24	120.5

7. Aprēķina kopējo cenu katrai pasūtījuma detaļai

```
SELECT
    OrderDetailID,
    OrderID,
    ProductID,
```

```

Quantity,
UnitPrice,
(Quantity * UnitPrice) AS KopējāCena
FROM OrderDetails;

```

ORDERDETAILID	ORDERID	PRODUCTID	QUANTITY	UNITPRICE	KOPĒJĀCENA
2001	1001	1	100	.85	85
2002	1001	3	5	3.5	17.5
2003	1002	2	50	1.2	60
2004	1003	4	100	.75	75
2005	1004	6	10	1.25	12.5
2006	1004	7	20	.5	10
2007	1005	2	100	1.2	120
2008	1005	5	50	.1	5
2014	1009	1	150	.85	127.5
2015	1009	3	10	3.5	35
2016	1009	5	75	.1	7.5
2017	1010	6	15	1.25	18.75
2018	1010	7	25	.5	12.5

Vidējas sarežģītības vaicājumi

1. Atlasa produktu nosaukumus, ražotāju nosaukumus un bāzes cenas, kur bāzes cena ir mazāka par 1.00

```
SELECT
    p.ProductName,
    m.ManufacturerName,
    p.BasePrice
FROM Products p
JOIN Manufacturers m ON p.ManufacturerID = m.ManufacturerID
WHERE p.BasePrice < 1.00;
```

PRODUCTNAME	MANUFACTURERNAME	BASEPRICE
Elektrības kabelis 3x1.5mm	Elektronika Ltd.	.85
Vadu savienojuma skava	Drošības Sistēmas	.1
Vadu marķieris	Drošības Sistēmas	.5

2. Atlasa produktus ar informāciju, kur ražotāja nosaukums satur vārdu “Elektronika”

```
SELECT ProductID, ProductName, BasePrice, Unit
FROM Products p
JOIN Manufacturers m ON p.ManufacturerID = m.ManufacturerID
WHERE m.ManufacturerName LIKE '%Elektronika%';
```

PRODUCTID	PRODUCTNAME	BASEPRICE	UNIT
1	Elektrības kabelis 3x1.5mm	.85	m
2	Elektrības kabelis 3x2.5mm	1.5	m

3. Atlasa klientus un viņu veikto pasūtījumu skaitu, sakārtotus pēc pasūtījumu skaita dilstošā secībā

```
SELECT c.ClientName, COUNT(o.OrderID) AS PasutijumuSkaitis
FROM Clients c
LEFT JOIN Orders o ON c.ClientID = o.ClientID
GROUP BY c.ClientName
ORDER BY PasutijumuSkaitis DESC;
```

CLIENTNAME	PASUTIJUMUSKAITS
Tesco SIA	2
Rimi Latvia	1
Kaprs SIA	1
Walmarts SIA	1
Elektro Serviss	1
Elko Grupa	1
Maxima SIA	1
Prisma SIA	1
LULA SIA	1
RISBEA SIA	0

4. Atlasa pasūtījumus, klientus un pasūtījuma datumu, kur pasūtījuma summa ir lielāka par 100, sakārtotus dilstošā secībā pēc summas

```
SELECT o.OrderID, c.ClientName, o.OrderDate, o.TotalAmount
FROM Orders o
JOIN Clients c ON o.ClientID = c.ClientID
```

```
WHERE o.TotalAmount > 100  
ORDER BY o.TotalAmount DESC;
```

ORDERID	CLIENTNAME	ORDERDATE	TOTALAMOUNT
1009	Elko Grupa	09-NOV-24	300
1003	Kaprs SIA	03-NOV-24	250
1006	Maxima SIA	06-NOV-24	220
1005	Walmarts SIA	05-NOV-24	180
1008	Prisma SIA	08-NOV-24	150.75
1001	Tesco SIA	01-NOV-24	125
1010	Elektro Serviss	10-NOV-24	120.5

5. Atlasa pasūtījumu detaļas ar produktu nosaukumiem, kas satur vārdu “kabelis”

```
SELECT o.OrderID, p.ProductName, od.Quantity  
FROM OrderDetails od  
JOIN Products p ON od.ProductID = p.ProductID  
JOIN Orders o ON od.OrderID = o.OrderID  
WHERE p.ProductName LIKE '%kabelis%';
```

ORDERID	PRODUCTNAME	QUANTITY
1001	Elektrības kabelis 3x1.5mm	100
1002	Elektrības kabelis 3x2.5mm	50
1005	Elektrības kabelis 3x2.5mm	100
1009	Elektrības kabelis 3x1.5mm	150

6. Atlasa klientu vārdus un atbilstošās atlaižu grupas

```
SELECT c.ClientName AS Klients, d.DiscountGroupName AS AtlaižuGrupa
FROM Clients c
JOIN DiscountGroups d
ON c.DiscountGroupID = d.DiscountGroupID;
```

KLIENTS	ATLAIŽUGRUPA
Tesco SIA	Klients C
Walmarts SIA	Klients B
Kaprs SIA	Klients A
RISBEA SIA	Klients B
LULA SIA	Klients C
Maxima SIA	Klients D
Elko Grupa	Klients E
Rimi Latvia	Klients A
Prisma SIA	Klients B
Elektro Serviss	Klients E

7. Atlasa unikālos ražotāju nosaukumus no produktiem

```
SELECT DISTINCT m.ManufacturerName
FROM Manufacturers m
JOIN Products p ON m.ManufacturerID = p.ManufacturerID;
```

MANUFACTURERNAME
Drošības Sistēmas
Kabeļu Risinājumi
Elektronika Ltd.

Paaugstinātas sarežģītības vaicājumi

1. Rangs un blīvais rangs klientiem pēc pasūtījuma summas atlaižu grupās

```
SELECT ProductID, ProductName, ManufacturerID, BasePrice,
       RANK() OVER (PARTITION BY ManufacturerID ORDER BY BasePrice DESC) AS PriceRank -- Produktu
FROM
  Products
```

PRODUCTID	PRODUCTNAME	MANUFACTURERID	BASEPRICE	PRICERANK
2	Elektrības kabelis 3x2.5mm	101	1.2	1
1	Elektrības kabelis 3x1.5mm	101	.85	2
3	Sadalne kārba 100x100mm	102	3.5	1
4	PVC truba 16mm	102	.75	2
6	Drošinātājs 16A	103	1.25	1
7	Vadu marķieris	103	.5	2
5	Vadu savienojuma skava	103	.1	3

Darbības un paskaidrojumi

PARTITION BY ManufacturerID: Nodrošina, ka rangs tiek aprēķināts atsevišķi katram ražotājam.

ORDER BY BasePrice DESC: Nodrošina, ka augstāka cena iegūst zemāku rangu.

RANK(): Ja produktiem ir vienāda cena, tiem tiek piešķirts vienāds rangs, bet nākamais produkts saņem nākamo rangu.

2. Klienti ar pasūtījumu kopējo summu un vidējo summu

```
-- Izvēlas klienta ID, vārdu un vidējo pasūtījumu summu
SELECT
    c.ClientID,
    c.ClientName,
    AVG(o.TotalAmount) AS AverageOrder
FROM
    Clients c
JOIN
    Orders o ON c.ClientID = o.ClientID
GROUP BY
    c.ClientID, c.ClientName
HAVING
    AVG(o.TotalAmount) > 100;
```


CLIENTID	CLIENTNAME	AVERAGEORDER
9	Prisma SIA	150.75
10	Elektro Serviss	120.5
3	Kaprs SIA	250
2	Walmarts SIA	180
6	Maxima SIA	220
7	Elko Grupa	300

Darbības un paskaidrojumi

AVG (o.TotalAmount) AS AverageOrder: Rezultātā attēlos pasūtījuma vidējos apjomus.

HAVING AVG(o.TotalAmount) > 100 : Apakšnosacījums, lai parāda tikai no pasūtījumiem, kuriem apjoms virs 100 eiro

3. Klienti ar pasūtījumu kopējo summu un vidējo summu

```
SELECT
    ProductID,
    ProductName,
    BasePrice
FROM
    Products
WHERE
    BasePrice > (SELECT AVG(BasePrice) FROM Products); -- Filtrē produktus ar cenu virs vidē,
```

PRODUCTID	PRODUCTNAME	BASEPRICE
2	Elektrības kabelis 3x2.5mm	1.5
3	Sadalne kārba 100x100mm	3.5
4	PVC truba 16mm	2.25

Darbības un paskaidrojumi

BasePrice > (SELECT AVG(BasePrice) FROM Products) : Attēlo vienīgi tos produktus, kuri izpilda salīdzinājumu starp individuālā produkta cenu un apakšvaicājuma rezultātā iegūto visu produktu vidējo cenu

-
4. Klienti, kuru vidējā pasūtījuma summa ir augstāka par visiem klientiem kopumā

```

SELECT
    c.ClientID,
    c.ClientName,
    AVG(o.TotalAmount) AS AverageOrder -- Vidējā pasūtījuma summa
FROM
    Clients c
JOIN
    Orders o ON c.ClientID = o.ClientID
GROUP BY
    c.ClientID, c.ClientName
HAVING
    AVG(o.TotalAmount) > (
        -- Apakšvaicājums, kas aprēķina vidējo pasūtījumu summu visiem klientiem
        SELECT
            AVG(TotalAmount)
        FROM
            Orders
    );

```

CLIENTID	CLIENTNAME	AVERAGEORDER
3	Kaprs SIA	250
2	Walmarts SIA	180
6	Maxima SIA	220
7	Elko Grupa	300

Darbības un paskaidrojumi

GROUP BY c.ClientID, c.ClientName : Nepieciešams AVG funkciju izpildīšanai.

HAVING : Izveido papildus nosacījumus datu filtrēšanai

(*SELECT AVG(TotalAmount) FROM Orders*) : Apakšvaicājums salīdzināšanai

5. Produktu vērtības salīdzinājums ražotāju griezumā

```

SELECT
    ProductID,
    ManufacturerID,
    BasePrice,
    RANK() OVER (
        PARTITION BY ManufacturerID
        ORDER BY BasePrice DESC NULLS LAST
    ) AS Total_Value_Rank,
    DENSE_RANK() OVER (
        PARTITION BY ManufacturerID
        ORDER BY BasePrice DESC NULLS LAST
    ) AS Total_Value_Dense_Rank
FROM
    Products
WHERE

```

```

    ManufacturerID IN (101, 102)
UNION
SELECT
    ProductID,
    ManufacturerID,
    BasePrice,
    RANK() OVER (
        PARTITION BY ManufacturerID
        ORDER BY BasePrice DESC NULLS LAST
    ) AS Total_Value_Rank,
    DENSE_RANK() OVER (
        PARTITION BY ManufacturerID
        ORDER BY BasePrice DESC NULLS LAST
    ) AS Total_Value_Dense_Rank
FROM
    Products
WHERE
    ManufacturerID IN (103, 104)
    AND BasePrice > 1.00;

```

PRODUCTID	MANUFACTURERID	BASEPRICE	TOTAL_VALUE_RANK	TOTAL_VALUE_DENSE_RANK
1	101	.85	2	2
2	101	1.5	1	1
3	102	3.5	1	1
4	102	2.25	2	2
6	103	1.25	1	1

Darbības un paskaidrojumi

```

RANK() OVER (
    PARTITION BY ManufacturerID
    ORDER BY BasePrice DESC NULLS LAST
) AS Total_Value_Rank

```

RANK(): Piešķir rangu katrai vērtībai

OVER (): Definē window (datu loga) parametrus

PARTITION BY: Sadala datus grupās vadoties pēc ManufacturerID
ORDER BY: Nosaka ranžēšanas kārtību, pamatojoties uz BasePrice
NULLS LAST: Nulles vērtības novieto saraksta beigās

Ja mums ir produkti no ražotāja 101:

- Produkts A: Cena 3.50 → Rank 1
- Produkts B: Cena 3.50 → Rank 1
- Produkts C: Cena 2.00 → Rank 3

```
SELECT ... WHERE ManufacturerID IN (101, 102)
UNION
SELECT ... WHERE ManufacturerID IN (103, 104)
```

Apvieno divu SELECT rezultātus. Izņem dublētās rindas. Ļauj kombinēt dažādus meklēšanas nosacījumus