НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО

КАФЕДРА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до курсової роботи

з дисципліни

«БАЗИ ДАНИХ ТА ІНФОРМАЦІЙНІ СИСТЕМИ»

на тему

«Інформаційна система ательє з пошиття одягу»

Викладач М. В. Коломицев

Студент М. О. Шестак

Група ФБ-11

Варіант 28

Київ – 2022

***Варіант 28.***

**Інформаційна система ательє з пошиття одягу**

Вхідні дані

* Каталог моделей: тканина, витрата тканини (за розмірами), фурнітура, вартість роботи, ціна.
* Каталог тканин: назва, ширини і ціна за 1 метр, наявність тканини на складі.
* Каталог фурнітури: назва, кількість, ціна за 1 шт.
* Журнал реєстрації замовлень клієнтів: П.І.Б. клієнта, інформація про модель, інформація про тканину, П.І.Б. виконавця замовлення, дату прийому, дату примірки, дату виконання, відмітка про оплати
* Журнал реєстрації замовлень постачальнику: назва тканини, кількість, сума, факт виконання і оплати

Вихідні дані

* Залишки тканини та фурнітури на складі.
* Вибірка тканини, моделі, фурнітури в каталозі за назвою.
* Поточні замовлення.
* Навантаження виконавців замовлень.
* Розрахунок вартості замовленя.
* Чек на оплату (детальний)
* Розширений аналіз замовлень (по сумі, кількості і т.п.)

Функціонал

* Ведення каталогу моделей, тканин, фурнітури.
* Врахування вартості ескізу.
* Облік витрати і залишків тканини та фурнітури.
* Друк чеків на оплату.
* Відстеження завантаженості виконавців
* Контроль оплати з формуванням переліку клієнтів – боржників
* Автоматичне формування замовлення, якщо матеріалу з замовлення немає на складі

**Зміст**

[**Вступ** 4](#_Toc124548546)

[**Аналіз предметної області** 6](#_Toc124548547)

[**Функціональне моделювання предметної області з використанням DFD – діаграм** 7](#_Toc124548548)

[**Концептуальна модель бази даних** 8](#_Toc124548549)

[**Реляційна модель бази даних** 9](#_Toc124548550)

[**Фізична модель бази даних** 10](#_Toc124548551)

[**Опис реалізації завдань відповідно обраного варіанту** 18](#_Toc124548552)

[**Висновки по роботі** 25](#_Toc124548553)

[**Список використаних джерел** 26](#_Toc124548554)

**Вступ**

Інформаційна система ательє з пошиття одягу - це одна з важливих складових ефективної діяльності будь-якого підприємства в цій сфері. Проектування бази даних для інформаційної системи ательє - це складний процес, який включає в себе багато аспектів. Основні етапи проектування бази даних складаються з аналізу вимог, проектування структури бази даних, розробки моделей даних та тестування.

Актуальність цієї роботи спрямована на те, що інформаційна система ательє з пошиття одягу може не тільки сприяти ефективній організації роботи підприємства, але також забезпечити контроль над виробництвом та збереження історії замовлень. Це дозволяє легко керувати процесом виготовлення одягу, відслідковувати замовлення та виконання робіт, а також зберігати історію замовлень, що дозволяє зробити аналіз та покращити процес виробництва.

Мета дослідження - створити інформаційну систему, яка забезпечить ефективну організацію роботи ательє, автоматизує процеси і дозволить керувати даними замовлень, матеріалами та виробництвом.

Завдання дослідження - провести аналіз вимог, створити модель бази даних, реалізувати інформаційну систему та провести тестування.

Порядок реалізації завдань проектування буде включати в себе наступні кроки: аналіз вимог, проектування структури бази даних, розробку моделей даних, реалізацію інформаційної системи та тестування. Для реалізації завдань будуть використовуватися сучасні інструменти та технології проектування та розробки баз даних.

Крім того, інформаційна система ательє з пошиття одягу має бути здатною до зберігання інформації про замовлення, матеріали, постачальників, клієнтів, а також про виробництво, так що це буде зручно і зрозуміло для користування і створення аналітики і звітів. Це дозволить легко відслідковувати статус замовлення, матеріали, які використовуються, інформацію про клієнтів і постачальників та звіти про виробництво.

Також, система повинна мати можливість взаємодії з іншими системами, наприклад, з фінансовою системою або системою збереження документів, що дозволить автоматизувати більшість бізнес-процесів і зберігати інформацію в одному місці. Це дозволить ефективно координувати бізнес-процеси, такі як розрахунок фінансових показників, збереження документів, облік продажів і закупівлі, без необхідності дублювання інформації в різних системах. Це знижує ризик помилок, заощаджує час і збільшує ефективність бізнес-процесів. Також, здатність до взаємодії з іншими системами дозволяє отримувати більш детальну інформацію і проводити більш дослідні аналізи.

В цілому, створення інформаційної системи для ательє з пошиття одягу є ключовим елементом для підвищення ефективності і конкурентоспроможності підприємства, а також для забезпечення інформаційної безпеки та конфіденційності даних. Проектування та реалізація цієї системи включає в себе використання сучасних методів та інструментів, які забезпечують ефективне керування даними і автоматизацію бізнес-процесів. Загалом, ця робота має важливе значення для розвитку і ефективності будь-якого підприємства в сфері пошиття одягу.

**Аналіз предметної області**

Інформаційна система ательє з пошиття одягу призначена для покращення обліку матеріалів, замовлень та інших важливих аспектів цього бізнесу. Покращення обліку призводить до збільшення обсягу товарообігу.

Дані, що можуть бути введені:

* Тканина, інформація про наявні позиції.
* Фурнітура, інформація про наявні позиції.
* Виконавці, інформація про виконавців.
* Клієнти, інформація про клієнтів.
* Моделі, інформація про наявні позиції.
* Замовлення тканини, інформація про замовлення тканини.
* Замовлення від клієнтів, інформація про замовлення від клієнтів.

Дані, що будуть виведені:

* Кількість тканини та фурнітури, що залишилась на складі.
* Детальна інформація по певній тканині, фурнітурі чи моделі.
* Інформація по замовленням від клієнтів, що ще не виконані.
* Кількість замовлень кожного виконавця.
* Вартість кожного замовлення.
* Детальну інформацію по кожному замовленню.
* Список клієнтів боржників з їхнім боргом.

Як можна побачити, дана інформаційна система має певний функціонал, призначений для полегшення обліку товарів. Також є певний функціонал для обліку клієнтів та працівників ательє.

**Функціональне моделювання предметної області з використанням DFD – діаграм**

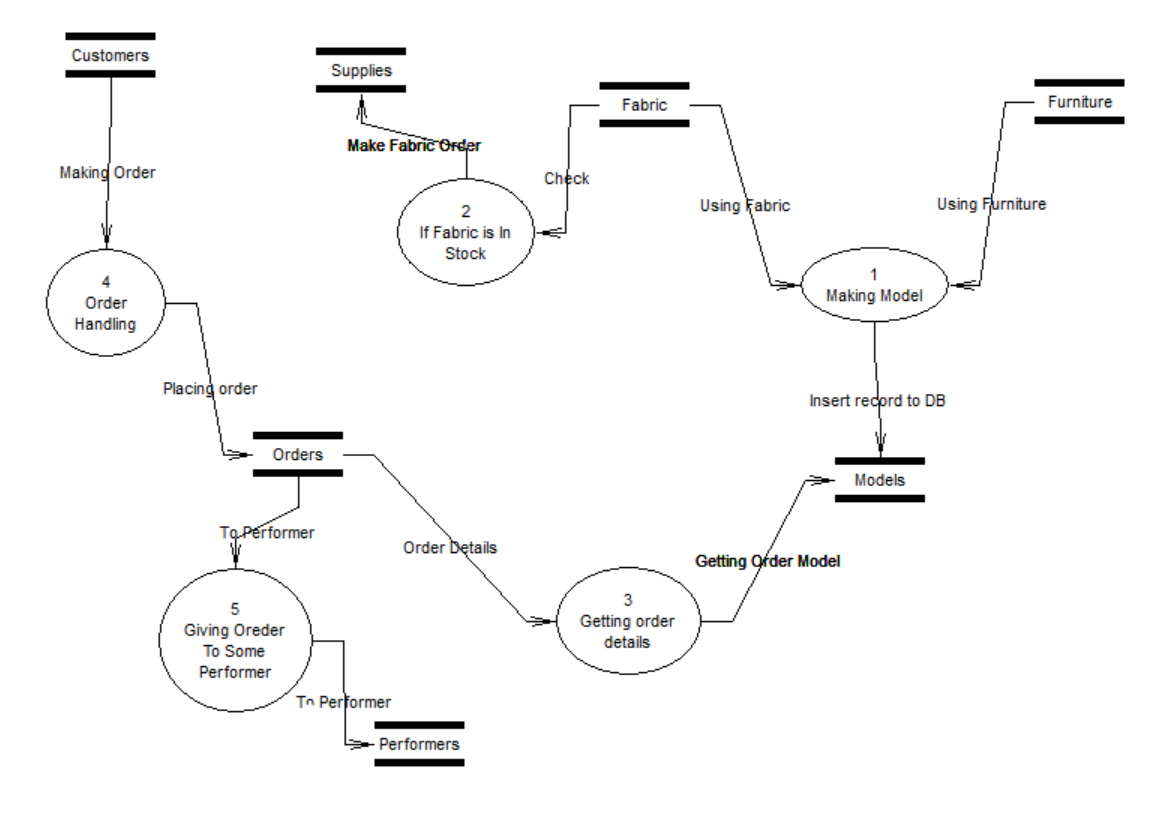
****

Рисунок 1.1

Як бачимо, на рисунку 1.1 представлена DFD-діаграма, має всі необхідні функції для коректного функціонування.

**Концептуальна модель бази даних**

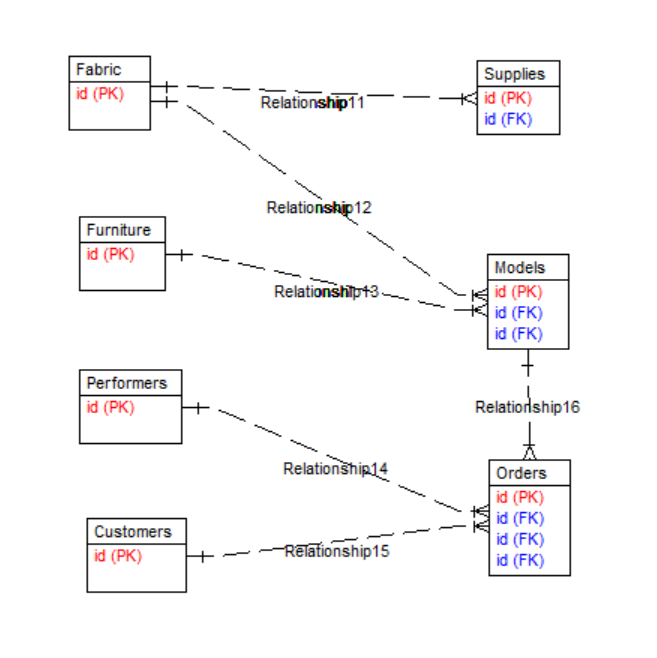


Рисунок 2.1

Як бачимо на рисунку 2.1, даний концепт має всі відношення між таблицями, які в подальшому будуть створені.

**Реляційна модель бази даних**

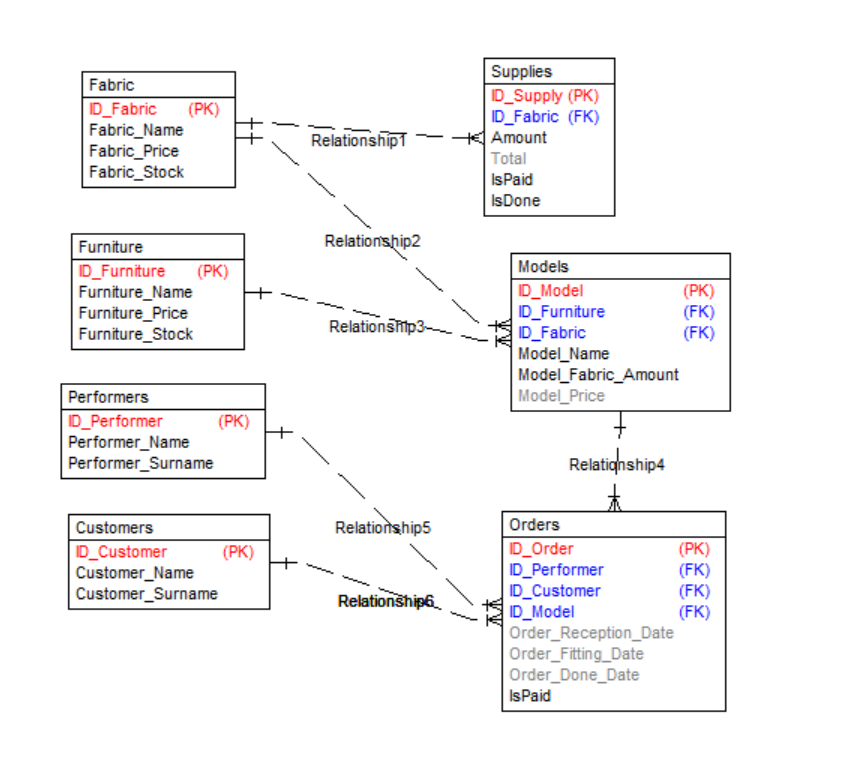


Рисунок 3.1

На рисунку 3.1 зображена реляційна модель (діаграма) бази даних. Створені всі таблиці, поля та відношення, які будуть використовуватись в інформаційній системі ательє з пошиття одягу.

**Фізична модель бази даних**

1. Спочатку створюємо необхідні таблиці:

CREATE TABLE Fabric (

ID\_Fabric int IDENTITY(1,1) PRIMARY KEY,

Fabric\_Name varchar(255) NOT NULL,

Fabric\_Price float NOT NULL,

Fabric\_Stock float NOT NULL,

CONSTRAINT Fabric\_Price\_Check CHECK(Fabric\_Price>0),

CONSTRAINT Fabric\_Stock\_Check CHECK(Fabric\_Stock>=0)

)

CREATE TABLE Furniture (

ID\_Furniture int IDENTITY(1,1) PRIMARY KEY,

Furniture\_Name varchar(255) NOT NULL,

Furniture\_Price float NOT NULL,

Furniture\_Stock float NOT NULL,

CONSTRAINT Furniture\_Price\_Check CHECK(Furniture\_Price>0),

CONSTRAINT Furniture\_Stock\_Check CHECK(Furniture\_Stock>=0)

)

CREATE TABLE Performers (

ID\_Performer int IDENTITY(1,1) PRIMARY KEY,

Performer\_Name varchar(255) NOT NULL,

Performer\_Surname varchar(255) NOT NULL

)

CREATE TABLE Customers (

ID\_Customer int IDENTITY(1,1) PRIMARY KEY,

Customer\_Name varchar(255) NOT NULL,

Customer\_Surname varchar(255) NOT NULL

)

CREATE TABLE Supplies (

ID\_Supply int IDENTITY(1,1) PRIMARY KEY,

ID\_Fabric\_Supply int FOREIGN KEY REFERENCES Fabric(ID\_Fabric),

Amount float NOT NULL,

Total float,

IsPaid bit NOT NULL,

IsDone bit NOT Null,

CONSTRAINT Supplies\_Amount\_Check CHECK(Amount>0),

CONSTRAINT Supplies\_Total\_Check CHECK(Total>0)

)

CREATE TABLE Models (

ID\_Model int IDENTITY(1,1) PRIMARY KEY,

Model\_Name varchar(255) NOT NULL,

ID\_Fabric\_Model int FOREIGN KEY REFERENCES Fabric(ID\_Fabric),

Model\_Fabric\_Amount float NOT NULL,

ID\_Furniture\_Model int FOREIGN KEY REFERENCES Furniture(ID\_Furniture),

Model\_Price float,

CONSTRAINT Model\_Fabric\_Amount\_Check CHECK(Model\_Fabric\_Amount>0),

CONSTRAINT Model\_Price\_Check CHECK(Model\_Price>0)

)

CREATE TABLE Orders (

ID\_Order int IDENTITY(1,1) PRIMARY KEY,

ID\_Model\_Order int FOREIGN KEY REFERENCES Models(ID\_Model),

ID\_Customer\_Order int FOREIGN KEY REFERENCES Customers(ID\_Customer),

ID\_Performer\_Order int FOREIGN KEY REFERENCES Performers(ID\_Performer),

Order\_Reception\_Date date,

Order\_Fitting\_Date date,

Order\_Done\_Date date,

IsPaid bit NOT NULL

)

1. Далі необхідно забезпечити цілісність бази даних, ввівши певні обмеження (див. попередній пункт) та створивши необхідні тригери для забезпечення правдивості та цілісності вводимої інформації.

CREATE TRIGGER SuppliesCorrection

ON Supplies

AFTER INSERT, UPDATE

AS

BEGIN

UPDATE Supplies

SET Total = Fabric.Fabric\_Price \* Amount

FROM Supplies

JOIN Fabric ON Supplies.ID\_Fabric\_Supply = Fabric.ID\_Fabric

END

Тригер вище автоматично додає значення в поле Supplies.Total

CREATE TRIGGER ModelsCorrection

ON Models

AFTER INSERT, UPDATE

AS

BEGIN

UPDATE Models

SET Model\_Price = Fabric.Fabric\_Price \* Model\_Fabric\_Amount + Furniture.Furniture\_Price

FROM Models

JOIN Fabric ON Models.ID\_Fabric\_Model = Fabric.ID\_Fabric

JOIN Furniture ON Models.ID\_Furniture\_Model = Furniture.ID\_Furniture

END

Тригер вище атоматично додає значення в поле Models.Model\_Price

CREATE TRIGGER UpdateFabricStock

ON Orders

AFTER INSERT

AS

BEGIN

DECLARE @amount float

SELECT @amount = Model\_Fabric\_Amount

FROM Models

JOIN Orders ON Models.ID\_Model = Orders.ID\_Model\_Order

UPDATE Fabric

SET Fabric\_Stock = Fabric\_Stock - @amount

FROM Fabric

JOIN Models ON Fabric.ID\_Fabric = Models.ID\_Fabric\_Model

JOIN Orders ON Models.ID\_Model = Orders.ID\_Model\_Order

END

Тригер вище автоматично оновлює значення Fabric.Fabric\_Stock

CREATE TRIGGER UpdateFurnitureStock

ON Orders

AFTER INSERT

AS

BEGIN

UPDATE Furniture

SET Furniture\_Stock = Furniture\_Stock - 1

FROM Furniture

JOIN Models ON Furniture.ID\_Furniture = Models.ID\_Furniture\_Model

JOIN Orders ON Models.ID\_Model = Orders.ID\_Model\_Order

END

Тригер вище автоматично оновлює значення Furniture.Furniture\_Stock

1. Приклади заповнення бази даних:

INSERT INTO Fabric (Fabric\_Name, Fabric\_Price, Fabric\_Stock)

VALUES ('Cotton', 10.50, 50),

('Silk', 20.00, 25),

('Linen', 15.75, 35),

('Wool', 25.00, 15),

('Polyester', 5.00, 75),

('Nylon', 7.50, 60),

('Acrylic', 8.25, 40),

('Rayon', 12.50, 30),

('Satin', 18.75, 20),

('Taffeta', 22.50, 10),

('Tweed', 27.50, 5),

('Velvet', 32.50, 2),

('Corduroy', 37.50, 1),

('Denim', 42.50, 3),

('Flannel', 47.50, 4),

('Suede', 52.50, 6),

('Organza', 57.50, 7),

('Chiffon', 62.50, 8),

('Tulle', 67.50, 9)

INSERT INTO Furniture (Furniture\_Name, Furniture\_Price, Furniture\_Stock)

VALUES ('Table', 100.00, 10),

('Chair', 50.00, 20),

('Sofa', 250.00, 5),

('Bed', 500.00, 2),

('Dresser', 150.00, 15),

('Couch', 300.00, 8),

('Bookshelf', 75.00, 25),

('Desk', 125.00, 18),

('Bedside table', 35.00, 30),

('Armchair', 80.00, 22),

('Coffee table', 60.00, 12),

('Ottoman', 40.00, 14),

('Dining table', 200.00, 6),

('Wardrobe', 400.00, 3),

('Rug', 20.00, 35),

('Lamp', 30.00, 28),

('Mirror', 75.00, 10),

('Pouf', 45.00, 20)

INSERT INTO Performers (Performer\_Name, Performer\_Surname)

VALUES ('John', 'Smith'),

('Jane', 'Doe'),

('Michael', 'Williams'),

('Emily', 'Jones'),

('Matthew', 'Brown'),

('Madison', 'Garcia'),

('Joshua', 'Miller'),

('Ashley', 'Davis'),

('Jacob', 'Rodriguez'),

('Samantha', 'Martinez')

INSERT INTO Customers (Customer\_Name, Customer\_Surname)

VALUES ('Amy', 'Taylor'),

('Brian', 'Moore'),

('Charlie', 'Jackson'),

('Diana', 'Martin'),

('Edward', 'Lee'),

('Felicity', 'Perez'),

('Gary', 'Thompson'),

('Hannah', 'White'),

('Isaac', 'Harris'),

('Julia', 'Young')

INSERT INTO Supplies (ID\_Fabric\_Supply, Amount, IsPaid, IsDone)

VALUES (1, 4.3, 1, 1),

(2, 2.1, 1, 1),

(3, 3.5, 1, 1),

(4, 1.2, 1, 1),

(5, 4.7, 1, 1),

(6, 3.2, 1, 1),

(7, 2.9, 1, 1),

(8, 1.5, 1, 1),

(9, 4.4, 1, 1),

(10, 2.8, 1, 1),

(11, 3.1, 1, 1),

(12, 1.3, 1, 1),

(13, 4.6, 1, 1),

(14, 3.4, 1, 1),

(15, 2.7, 1, 1),

(16, 1.1, 1, 1),

(17, 4.5, 0, 0),

(18, 3.3, 0, 1),

(19, 2.6, 1, 0)

INSERT INTO Models (Model\_Name, ID\_Fabric\_Model, Model\_Fabric\_Amount, ID\_Furniture\_Model)

VALUES ('Model1', 1, 2.5, 5),

('Model2', 3, 1.7, 12),

('Model3', 8, 2.1, 15),

('Model4', 12, 2.3, 4),

('Model5', 18, 1.9, 8),

('Model6', 14, 2.7, 11),

('Model7', 2, 1.2, 7),

('Model8', 9, 3.0, 3),

('Model9', 6, 2.8, 18),

('Model10', 5, 1.5, 16),

('Model11', 15, 2.6, 6),

('Model12', 11, 1.1, 10),

('Model13', 7, 2.9, 13),

('Model14', 17, 3.4, 17),

('Model15', 4, 2.7, 14),

('Model16', 1, 1.3, 2),

('Model17', 16, 2.5, 9),

('Model18', 13, 1.7, 7),

('Model19', 10, 2.1, 1),

('Model20', 19, 2.3, 5)

INSERT INTO Orders (ID\_Model\_Order, ID\_Customer\_Order, ID\_Performer\_Order, Order\_Reception\_Date, Order\_Fitting\_Date, Order\_Done\_Date, IsPaid)

VALUES (5, 4, 7, '2022-01-01', '2022-02-02', '2022-03-03', 1),

(13, 5, 4, '2022-01-02', '2022-02-01', '2022-03-01', 1),

(11, 8, 6, '2022-01-03', '2022-02-04', '2022-03-05', 1),

(4, 9, 1, '2022-01-04', '2022-02-03', '2022-03-02', 1),

(20, 2, 8, '2022-01-05', '2022-02-06', '2022-03-07', 1),

(18, 1, 4, '2022-01-06', '2022-02-05', '2022-03-04', 1),

(2, 5, 5, '2022-01-07', '2022-02-08', '2022-03-09', 1),

(9, 8, 6, '2022-01-08', '2022-02-07', '2022-03-06', 1),

(7, 3, 2, '2022-01-09', '2022-02-10', '2022-03-11', 1),

(12, 9, 8, '2022-01-10', '2022-02-09', '2022-03-08', 1),

(3, 2, 7, '2022-01-11', '2022-02-12', '2022-03-13', 1),

(14, 4, 5, '2022-01-12', '2022-02-11', '2022-03-10', 1),

(16, 5, 6, '2022-01-13', '2022-02-14', '2022-03-15', 1),

(1, 8, 3, '2022-01-14', '2022-02-13', '2022-03-12', 1),

(17, 7, 8, '2022-01-15', '2022-02-16', '2022-03-17', 1),

(19, 10, 7, '2022-01-16', '2022-02-15', NULL, 0),

(10, 2, 5, '2022-01-17', '2022-02-18', NULL, 0),

(8, 3, 6, '2022-01-18', NULL, NULL, 1),

(6, 9, 3, '2022-01-19', NULL, NULL, 1),

(15, 4, 10, '2022-01-20', NULL, NULL, 0)

Всі записи відповідають всім умовам та заносяться коректно.

**Опис реалізації завдань відповідно обраного варіанту**

Всі реалізації завдань було виконано за допомогою створення процедур.

1. Показати, скільки залишилось тканини та фурнітури на складі:

CREATE PROCEDURE ShowStock

AS

BEGIN

SELECT 'Fabric' as ProductType, Fabric\_Name, Fabric\_Stock FROM Fabric

UNION

SELECT 'Furniture' as ProductType, Furniture\_Name, Furniture\_Stock FROM Furniture

END

Після команди EXEC ShowStock; бачимо наступний результат:



Рисунок 4.1

Тобто спочатку виводиться назви та кількості тканин, що залишились на складі, потім аналогічно фурнітура.

1. Вивести детальну інформацію по певній тканині, фурнітурі чи моделі:

CREATE PROCEDURE GetProduct

@ProductName varchar(255)

AS

BEGIN

IF EXISTS (SELECT Fabric\_Name FROM Fabric WHERE Fabric\_Name = @ProductName)

BEGIN

SELECT 'Fabric' as ProductType, ID\_Fabric, Fabric\_Name, Fabric\_Price, Fabric\_Stock FROM Fabric

WHERE Fabric\_Name = @ProductName

END

ELSE IF EXISTS (SELECT Furniture\_Name FROM Furniture WHERE Furniture\_Name = @ProductName)

BEGIN

SELECT 'Furniture' as ProductType, ID\_Furniture, Furniture\_Name, Furniture\_Price, Furniture\_Stock FROM Furniture

WHERE Furniture\_Name = @ProductName

END

ELSE IF EXISTS (SELECT Model\_Name FROM Models WHERE Model\_Name = @ProductName)

BEGIN

SELECT 'Model' as ProductType, ID\_Model, Model\_Name, Model\_Price, Model\_Fabric\_Amount FROM Models

WHERE Model\_Name = @ProductName

END

ELSE

BEGIN

PRINT 'Product not found'

END

END

Після виконання команди EXEC GetProduct @ProductName = 'Cotton'; (замість 'Cotton' може бути будь-яка інша існуюча назва) бачимо наступний результат:



Рисунок 4.2

Тобто виводиться інформація про певний продукт.

1. Вивести інформацію по замовленням від клієнтів, що ще не виконані:

CREATE PROCEDURE ShowIncompletedOrders

AS

BEGIN

SELECT ID\_Order, ID\_Model\_Order, ID\_Customer\_Order, ID\_Performer\_Order, Order\_Reception\_Date, Order\_Fitting\_Date, Order\_Done\_Date, IsPaid

FROM Orders

WHERE Order\_Done\_Date IS NULL

END

Після виконання команди EXEC ShowIncompletedOrders; бачимо наступне:

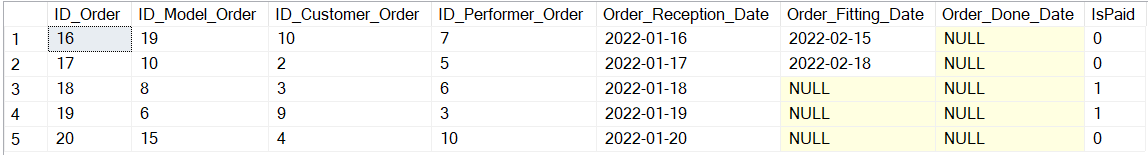


Рисунок 4.3

Тобто виводяться ті замовлення, що мають поле Order\_Done\_Date заповнене значенням NULL.

1. Вивести кількість замовлень кожного виконавця:

CREATE PROCEDURE ShowPerformersWithOrderCount

AS

BEGIN

SELECT ID\_Performer, Performer\_Name, Performer\_Surname, COUNT(ID\_Performer\_Order) as Order\_Count

FROM Performers

LEFT JOIN Orders ON Performers.ID\_Performer = Orders.ID\_Performer\_Order

GROUP BY ID\_Performer, Performer\_Name, Performer\_Surname

END

Після виконання команди EXEC ShowPerformersWithOrderCount; бачимо наступне:

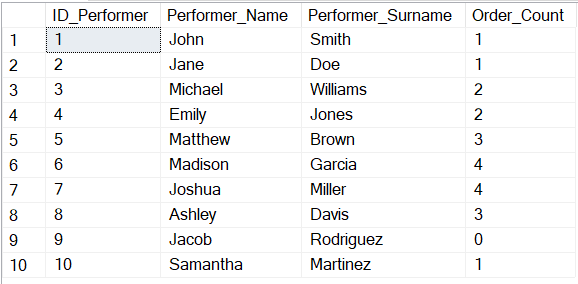


Рисунок 4.4

Тобто вивід з інформацією про кожного виконавця та кількість замовлень у кожного.

1. Вивести вартість кожного замовлення:

CREATE PROCEDURE ShowOrdersWithPrice

AS

BEGIN

SELECT ID\_Order, Models.Model\_Price as Order\_Price

FROM Orders

JOIN Models ON Orders.ID\_Model\_Order = Models.ID\_Model

END

Після виконання команди EXEC ShowOrdersWithPrice; бачимо наступний вивід:

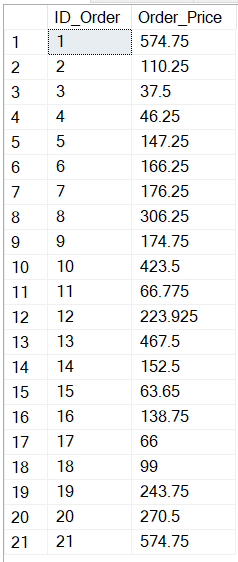


Рисунок 4.5

Тобто id кожного замовлення та вартість моделі, що зазначена у замовленні.

1. Вивести детальну інформацію по кожному замовленню:

CREATE PROCEDURE ShowOrdersDetails

AS

BEGIN

SELECT

ID\_Order,

Models.Model\_Name as Model\_Name,

Models.Model\_Price as Model\_Price,

Customers.Customer\_Name as Customer\_Name,

Customers.Customer\_Surname as Customer\_Surname,

Performers.Performer\_Name as Performer\_Name,

Performers.Performer\_Surname as Performer\_Surname,

Order\_Reception\_Date as Reception\_Date

FROM Orders

JOIN Models ON Orders.ID\_Model\_Order = Models.ID\_Model

JOIN Customers ON Orders.ID\_Customer\_Order = Customers.ID\_Customer

JOIN Performers ON Orders.ID\_Performer\_Order = Performers.ID\_Performer

END

Після виконання команди EXEC ShowOrdersDetails; бачимо наступне:



Рисунок 4.6

Тобто всю інформацію по кожному замовленню.

1. Вивести список клієнтів боржників з їхнім боргом:

CREATE PROCEDURE ShowCustomerDebt

AS

BEGIN

SELECT

ID\_Customer,

SUM(Models.Model\_Price) as Customer\_Debt

FROM Orders

JOIN Models ON Orders.ID\_Model\_Order = Models.ID\_Model

JOIN Customers ON Orders.ID\_Customer\_Order = Customers.ID\_Customer

WHERE IsPaid = 0

GROUP BY ID\_Customer

END

Після виконання команди EXEC ShowCustomerDebt; бачимо наступне:

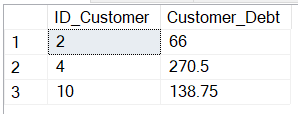


Рисунок 4.7

Тобто id клієнта та борг перед магазином.

**Висновки по роботі**

Було розроблено та створено інформаційну систему ательє з пошиття одягу, яка повноцінно функціонує та виконує певний спектр завдань.

Спочатку було створено три діаграми у середовищі Case Studio 2. Виконано функціональне моделювання предметної області з використанням DFD – діаграми, створена концептуальна модель бази даних та реляційна модель бази даних. На основі даних розробок було почато створення фізичної моделі бази даних. Після цього, над фізичною моделлю були проведені заходи щодо забезпечення цілісності даної моделі. Наступним етапом було наповнення її тестовими даними.

Також були створені ряд процедур, що забезпечували коректне виконання поставлених перед нею завдань. Всі завдання були виконані у відповідності з умовою.

Дану курсову роботу можна розглядати, як ще достатньо сирий проте працюючий варіант справжньої інформаційної системи ательє з пошиття одягу та використовувати в подальшій розробці справжнього проекту.

**Список використаних джерел**

1. "Інформаційні системи в менеджменті виробництва" А.В. Коваленко
2. "Інформаційні технології у діяльності підприємства" О.О. Лізунов
3. "ERP-системи у текстильній промисловості" І.В. Степанюк
4. "Автоматизація виробничої діяльності у пошивній промисловості" О.В. Горохов
5. "Інформаційні системи управління виробництвом" М.О. Костюк
6. "Інформаційні технології у менеджменті пошивного виробництва" О.В. Калінін