

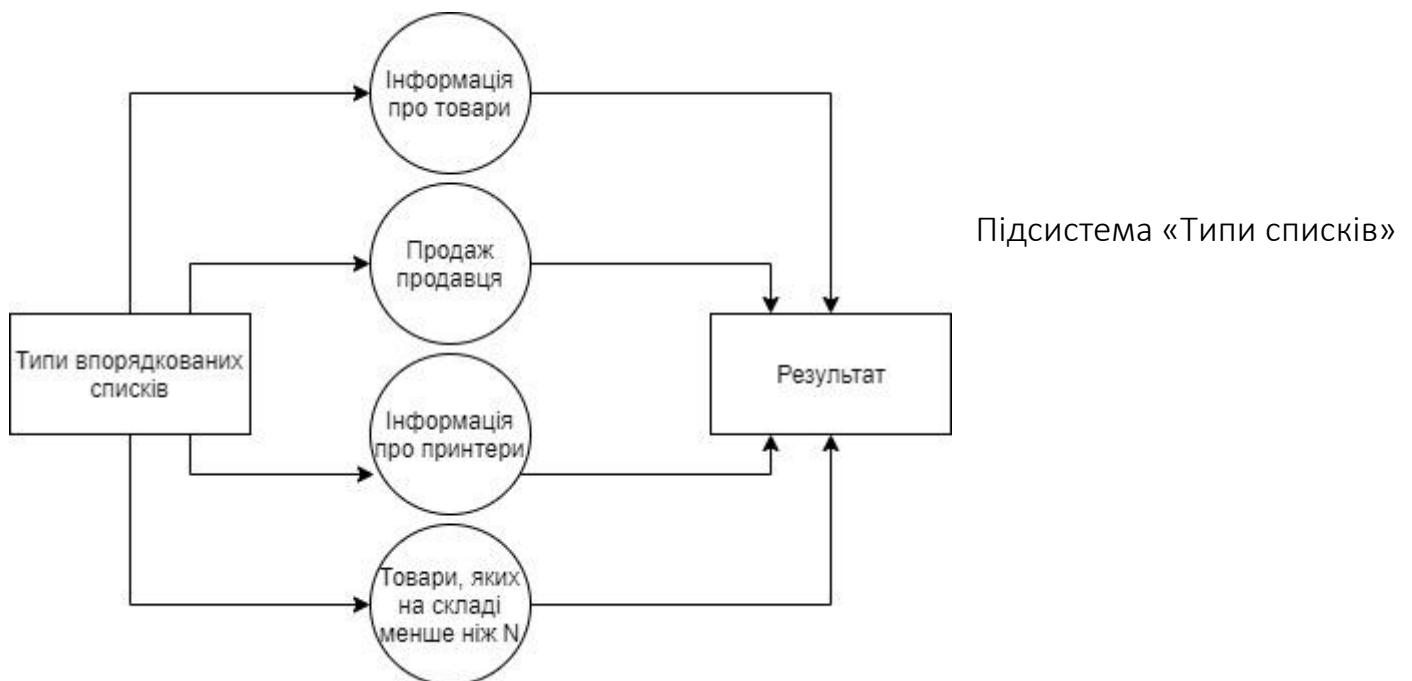
Предметна область - відділ комп'ютерної техніки в магазині.

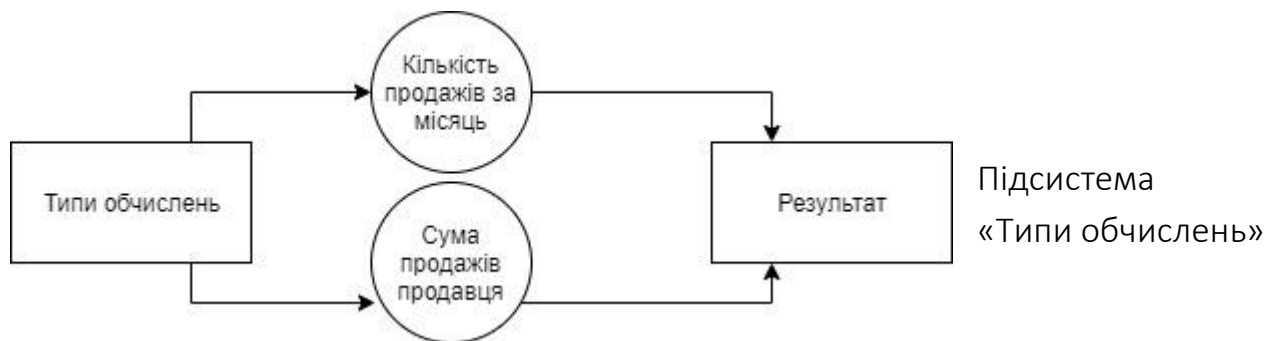
Задачі, що вирішує інформаційна система: видача відомостей про товари (опис, ціна в грн. та у.о., кількість на складі, марка і т.п.) і про їх реалізації (марка товару, продавець, дата продажу)..

Завдання на обробку даних:

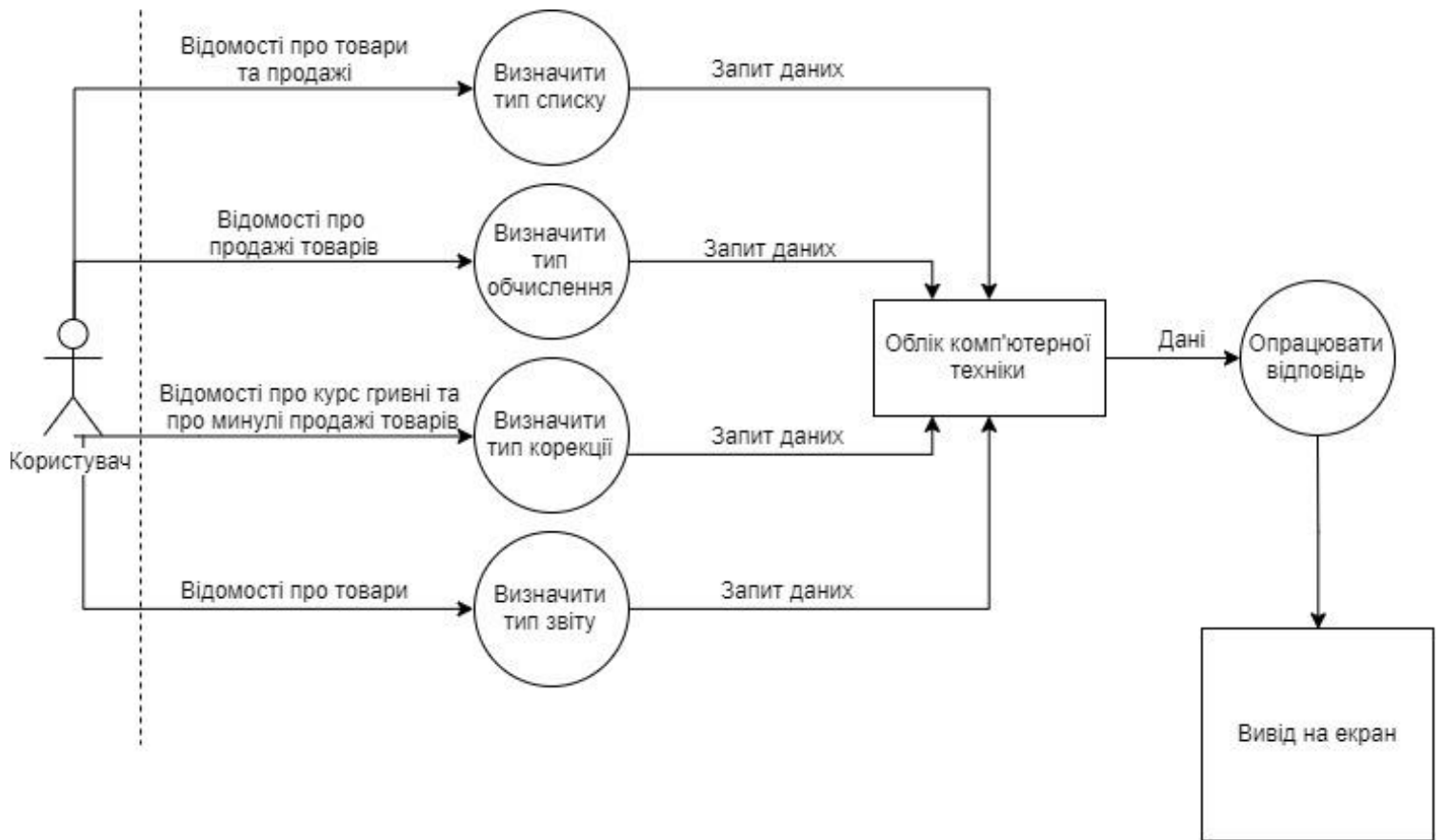
1. Створити впорядковані списки:
 - Інформація про товари: тип товару і кількість на складі; тип товару і ціна;
 - Продажу продавця УУ в день ХХ;
 - Інформація про принтери за ціною не більше Y;
 - Товари, яких на складі залишилося менше N штук
2. обчислення:
 - Кількість продажів за місяць;
 - Сума продажів продавця УУ
3. Корекція:
 - видалення відомостей про продажі за минулий рік;
 - зміна ціни в гривнях у зв'язку із зміною курсу гривні;
4. Звіти виду:
 - Характеристики комп'ютера зазначеної марки;
 - Інформація про товари; угруповання за типами (окремо - принтери, комп'ютери, монітори і т.п.);
 - Рекламний ярлик до товару (назва, характеристики, ціна)

Створення діаграми потоків даних (DFD-моделі)





Головна діаграма потоків даних



Проектування бази даних

```

CREATE TABLE JOBS(
  ID INT PRIMARY KEY IDENTITY,
  TITLE VARCHAR(30) NOT NULL)
  
```

```

CREATE TABLE SELLERS(
  ID INT PRIMARY KEY IDENTITY,
  FIRSTNAME VARCHAR(30) NOT NULL,
  LASTNAME VARCHAR(30) NOT NULL,
  ID_JOB INT NOT NULL
  CONSTRAINT JOB_KEY FOREIGN KEY (ID_JOB) REFERENCES JOBS(ID))
  
```

```

CREATE TABLE MARKS(
  ID INT PRIMARY KEY IDENTITY,
  TITLE VARCHAR(30) NOT NULL)
  
```

```

CREATE TABLE ITEM_TYPES(
  ID INT PRIMARY KEY IDENTITY,
  TITLE VARCHAR(30) NOT NULL)
  
```

```

CREATE TABLE CURRENCY(
  ID INT PRIMARY KEY IDENTITY,
  VALUTA VARCHAR(30) NOT NULL,
  
```

PRICE DECIMAL(10,2) NOT NULL CHECK (PRICE > 0))

```
CREATE TABLE ITEMS(  
  ID INT PRIMARY KEY IDENTITY,  
  FEATURE VARCHAR(120) NOT NULL,  
  STOCK_AMOUNT INT NOT NULL CHECK(STOCK_AMOUNT >= 0),  
  ID_PRICE INT NOT NULL  
  CONSTRAINT PRICE_KEY FOREIGN KEY(ID_PRICE) REFERENCES CURRENCY(ID),  
  PRICE_YO DECIMAL(10,2) NOT NULL CHECK(PRICE_YO > 0),  
  ID_MARK INT NOT NULL  
  CONSTRAINT MARK_KEY FOREIGN KEY (ID_MARK) REFERENCES MARKS(ID),  
  ID_TYPE INT NOT NULL  
  CONSTRAINT TYPE_KEY FOREIGN KEY (ID_TYPE) REFERENCES ITEM_TYPES(ID))
```

```
CREATE TABLE SALES(  
  ID_SELLER INT NOT NULL  
  CONSTRAINT SELLER_KEY FOREIGN KEY (ID_SELLER) REFERENCES SELLERS(ID),  
  DATE_SALE DATE NOT NULL CHECK (DATE_SALE >= '2000' AND DATE_SALE <= GETDATE()),  
  ID_ITEM INT NOT NULL  
  CONSTRAINT ITEM_KEY FOREIGN KEY (ID_ITEM) REFERENCES ITEMS(ID),  
  AMOUNT INT NOT NULL CHECK (AMOUNT > 0))
```

```
INSERT INTO CURRENCY VALUES  
( 'DOLLAR', 23.8)
```

```
INSERT INTO ITEM_TYPES VALUES  
( 'COMPUTER'),  
( 'PRINTER'),  
( 'MONITOR')
```

```
INSERT INTO MARKS VALUES  
( 'ACER'),  
( 'ASUS'),  
( 'LENOVO'),  
( 'HP')
```

```
INSERT INTO JOBS VALUES  
( 'JUNIOR SELLER'),  
( 'SELLER'),  
( 'MAIN SELLER')
```

```
INSERT INTO SELLERS VALUES  
( 'JOHN', 'NEWMAN', 1),  
( 'MICHAEL', 'WHITE', 1),  
( 'MEGAN', 'FOX', 2),  
( 'MARK', 'ROBBIN', 3)
```

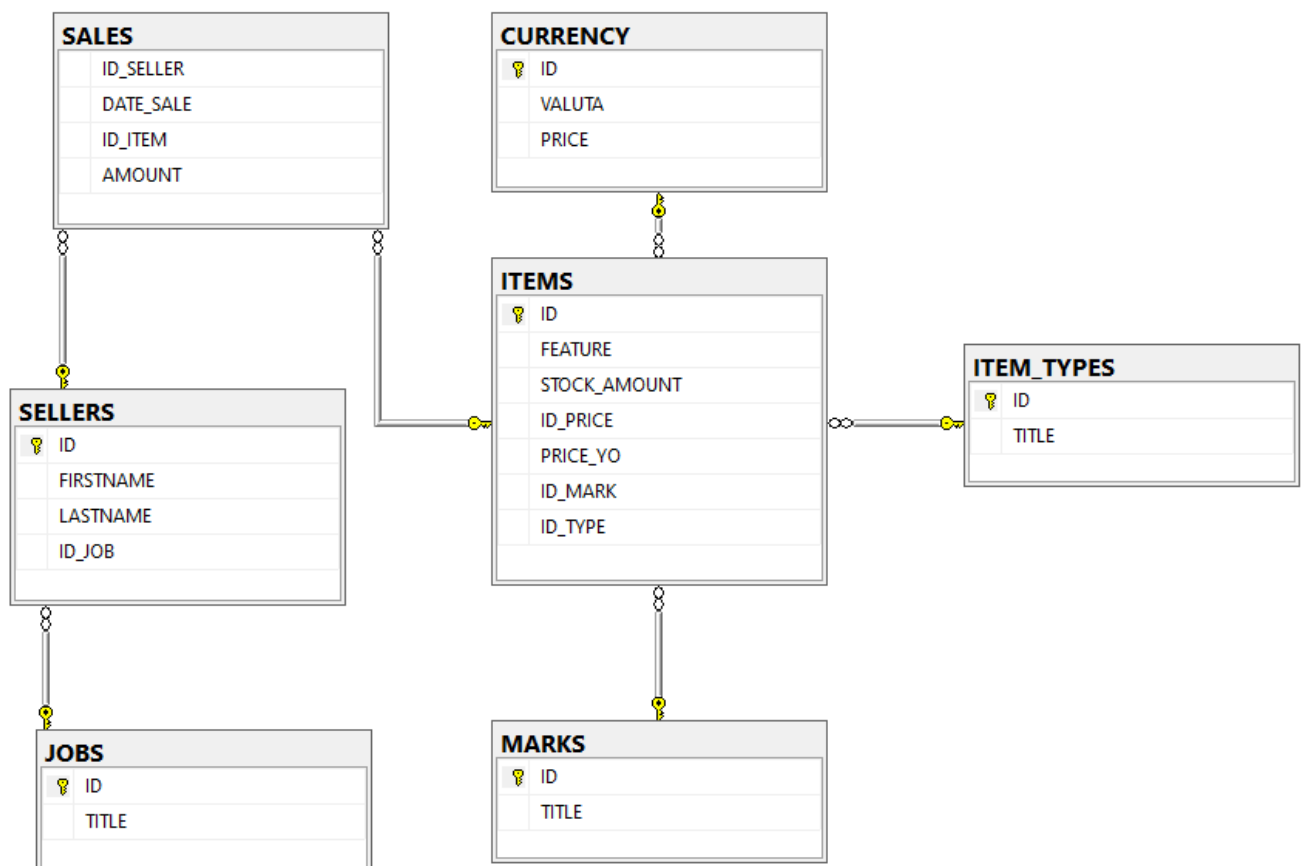
```
INSERT INTO ITEMS VALUES  
( 'WORK PC', 10, 1, 500, 1, 1),  
( 'GAME PC', 15, 1, 600, 1, 1),  
( 'PRO PC', 9, 1, 700, 2, 1),  
( 'BUDGET LOW PC', 1, 1, 400, 2, 1),  
( 'NETWORK PRINTER', 5, 1, 100, 2, 2),  
( 'COLORFUL PRINTER', 6, 1, 150, 3, 2),  
( 'BLACK&WHITE PRINTER', 4, 1, 90, 3, 2),
```

('LCD MONITOR',50,1,200,3,3),
('LED MONITOR',14,1,250,4,3),
('FULL HD MONITOR',2,1,300,4,3)

INSERT INTO SALES VALUES

(1,'2019-12-01',1,5),
(1,'2019-11-29',2,8),
(1,'2019-11-30',3,50),
(2,'2019-12-01',4,13),
(2,'2019-12-02',5,66),
(3,'2019-12-01',6,31),
(3,'2019-12-03',7,58),
(4,'2019-12-01',8,96),
(4,'2019-11-29',9,1),
(4,'2019-11-28',10,15)

ER-діаграма



Створення необхідного функціоналу

```
CREATE VIEW ITEMS_LIST
AS SELECT ITEMS.ID, ITEMS.FEATURE, ITEMS.STOCK_AMOUNT, CURRENCY.PRICE*PRICE_YO AS PRICE, MARKS.TITLE AS MARK,
ITEM_TYPES.TITLE AS TP FROM ITEMS
INNER JOIN CURRENCY ON ITEMS.ID_PRICE=CURRENCY.ID
INNER JOIN MARKS ON ITEMS.ID_MARK=MARKS.ID
INNER JOIN ITEM_TYPES ON ITEMS.ID_TYPE=ITEM_TYPES.ID
```

Впорядковані списки:

-- Інформація про товари: тип товару і кількість на складі; тип товару і ціна;

```
CREATE VIEW TYPE_AMOUNT
AS SELECT ID,STOCK_AMOUNT,MARK, TP FROM ITEMS_LIST
```

```
CREATE VIEW TYPE_PRICE
AS SELECT ID,PRICE,MARK, TP FROM ITEMS_LIST
```

- Продажу продавця УУ в день ХХ;

```
CREATE PROC SELLER_DAY
@SELLER VARCHAR(30),
@DATE DATE
AS
IF EXISTS(SELECT 'TRUE' FROM SALES
INNER JOIN SELLERS ON SALES.ID_SELLER=SELLERS.ID
WHERE SELLERS.LASTNAME = @SELLER AND DATE_SALE=@DATE)
BEGIN
SELECT SELLERS.FIRSTNAME, SELLERS.LASTNAME, AMOUNT FROM SALES
INNER JOIN SELLERS ON SALES.ID_SELLER=SELLERS.ID
WHERE SELLERS.LASTNAME = @SELLER AND DATE_SALE=@DATE
END
ELSE
BEGIN
PRINT 'DOES NOT EXIST'
END
```

- Інформація про принтери за ціною не більше Y;

```
CREATE PROC PRINTERS
@PRICE DECIMAL(10,2)
AS
IF EXISTS(SELECT 'TRUE' FROM ITEMS_LIST WHERE PRICE <=@PRICE AND TP='PRINTER')
BEGIN
SELECT FEATURE FROM ITEMS_LIST WHERE PRICE <=@PRICE AND TP='PRINTER'
END
ELSE
BEGIN
PRINT 'THERE ARE NO SUPPORTED PRINTERS'
END
```

- Товари, яких на складі залишилося менше N штук

```
CREATE PROC STOCK_LESS
@AMOUNT INT
AS
IF EXISTS(SELECT 'TRUE' FROM ITEMS_LIST WHERE STOCK_AMOUNT < @AMOUNT)
BEGIN
SELECT * FROM ITEMS_LIST WHERE STOCK_AMOUNT < @AMOUNT
END
ELSE
BEGIN
PRINT 'THERE ARE ALL PRODUCTS AMOUNT BIGGER THAN IT'
END
```

Обчислення:

- Кількість продажів за місяць;

```
CREATE PROC MONTH_SALE
@DATE DATE
AS
IF EXISTS(SELECT 'TRUE' FROM SALES WHERE MONTH(Date_Sale) = MONTH(@DATE) AND YEAR(Date_Sale) = YEAR(@DATE))
BEGIN
SELECT SUM(AMOUNT) FROM SALES
WHERE MONTH(Date_Sale) = MONTH(@DATE) AND YEAR(Date_Sale) = YEAR(@DATE)
END
ELSE
BEGIN
PRINT 'THERE ARE NO SALES IN THIS MONTH'
END
```

- Сума продажів продавця УУ

```
CREATE PROC SELLER_SUM
@SELLER VARCHAR(30)
AS
IF EXISTS(SELECT 'TRUE' FROM SALES
INNER JOIN SELLERS ON SALES.ID_SELLER=SELLERS.ID
WHERE SELLERS.LASTNAME = @SELLER)
BEGIN
SELECT SELLERS.FIRSTNAME, SELLERS.LASTNAME, SUM(AMOUNT) FROM SALES
INNER JOIN SELLERS ON SALES.ID_SELLER=SELLERS.ID
WHERE SELLERS.LASTNAME = @SELLER
GROUP BY SELLERS.FIRSTNAME,SELLERS.LASTNAME
END
ELSE
BEGIN
PRINT 'DOES NOT EXIST'
END
```

Корекція:

- видалення відомостей про продажі за минулий рік;

```
CREATE PROC DELETE_SALES
AS
IF EXISTS( SELECT 'TRUE' FROM SALES WHERE YEAR(Date_Sale) = YEAR(GETDATE())-1)
BEGIN
DELETE FROM SALES WHERE YEAR(Date_Sale) = YEAR(GETDATE())-1
END
ELSE
BEGIN
PRINT 'THERE ARE NO PAST SALES ENTRIES'
END
```

-зміна ціни в гривнях у зв'язку із зміною курсу гривні;

```
CREATE PROC CHANGE_CURRENCY
@VALUTA VARCHAR(30),
@NEW_C DECIMAL(10,2)
AS
IF EXISTS(SELECT 'TRUE' FROM CURRENCY WHERE VALUTA=@VALUTA)
BEGIN
UPDATE CURRENCY SET PRICE=@NEW_C WHERE VALUTA=@VALUTA
END
ELSE
BEGIN
PRINT 'DOES NOT EXIST'
```

END

Звіти:

- Характеристики комп'ютера зазначеної марки;

```
CREATE PROC FEATURE_COMP
```

```
@MARK VARCHAR(30)
```

```
AS
```

```
IF EXISTS(SELECT 'TRUE' FROM ITEMS_LIST WHERE MARK=@MARK AND TP='COMPUTER')
```

```
BEGIN
```

```
SELECT FEATURE FROM ITEMS_LIST WHERE MARK=@MARK AND TP='COMPUTER'
```

```
END
```

```
ELSE
```

```
BEGIN
```

```
PRINT 'DOES NOT EXIST'
```

```
END
```

- Інформація про товари; угруповання за типами (окремо - принтери, комп'ютери, монітори і т.п.);

```
SELECT * FROM ITEMS_LIST WHERE TP='COMPUTER'
```

```
SELECT * FROM ITEMS_LIST WHERE TP='PRINTER'
```

```
SELECT * FROM ITEMS_LIST WHERE TP='MONITOR'
```

- Рекламний ярлик до товару (назва, характеристики, ціна)

```
SELECT ID,FEATURE,PRICE FROM ITEMS_LIST
```