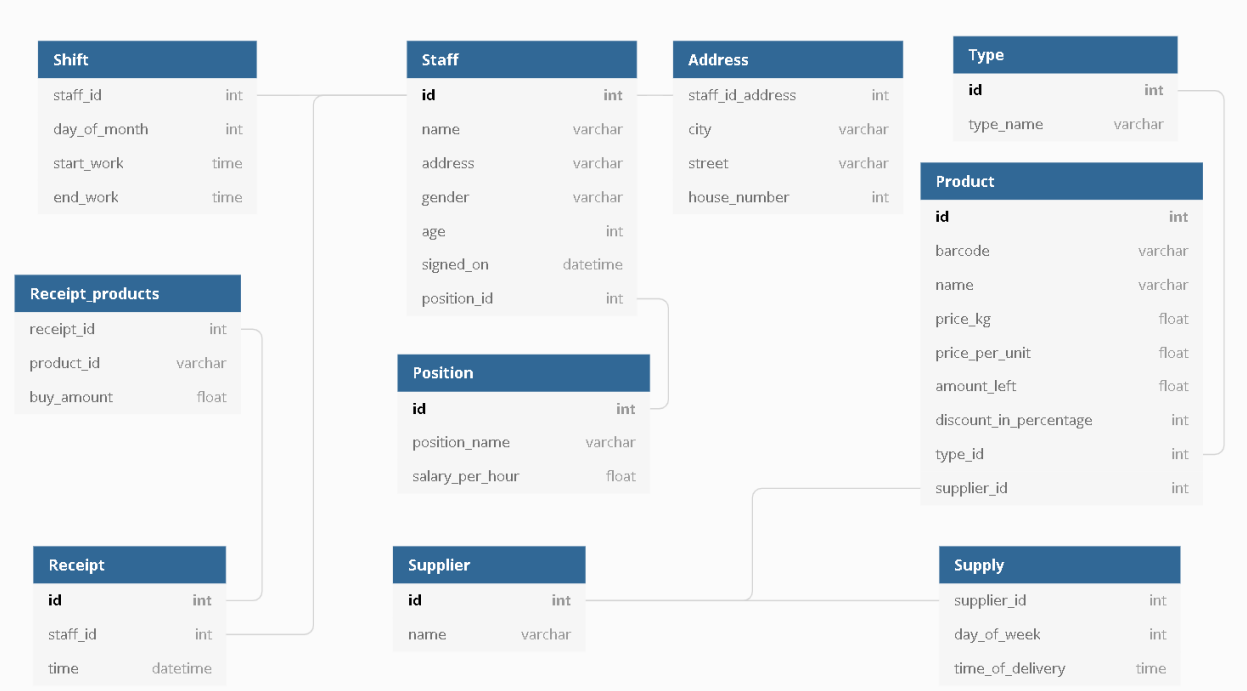
Магазин за хранителни стоки

Базата данни е предвидена да удовлетворява основни нужди на магазин за хранителни стоки, като бизнес логиката на системата не е изцяло в базата данни. Покрива информация за продуктите, които се предлагат в един хранителен магазин,като наличност и цена на продуктите, информация за неговите служители, като лична информация и информация за работното им време, информация за касови бележки, както и за доставчици на стока. По подробна информация може да се види в схемата на релациите, както и от наличните заявки.



Добавяне на примерно съдържание

Въвеждането на стойности е разделено по таблици както следва:

```
----Position----
INSERT INTO Position (id,position_name,salary_per_hour)
VALUES (1,'Manager',9);

INSERT INTO Position (id,position_name,salary_per_hour)
VALUES (2,'Cashier',6);

INSERT INTO Position (id,position_name,salary_per_hour)
VALUES (3,'Hygienist',5);

INSERT INTO Position (id,position_name,salary_per_hour)
VALUES (4,'Accountant',9);

INSERT INTO Position (id,position_name,salary_per_hour)
VALUES (5,'HR',7.5);
```

```sql
----Staff----
INSERT INTO Staff (id,name,phone_number,gender,age,signed_on,position_id)
VALUES (1,'Stanislav Dimov','0889047433','M','20','2018-07-07',2);

INSERT INTO Staff (id,name,phone_number,gender,age,signed_on,position_id)
VALUES (2,'Georgi Kovachev','0889047436','M','21','2019-01-14',2);

INSERT INTO Staff (id,name,phone_number,gender,age,signed_on,position_id)
VALUES (3,'Georgi Katsarski','0889327433','M','20','2019-10-28',2);

INSERT INTO Staff (id,name,phone_number,gender,age,signed_on,position_id)
VALUES (4,'Petar Petrov','0889047444','M','40','2015-02-15',1);

INSERT INTO Staff (id,name,phone_number,gender,age,signed_on,position_id)
VALUES (5,'Mincho Minchev','0889047343','M','60','2018-12-03',3);

INSERT INTO Staff (id,name,phone_number,gender,age,signed_on,position_id)
VALUES (6,'Cristiano Ronaldo','0889047458','M','35','2020-03-14',4);

INSERT INTO Staff (id,name,phone_number,gender,age,signed_on,position_id)
VALUES (7,'Georgi Iliev','0889247433','M','29','2017-05-12',5);

INSERT INTO Staff (id,name,phone_number,gender,age,signed_on,position_id)
VALUES (8,'Vasil Iliev','0889947433','M','30','2017-11-10',3);

INSERT INTO Staff (id,name,phone_number,gender,age,signed_on,position_id)
VALUES (9,'Ginka Varbakova','0880147433','F','40','2016-04-05',4);

----Address----
INSERT INTO Address (staff_id_address,city,street,house_number)
VALUES (1,'Varna','Ul. Tsar Boris',12);

INSERT INTO Address (staff_id_address,city,street,house_number)
VALUES (2,'Varna','Ul. Perperikon',8);

INSERT INTO Address (staff_id_address,city,street,house_number)
VALUES (3,'Varna','Bul. Tsar Osvoboditel',154);

INSERT INTO Address (staff_id_address,city,street,house_number)
VALUES (4,'Varna','Bul. Slivnica',14);

INSERT INTO Address (staff_id_address,city,street,house_number)
VALUES (5,'Varna','Bul. Tsar Osvoboditel',54);

INSERT INTO Address (staff_id_address,city,street,house_number)
VALUES (6,'Varna','Bul. 3ti mart',40);

INSERT INTO Address (staff_id_address,city,street,house_number)
VALUES (7,'Varna','Bul. Tsar Osvoboditel',100);

INSERT INTO Address (staff_id_address,city,street,house_number)
VALUES (8,'Varna','Ul. Prespa',1);

----Shift----
INSERT INTO Shift (staff_id,day_of_month,start_work,end_work)
VALUES (1,1,'08:00:00','16:00:00');
```

```sql
INSERT INTO Shift (staff_id,day_of_month,start_work,end_work)
VALUES (2,1,'16:00:00','22:00:00');

INSERT INTO Shift (staff_id,day_of_month,start_work,end_work)
VALUES (3,2,'08:00:00','16:00:00');

INSERT INTO Shift (staff_id,day_of_month,start_work,end_work)
VALUES (1,2,'16:00:00','22:00:00');

INSERT INTO Shift (staff_id,day_of_month,start_work,end_work)
VALUES (5,5,'08:00:00','12:00:00');

INSERT INTO Shift (staff_id,day_of_month,start_work,end_work)
VALUES (5,10,'08:00:00','12:00:00');

INSERT INTO Shift (staff_id,day_of_month,start_work,end_work)
VALUES (4,4,'10:00:00','18:00:00');

INSERT INTO Shift (staff_id,day_of_month,start_work,end_work)
VALUES (4,5,'10:00:00','16:00:00');

INSERT INTO Shift (staff_id,day_of_month,start_work,end_work)
VALUES (6,8,'08:00:00','16:00:00');

INSERT INTO Shift (staff_id,day_of_month,start_work,end_work)
VALUES (7,8,'08:00:00','16:00:00');

INSERT INTO Shift (staff_id,day_of_month,start_work,end_work)
VALUES (8,10,'08:00:00','16:00:00');

INSERT INTO Shift (staff_id,day_of_month,start_work,end_work)
VALUES (9,11,'08:00:00','16:00:00');

INSERT INTO Shift (staff_id,day_of_month,start_work,end_work)

VALUES (1,1,'08:00:00','16:00:00');

  ----Type----
INSERT INTO Type (id, name)
VALUES
(1, 'Bakery products'),
(2, 'Sweets'),
(3, 'Fruits'),
(4, 'Vegetables'),
(5, 'Soft drinks'),
(6, 'Alcohol'),
(7, 'Basic food'),
(8, 'Milk products'),
(9, 'Other');

----Supplier----
 INSERT INTO Supplier(id, name)
 VALUES
 (10, 'Vita'),
 (11, 'Dobruja'),
 (12, 'Makao'),
 (13, 'Aro'),
 (14, 'Svoge'),
```

```
 (15, 'Nestle'),
 (16, 'Naya'),
 (17, 'Pepsi'),
 (18, 'Coca Cola'),
 (19, 'Devin'),
 (20, 'Savoy'),
 (21, 'Krina'),
 (22, 'Sofia Mel'),
 (23, 'Kristal'),
 (24, 'Elena'),
 (25, 'Alfatar'),
 (26, 'Vedrare'),
 (27, 'Oranjeria');

----Product----
INSERT INTO Product (id, barcode, name, price_per_kg, price_per_unit, amount_left,
type_id, supplier_id)
 VALUES
 (1000, 81932, 'Vita Bread', 4, 2, 30, 1, 10),
 (1001, 81930, 'Dobruja Bread', 2, 1.50, 15, 1, 11),
 (1002, 81929, 'Makao Bread', 1.50, 1.10, 3, 1, 12),
 (1003, 81915, 'Croissant with butter Makao', 3, 1.50, 18, 1, 12),
 (1004, 81923, 'Arabic bread', 4, 1.50, 30, 1, 13),
 (1005, 82930, 'Svoge Chocolate', 2, 1.50, 20, 2, 14),
 (1006, 82932, 'Bounty dessert', 29, 1.15, 26, 2, 15),
 (1007, 82934, 'Twix dessert', 30, 1.15, 32, 2, 15),
 (1008, 82933, 'Waffles Naya', 10, 2.50, 30, 2, 16),
 (1009, 82936, 'Waffles Makao', 7, 1.50, 30, 2, 12),
 (1010, 83930, 'Pepsi', 2, 1, 30, 5, 17),
 (1011, 83932, 'Coca Cola', 2, 1, 30, 5, 18),
 (1012, 83934, 'Fanta Baobab', 2, 1, 30,5 , 18),
 (1013, 83936, 'Water', 1, 0.50, 30, 5,19 ),
 (1014, 83937, 'Soda', 1, 0.50, 30,5, 19),
 (1015, 81938, 'Vodka Savoy', 10, 10, 30,6, 20),
 (1016, 81939, 'Rum Savoy', 10, 10, 30, 6, 20),
 (1017, 81940, 'Whiskey Savoy', 10, 10, 30, 6, 20),
 (1018, 81941, 'Tequila Savoy', 10, 10, 30, 6 , 20),
 (1019, 81942, 'Gin Savoy', 10, 10, 30, 6, 20),
 (1020, 81943, 'Beans', 2, 2, 30, 7, 21),
 (1021, 81944, 'Lentils', 1.80, 1.80, 30, 7, 21),
 (1022, 81945, 'Rice', 1.50, 1.50, 30, 7, 21),
 (1023, 81946, 'Flour', 1, 1, 30, 7, 22),
 (1024, 81947, 'Salt', 0.90, 0.90, 30, 7, 23),
 (1025, 81948, 'Yoghurt', 2, 1.15, 30, 8, 24),
 (1026, 81949, 'Milk', 2, 2, 30, 8, 25),
 (1027, 81950, 'Cheese', 10, 5, 30, 8, 24),
 (1028, 81951, 'Yellow Cheese', 14, 7, 30, 8, 24),
 (1029, 81952, 'Curd', 2, 1.5, 30, 8, 26);
 INSERT INTO Product  (id, barcode, name, price_per_kg, amount_left, type_id,
supplier_id) /* no price per unit, amount in kg*/
 VALUES
 (1030,81953, 'Apples', 1.20, 20, 3, 27),
 (1031,81954,'Pear', 3.50, 20, 3, 27),
 (1032,81955, 'Orange', 2, 20, 3, 27),
 (1033,81956, 'Banana', 2.60, 20, 3, 27),
 (1034,81957, 'Strawberry', 5, 20, 3, 27),
 (1035,81959, 'Tomato', 3.50, 20, 4, 27),
 (1036,81962, 'Cucumber', 2.90, 4, 4, 27),
```

```
(1037,81960, 'Potato', 1.10, 30, 4, 27),
(1038,81961, 'Carrot', 0.80, 15, 4, 27),
(1039,81963, 'Cabbage', 2, 20, 3, 27);



----Supply---
INSERT INTO Supply (supplier_id,day_of_week,time_of_delivery)
VALUES (10,1,'10:30:00');

INSERT INTO Supply (supplier_id,day_of_week,time_of_delivery)
VALUES (11,1,'11:00:00');
INSERT INTO Supply (supplier_id,day_of_week,time_of_delivery)
VALUES (12,1,'11:30:00');
INSERT INTO Supply (supplier_id,day_of_week,time_of_delivery)
VALUES (13,1,'12:00:00');
INSERT INTO Supply (supplier_id,day_of_week,time_of_delivery)
VALUES (14,2,'09:00:00');
INSERT INTO Supply (supplier_id,day_of_week,time_of_delivery)
VALUES (15,3,'09:30:00');
INSERT INTO Supply (supplier_id,day_of_week,time_of_delivery)
VALUES (16,3,'10:30:00');
INSERT INTO Supply (supplier_id,day_of_week,time_of_delivery)
VALUES (17,3,'10:30:00');
INSERT INTO Supply (supplier_id,day_of_week,time_of_delivery)
VALUES (18,4,'10:00:00');
INSERT INTO Supply (supplier_id,day_of_week,time_of_delivery)
VALUES (11,4,'11:00:00');
INSERT INTO Supply (supplier_id,day_of_week,time_of_delivery)
VALUES (19,5,'12:00:00');
INSERT INTO Supply (supplier_id,day_of_week,time_of_delivery)
VALUES (20,5,'12:00:00');

----Receipt----
INSERT INTO Receipt(id,staff_id)
VALUES (1,1);

INSERT INTO Receipt(id,staff_id)
VALUES (2,1);

INSERT INTO Receipt(id,staff_id)
VALUES (3,2);

INSERT INTO Receipt(id,staff_id)
VALUES (4,2);

INSERT INTO Receipt(id,staff_id)
VALUES (5,3);

INSERT INTO Receipt(id,staff_id)
VALUES (6,3);

----Receipt_products
INSERT INTO Receipt_products(receipt_id,product_id,buy_amount)
VALUES (1,1001,3)

INSERT INTO Receipt_products(receipt_id,product_id,buy_amount)
VALUES (1,1002,1)
```

```sql
INSERT INTO Receipt_products(receipt_id,product_id,buy_amount)
VALUES (1,1004,2)

INSERT INTO Receipt_products(receipt_id,product_id,buy_amount)
VALUES (2,1005,3)

INSERT INTO Receipt_products(receipt_id,product_id,buy_amount)
VALUES (2,1007,3)

INSERT INTO Receipt_products(receipt_id,product_id,buy_amount)
VALUES (2,1009,2)

INSERT INTO Receipt_products(receipt_id,product_id,buy_amount)
VALUES (3,1011,5)

INSERT INTO Receipt_products(receipt_id,product_id,buy_amount)
VALUES (4,1020,3)

INSERT INTO Receipt_products(receipt_id,product_id,buy_amount)
VALUES (5,1015,1)

INSERT INTO Receipt_products(receipt_id,product_id,buy_amount)
VALUES (5,1016,2)

INSERT INTO Receipt_products(receipt_id,product_id,buy_amount)
VALUES (6,1005,3)
```

Примерни заявки

```sql
----Прости Заявки----

--Извежда имената на служители, които са сключили договор след определена
дата(2019.10.10)--
SELECT name
FROM Staff
WHERE signed_on < '2019.10.10'
```

name
Stanislav Dimov
Georgi Kovachev
Petar Petrov
Mincho Minchev
Georgi Iliev
Vasil Iliev
Ginka Varbakova

```
--Извежда имената на служители, които са под определена възраст(25)--
SELECT name
FROM staff
WHERE age < 25
```

name
Stanislav Dimov
Georgi Kovachev
Georgi Katsarski

```
--Извежда позициите, чиито заплата са над определен брой левове(6) на час--
SELECT position_name
FROM Position
WHERE salary_per_hour > 6
```

position_name
Manager
Accountant
HR

```
--Извежда имената на всички продукти, чиято цена е под дадена сума(1) за бройка--
SELECT name
FROM Product
WHERE price_per_unit < 1
```

name
Water
Soda
Salt

```
--Извежда имената на продукти, чиято цена е намалена--
SELECT name
FROM Product
WHERE discount_in_percentage IS NOT NULL


----Заявки върху две и повече релации----

--Извежда имената на всички продукти,чиято нормална цена е под дадена сума(2лв) и също са
намалени--
(SELECT name
FROM Product
WHERE discount_in_percentage IS NOT NULL)
INTERSECT
(SELECT name
FROM Product
WHERE price_per_unit<2)

--Извежда имената на всички служители, които са сключили договор след определена дата и
са под определена възраст--
(SELECT name
FROM Staff
WHERE signed_on < '2019.10.10')
INTERSECT
(SELECT name
FROM staff
WHERE age < 25)
```

name

Georgi Kovachev
Stanislav Dimov

```
--Извежда имената на всички служители в даден възрастов диапазон--
(SELECT name,age
FROM Staff
WHERE age > 21)
INTERSECT
(SELECT name,age
FROM Staff
WHERE age < 30)
```

name	age
Georgi Iliev	29

```
--Извежда имената на всички продукти, чиято наличност е над даден брой и цената за брой е
под дадена сума--
(SELECT name
FROM Product
WHERE amount_left > 5 AND price_per_unit IS NOT NULL)
INTERSECT
(SELECT name
FROM Product
WHERE price_per_unit<2)
```

name
Dobruja Bread
Croissant with butter Makao
Arabic bread
Svoge Chocolate
Bounty dessert
Twix dessert
Waffles Makao
Pepsi
Coca Cola
Fanta Baobab
Water
Soda
Lentils
Rice
Flour
Salt
Yoghurt
Curd

```
--          Извежда имената на всички продукти от даден тип,
            зададен по името на типа, и наличност под даден
брой-       -
            (SELECT Product.name
FROM        Product
JOIN        Type ON type_id=Type.id
WHERE       Type.name='Sweets')
            INTERSECT
            (SELECT name
FROM        Product
WHERE       amount_left<5)
```

name
Svoge Chocolate

```
----Подзаявки----

--Извежда позицията с най-висока заплата за час--
SELECT position_name
FROM Position
WHERE salary_per_hour >= ALL (SELECT salary_per_hour
                              FROM Position)
```

position_name
Manager
Accountant

```sql
--Извежда имената на всички служители на дадена позиция--
SELECT name
FROM Staff
WHERE position_id IN(SELECT id
                     FROM Position
                     WHERE position_name='Cashier')
```

name
Stanislav Dimov
Georgi Kovachev
Georgi Katsarski

```sql
--Извежда имената на продукти, които се доставят в определен ден--
SELECT name
FROM Product
WHERE supplier_id IN (SELECT supplier_id
                      FROM Supply
                      WHERE day_of_week=1)
```

name
Vita Bread
Dobruja Bread
Makao Bread
Croissant with butter Makao
Arabic bread
Waffles Makao

```sql
--Извежда имената на доставчиците, които доставят в определен ден--
SELECT name
FROM Supplier
WHERE id IN (SELECT supplier_id
             FROM Supply
             WHERE day_of_week=1)
```

name
Vita
Dobruja
Makao
Aro

```sql
--Извежда имената на служителите, които са на работа в определен ден--
SELECT name
FROM Staff
WHERE id IN(SELECT staff_id
            FROM Shift
            WHERE day_of_month=1)
```

name
Stanislav Dimov
Georgi Kovachev

```sql
----Съединения----

--Извежда информация за доставчиците(име,ден за доставки,час на доставка)
SELECT name,day_of_week, FORMAT(CAST(time_of_delivery AS DATETIME),'HH:mm') as
time_of_delivery
FROM Supplier
INNER JOIN Supply ON id=supplier_id
```

name	day_of_week	time_of_delivery
Vita	1	10:30
Dobruja	1	11:00

Makao	1	11:30
Aro	1	12:00
Svoge	2	09:00
Nestle	3	09:30
Naya	3	10:30
Pepsi	3	10:30
Coca Cola	4	10:00
Dobruja	4	11:00
Devin	5	12:00
Savoy	5	12:00

```
--Извежда информация за служителите(име,пол,възраст,адрес)--
SELECT name,gender,age,city,street,house_number
FROM Staff
INNER JOIN Address ON id=staff_id_address
```

name	gender	age	city	street	house_number
Stanislav Dimov	M	20	Varna	Ul. Tsar Boris	12
Georgi Kovachev	M	21	Varna	Ul. Perperikon	8
Georgi Katsarski	M	20	Varna	Bul. Tsar Osvoboditel	154
Petar Petrov	M	40	Varna	Bul. Slivnica	14
Mincho Minchev	M	60	Varna	Bul. Tsar Osvoboditel	54
Cristiano Ronaldo	M	35	Varna	Bul. 3ti mart	40
Georgi Iliev	M	29	Varna	Bul. Tsar Osvoboditel	100
Georgi Iliev	M	29	Varna	Ul. Prespa	1
Vasil Iliev	M	30	Varna	Ul. Prespa	1
Ginka Varbakova	**F**	40	NULL	NULL	NULL

```
--Извежда информация за продукти(име,тип на продукта)--
SELECT Product.name, Type.name
FROM Product
LEFT JOIN Type ON type_id=Type.id
```

name	name
Vita Bread	Bakery products
Dobruja Bread	Bakery products
Makao Bread	Bakery products
Croissant with butter Makao	Bakery products
Arabic bread	Bakery products
Svoge Chocolate	Sweets
Bounty dessert	Sweets
Twix dessert	Sweets
Waffles Naya	Sweets
Waffles Makao	Sweets
Pepsi	Soft drinks
Coca Cola	Soft drinks
Fanta Baobab	Soft drinks
Water	Soft drinks
Soda	Soft drinks
Vodka Savoy	Alcohol
Rum Savoy	Alcohol
Whiskey Savoy	Alcohol
Tequila Savoy	Alcohol
Gin Savoy	Alcohol
Beans	Basic food
Lentils	Basic food
Rice	Basic food
Flour	Basic food

name	position_name	salary_per_hour
Cristiano Ronaldo	Accountant	9
Ginka Varbakova	Accountant	9

Salt	Basic food
Yoghurt	Milk products
Milk	Milk products
Cheese	Milk products
Yellow Cheese	Milk products
Curd	Milk products
Apples	Fruits
Pear	Fruits
Orange	Fruits
Banana	Fruits
Strawberry	Fruits
Tomato	Vegetables
Cucumber	Vegetables
Potato	Vegetables
Carrot	Vegetables
Cabbage	Vegetables

```sql
--Извежда информация за заплата на час за служители с определена позиция--
SELECT name, position_name, salary_per_hour
FROM Staff
JOIN Position ON position_id=Position.id
WHERE position_id IN(SELECT id
                     FROM Position
                     WHERE position_name='Accountant')
--Извежда графика за работа на определен служител по име--
SELECT day_of_month,FORMAT(CAST(start_work AS DATETIME),'HH:mm') as
start_work,FORMAT(CAST(end_work AS DATETIME),'HH:mm') as
end_work,datediff(HOUR,start_work,end_work) as work_time
FROM Shift
WHERE staff_id IN (SELECT id
FROM Staff
WHERE name='Petar Petrov')
```

day_of_month	start_work	end_work	work_time
4	10:00	18:00	8
5	10:00	16:00	6

```sql
----Групиране и агрегация----

--Извежда информация за цената на всеки касов бон--
SELECT receipt_id,SUM(price_per_unit*buy_amount) as price
FROM Receipt_products
JOIN Product ON product_id=id
GROUP BY receipt_id
```

receipt_id	price
1	8.6
2	10.95
3	5
4	6
5	30
6	4.5

```sql
--Извежда информация за месечното работно време на служителите в часове--
SELECT staff_id, SUM(datediff(HOUR,start_work,end_work)) as month_work_time
FROM Shift
JOIN Staff ON staff_id=id
```

```
GROUP BY staff_id
```

staff_id	month_work_time
1	22
2	6
3	8
4	14
5	8
6	8
7	8
8	8
9	8

```
 --Извежда информация, колко доставки се очакват в определен ден от седмицата--
SELECT day_of_week, COUNT(*) as number_of_supplies
FROM Supply
GROUP BY day_of_week
```

day_of_week	number_of_supplies
1	4
2	1
3	3
4	2
5	2

```
--Извежда броя служители, работещи на дадена позиция--
SELECT COUNT(*) as number_of_cashiers
FROM Staff
WHERE position_id IN (SELECT id
                      FROM Position
                      WHERE position_name='Cashier')
```

number_of_cashiers
3

```
--Извежда в намаляващ ред най-купуваните продукти--
SELECT product_id,SUM(buy_amount) as total_amount
FROM Receipt_products
GROUP BY product_id
ORDER BY total_amount DESC
```

product_id	total_amount
1005	6
1011	5
1007	3
1001	3
1020	3
1009	2
1004	2
1016	2
1015	1
1002	1

```
--Извежда брой продукти за всички типове--
SELECT Type.name,COUNT(*) number_of_product_by_type
FROM Type
JOIN Product ON Type.id=type_id
GROUP BY Type.name
```

name	number_of_product_by_type

Alcohol	5
Bakery products	5
Basic food	5
Fruits	6
Milk products	5
Soft drinks	5
Sweets	5
Vegetables	4

```
--Извежда заплатата, която служител трябва да получи спрямо позицията си и месечните си
работни часове--
SELECT name, CASE WHEN month_work_time <= 40 THEN salary_per_hour*month_work_time
                             ELSE salary_per_hour*month_work_time +
salary_per_hour*0.5*(month_work_time-40)
                             END AS monthly_salary
FROM Staff
JOIN Position ON Staff.id=Position.id
JOIN(SELECT staff_id, SUM(datediff(HOUR,start_work,end_work)) as month_work_time
        FROM Shift
        GROUP BY staff_id) as Time ON staff_id=Staff.id
```

name	monthly_salary
Stanislav Dimov	207
Georgi Kovachev	42
Georgi Katsarski	45
Petar Petrov	135
Mincho Minchev	67.5
Cristiano Ronaldo	NULL
Georgi Iliev	NULL
Vasil Iliev	NULL
Ginka Varbakova	NULL

Заключения

Ограничения на ниво таблица:

```
ALTER Table Position ADD CONSTRAINT pk_position PRIMARY KEY(id);
ALTER Table Position ADD CONSTRAINT minimal_pay CHECK (salary_per_hour > 3.66);

ALTER Table Staff ADD CONSTRAINT pk_staff PRIMARY KEY(id);
ALTER Table Staff ADD CONSTRAINT gender_M_F CHECK (gender='M' OR gender='F');
ALTER Table Staff ADD CONSTRAINT minimal_age CHECK (age>=14);
ALTER Table Staff ADD CONSTRAINT start_date CHECK(signed_on<=GETDATE());
ALTER Table Staff ADD CONSTRAINT df_signed_on DEFAULT GETDATE() FOR signed_on;
ALTER Table Staff ADD CONSTRAINT fk_position_id FOREIGN KEY (position_id) REFERENCES
Position(id);

ALTER Table Shift ADD CONSTRAINT fk_staff_id FOREIGN KEY (staff_id) REFERENCES Staff(id);
ALTER Table Shift ADD CONSTRAINT start_end_correct CHECK (start_work < end_work);

ALTER Table Supplier ADD CONSTRAINT pk_supplier PRIMARY KEY(id)
ALTER Table Supplier ADD CONSTRAINT unique_name_supplier UNIQUE (name);

ALTER Table Supply ADD CONSTRAINT day_correct CHECK (day_of_week >= 1 AND day_of_week
<=7);

ALTER Table Type ADD CONSTRAINT pk_type PRIMARY KEY (id);
ALTER Table Type ADD CONSTRAINT unique_name_type UNIQUE(name);

ALTER Table Product ADD CONSTRAINT pk_product PRIMARY KEY(id);
ALTER Table Product ADD CONSTRAINT unique_barcode UNIQUE(barcode);
ALTER Table Product ADD CONSTRAINT unique_name UNIQUE(name);
ALTER Table Product ADD CONSTRAINT df_price_per_kg DEFAULT NULL FOR price_per_kg;
ALTER Table Product ADD CONSTRAINT df_amount_left DEFAULT NULL FOR amount_left;
ALTER Table Product ADD CONSTRAINT df_discount DEFAULT NULL FOR discount_in_percentage;
ALTER Table Product ADD CONSTRAINT fk_type_id FOREIGN KEY (type_id) REFERENCES Type(id);
ALTER Table Product ADD CONSTRAINT fk_supplier_id FOREIGN KEY (supplier_id) REFERENCES
Supplier(id)

ALTER Table Receipt ADD CONSTRAINT pk_receipt PRIMARY KEY(id);
ALTER Table Receipt ADD CONSTRAINT fk_staff_id_receipt FOREIGN KEY  (staff_id) REFERENCES
Staff(id);
ALTER Table Receipt ADD CONSTRAINT df_time DEFAULT GETDATE() FOR time_of_receipt

ALTER Table Address ADD CONSTRAINT fk_staff_id_address FOREIGN KEY (staff_id_address)
REFERENCES Staff(id);
```

Индекси, тригъри, изгледи

Индекси:

Няма добавени индекси, поради факта, че по-големите ни таблици (Product, Receipt, Receipt_products, Shifts) вероятно ще бъдат често променяни, а наличието на индекси ще направи тези операции по-бавни. Таблиците,които нямат да бъдат променяни често(Staff, Suppliers) в случая имат малко количество елементи, което от своя страна обезмисля поставянето на индекс. При използването на базата данни от по-голям магазин с повече служители и доставчици ще бъде удачно използването на индекси за колони, които често се използват в заявките

Тригъри:

```
CREATE TRIGGER shift_insert
ON Shift
INSTEAD OF INSERT
AS
IF EXISTS ( SELECT * FROM Shift INNER JOIN  inserted i ON i.staff_id=Shift.staff_id AND
i.day_of_month=Shift.day_of_month)
BEGIN
RAISERROR('Duplicate Data',1,1)
ROLLBACK
END
GO

CREATE TRIGGER receipt_delete
ON Receipt
INSTEAD OF DELETE
AS
BEGIN
RAISERROR('Deleting data is not allowed',1,2)
ROLLBACK
END
GO
```

Изгледи:

Употребата на изглед би била полезна в случай на изтриване на дадена таблица.Тъй като нашата база данни е на базово ниво, не се очаква да бъдат изтривани таблици на този етап. Друг случай на употреба би било честото използване на SELECT заявка. Можем да създадем изглед за съответната заявка. В нашата база данни често ни се налага да използваме информация от две таблици, затова можем да създадем изгледи както следва:

```
CREATE VIEW [Staff_Position] AS
SELECT Staff.name, Position.position_name
FROM Staff
JOIN Position ON position_id=Position.id


CREATE VIEW [Product_Type] AS
SELECT Product.name as product_name, Type.name as type_name
FROM Product
JOIN Type ON Product.type_id=Type.id
```