

Національний технічний університет України

“Київський політехнічний інститут”

БАЗИ ДАНИХ ТА ІНФОРМАЦІЙНІ СИСТЕМИ

**Розрахункова робота на тему:**

«Облік магазину по продажу аудіо та відео товарів»

Виконав:

Студент гр. ФІ-91

Варіант 12

Шеремета Назар

Перевірив:

Коломицев М.В.

Київ 2012

**Зміст**

[Вступ 3](#_Toc343162776)

[1. Проектування баз даних 4](#_Toc343162777)

[1.1. Системний аналіз предметної області 4](#_Toc343162778)

[1.2. Концептуальне проектування 6](#_Toc343162779)

[1.3. Логічне проектування 7](#_Toc343162780)

[2. Завдання 8](#_Toc343162781)

[3. Створення діаграми потоків даних (DFD-моделі) 9](#_Toc343162782)

[4. Створення концептуальної моделі (EF-моделі) 12](#_Toc343162783)

[5. Проектування бази даних у середовищі MS SQL 13](#_Toc343162784)

[6. Створення необхідного функціонування 16](#_Toc343162785)

[6.1. Створення впорядкованих списків 16](#_Toc343162786)

[6.2. Перевірка на наявність вільних місць для жінок чи чоловіків 17](#_Toc343162787)

[6.3. Корегування даних 19](#_Toc343162788)

[6.4. Створення звіту 19](#_Toc343162789)

[7. Перевірка нормалізації розробленої моделі 22](#_Toc343162790)

[Висновок 23](#_Toc343162791)

# Вступ

Основною метою даної розрахункової роботи є освоєння навичок проектування баз даних, в частковому випадку бази даних постояльців у готелі.

Етап проектування даних зазвичай відбувається з участю потенційних користувачів. Іноді в процесі проектування даних створюються прототипи працюючих додатків, що дозволяють уточнити і доповнити вимоги до кінцевого продукту. Якщо передбачається, що впровадження нової інформаційної системи буде супроводжуватися виведенням з експлуатації її попередника, то приймаються рішення про те, яким чином використовувати успадковані дані, і в модель даних, що є підсумком цього етапу, вносяться необхідні зміни.

Автори багатьох робіт, присвячених загальним принципам розробки проектів інформаційних систем, стверджують, що вартість виправлення помилки, допущеної на попередньому етапі життєвого циклу, приблизно в десять разів перевищує витрати на її виправлення на поточному етапі. Зокрема, багато розробників стикаються з тим, що помилки проектування даних приводять іноді до написання коду великого обсягу, так чи інакше їх компенсуючого, і нерідко викликають проблеми на етапі супроводу готового продукту. Оскільки проектування даних слід безпосередньо за перед проектних обстеженням, дуже важливо, щоб ця частина роботи над проектом була виконана максимально якісно. Саме важливість цього етапу зумовила стрімке зростання популярності такої категорії програмного забезпечення, як засоби проектування даних, протягом останніх десяти років.

# Проектування баз даних

Проектування **бази даних** - це ітераційний, багатоетапний процес прийняття аргументованих рішень під час аналізу інформаційної моделі предметної області, вимог до даних з боку прикладних програмістів і користувачів, синтезу логічних і фізичних структур даних, аналізу та обґрунтування вибору програмних та апаратних засобів.

Проектування виконують за допомогою вивчення ПЗ та вимог, які ставляться до ІС. На цій «паперовій» стадії життя системи вибирають:

* структуру даних і стратегію їх зберігання у пам'яті ЕОМ;
* технологію обслуговування ІС та взаємодію з нею кінцевих користувачів;
* технічні та стандартні програмні засоби; а також розробку оригінальних програмних.

**Етапи проектування БД:**

* системний аналіз предметної області;
* концептуальне проектування;
* логічне проектування;
* фізичне проектування.

## Системний аналіз предметної області

**Системний аналіз** передбачає мовний опис реальних об'єктів предметної області, визначення зв'язків між об'єктами, дослідження характеристик об'єктів і зв'язків. Результати дослідження використовуються при концептуальному проектуванні БД.

Для визначення складу і структури предметної області застосовуються або функціональний, або предметний підходи.

***Функціональний підхід*** застосовує рух "від задач" і використовується у тих випадках, коли заздалегідь відомі функції майбутніх користувачів БД, а також відомі всі задачі, для інформаційних потреб яких створюються БД. В цьому випадку на основі виробничих документів, опитувань замовників можна чітко визначити мінімальний набір об'єктів предметної області та їх взаємозв'язок.

***Предметний підхід*** застосовується у тому випадку, коли інформаційні потреби майбутніх користувачів чітко не визначені. В цьому випадку не можна чітко визначити мінімальний набір об'єктів предметної області. В опис предметної області включаються об'єкти та зв'язки, які є найбільш характерними та найбільш суттєвими для неї. БД називається предметною і може використовуватися для розв’язання задач, які заздалегідь не визначені.

У практичній діяльності використовується комплексний підхід, який з одного боку дозволяє розв’язувати конкретні інформаційні та функціональні задачі, а з іншого боку – враховує можливість додавання нових застосувань. У загальному випадку існує два підходи до проектування БД: низхідне проектування і висхідне проектування.

***Низхідне проектування*** починається з визначення наборів даних, потім визначаються елементи даних для кожного з таких наборів. Цей процес включає в себе ідентифікацію різних типів сутностей і визначення атрибутів кожної сутності.

Низхідне проектування включає операції декомпозиції, що передбачає заміну вихідної множини відношень, що входять в схему БД, іншою множиною відношень, які є проекціями вихідних відношень. Цей підхід рекомендується застосовувати у тих випадках, коли кількість, різноманітність та складність сутностей, зв'язків і транзакцій значна за розмірами. Найбільш поширеними моделями для цього проектування є моделі "сутність − зв'язок" (ER-моделі, Entity-Relationship model).

***Висхідне проектування*** починається з виявлення елементів даних, які потім групуються в набори даних. Спочатку визначаються атрибути, які потім об'єднуються в сутності. Висхідне проектування включає операції синтезу, що передбачає виконання компоновки із заданої множини функціональних залежностей між об'єктами предметної області вихідних відношень схеми БД.

Цей підхід рекомендується застосовувати у тому випадку, якщо розробляється невелика БД з незначною кількістю об'єктів, атрибутів і транзакцій.

## Концептуальне проектування

З **концептуального проектування** починається створення концептуальної схеми БД, в основі якої лежить концептуальна модель даних.

***Концептуальна модель*** представляє загальний погляд на дані. Розрізняють два головних підходи до моделювання даних при концептуальному проектуванні:

− семантичні моделі;

− об'єктні моделі.

*Семантичні моделі* головну увагу приділяють структурі даних. Найбільш поширеною семантичною моделлю є модель *"сутність – зв'язок"* (Entity Relationship model, ER-модель). ER-модель складається із сутностей, зв'язків, атрибутів, доменів атрибутів, ключів. Моделювання даних відображає логічну структуру даних, так само, як блок-схеми алгоритмів відображають логічну структуру програми.

*Об'єктні моделі* головну увагу приділяють поведінці об'єктів даних і засобам маніпуляції даними. Головне поняття таких моделей − об'єкт, тобто сутність, яка має стан і поведінку. Стан об'єкта визначається сукупністю його атрибутів, а поведінка об'єкта визначається сукупністю операцій специфікованих для нього.

Зближення цих моделей реалізується в розширеному ER-моделюванні (Extended Entity Relationship model, EER-модель).

## Логічне проектування

Логічне проектування полягає в створенні логічної моделі на основі вибраної моделі даних. На цьому етапі необхідно вже знати яка СУБД буде застосовуватися в системі (ієрархічна, мережна, реляційна, об'єктно-орієнтована). Для перевірки вірності логічної моделі застосовується нормалізація. Крім того логічна модель перевіряється на умову забезпечення всіх транзакцій користувачів. Фізичне проектування полягає в описі засобів фізичної реалізації логічного проекту БД. Фізичні моделі визначають засоби розміщення даних в середовищі зберігання і засоби доступу до цих даних, які підтримуються на фізичному рівні.

У процесі проектування визначається структура реляційної БД (склад таблиць, їх структура і логічні зв'язки). Структура таблиці визначається складом стовпців, типом даних і розмірами стовпців, ключами таблиці.

# Завдання

**Предметная область** - магазин по продаже аудио-, видеотехники и деталей к ним.

**Решаемые задачи**: учет товаров и их поставщиков.

1. **Задание:**

1. Создать упорядоченные списки :

- Прайс-лист: Товар – фирма - поставщик, цена товара;

- видеомагнитофоны по цене от X до Y; телевизоры фирмы ХХ;

- фирма ХХ и все, что с ней связано.

2. Вычисления:

- количество типов товаров, поставляемых фирмой ХХ;

- средняя цена на телевизоры фирмы ХХ

3. Коррекция:

- удаление сведений о поставщике и всех его товарах;

- изменение цены на товары заданного поставщика на ХХ % (например, телевизоры фирмы SONY на 5 %);.

4. Отчеты вида:

- информация о телевизорах (марка, цена, название, адрес и телефон фирмы -поставщика), группировка по полю "фирма";

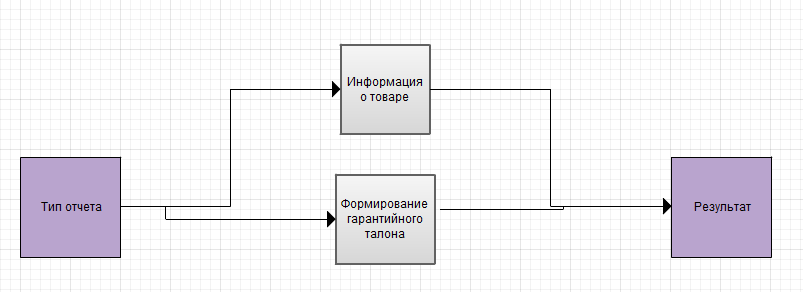
- Гарантийный талон на товар (марка и цена товара, фирма-производитель, срок гарантии, название и адрес магазина ).

# Створення діаграми потоків даних (DFD-моделі)

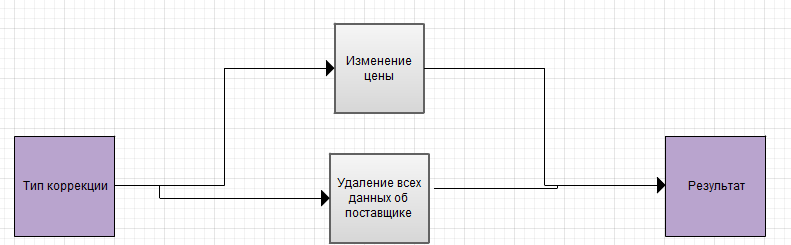
Після аналізу завдання, розробляємо можливу діаграму потоків даних. Для створення DFD-діаграми будемо користуватися програмою Microsoft visio.

Для більш компактного і структурованого проектування визначемо декілька підсистем.

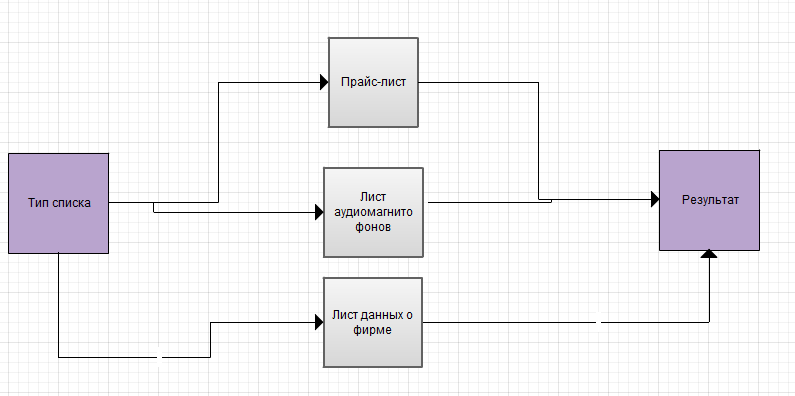
Результат роботи можно побачити на мал. 1,2,3,4.



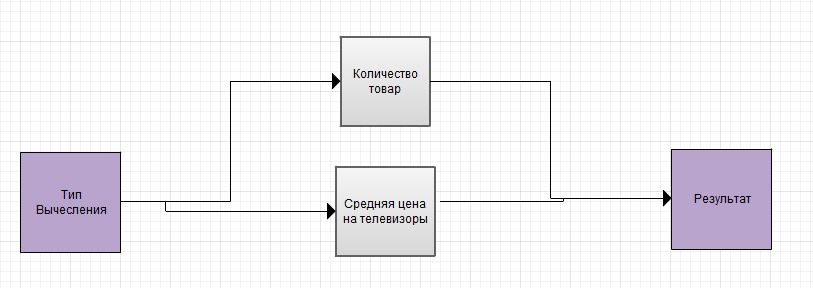
Мал. 1. Підсистема «Визначення типу звіту».



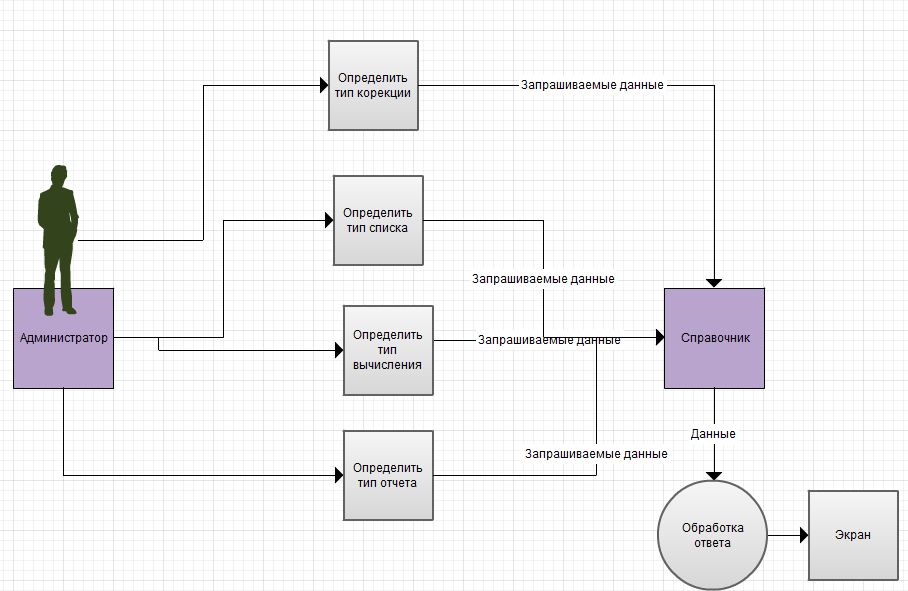
Мал. 2.Підсистема «Визначення типу корегування».



Мал. 3. Підсистема «Визначення типу необхідного списку».



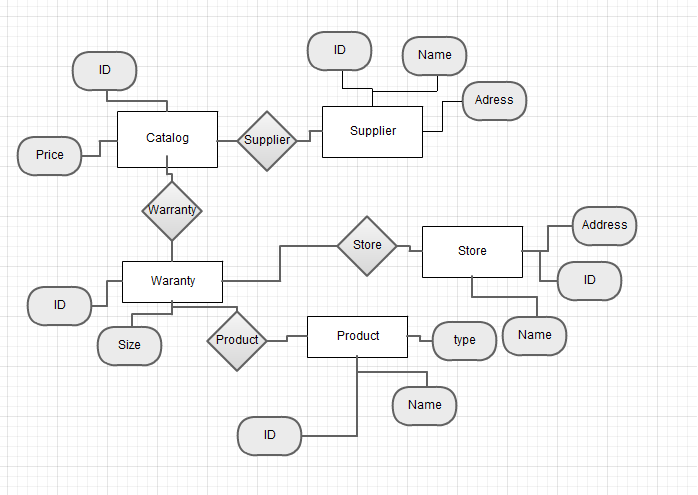
Мал. 4. Підсистема «Визначення типу обчислення»



Мал. 5. Головна діаграма потоків даних (DFD).

# Створення концептуальної моделі (EF-моделі)

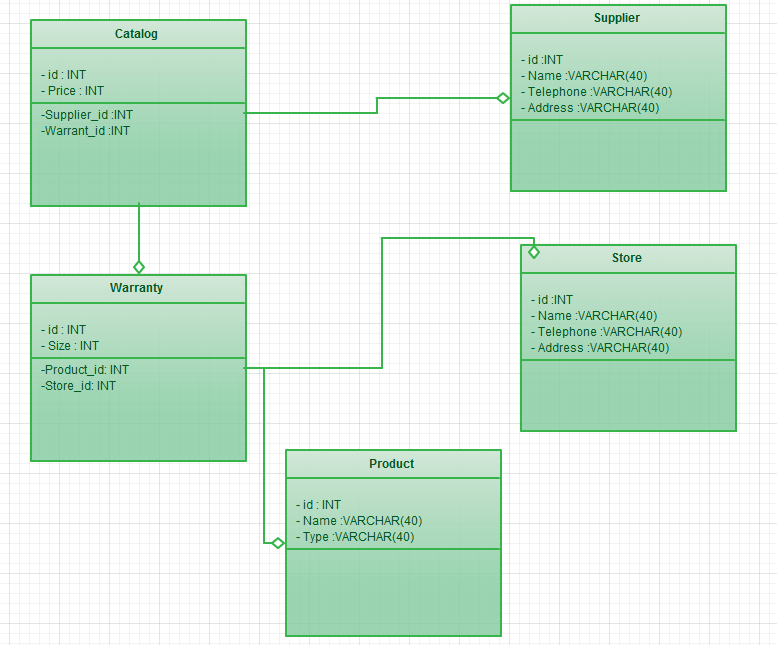
Проаналізувавши діаграму потоків даних, а також мету проектування і засоби для створення, будуємо концептуальну модель (мал. 5).



Мал. 5. Концептуальна модель даних

# Проектування бази даних у середовищі MS SQL

Проаналізувавши усі діаграми, а також предметну область, побудуємо діаграму класів, яка однозначно задасть сутності, атрибути і ключі для MS SQL бази даних (мал. 6).



Мал. 6. Діаграма класів

Виходячи з цього, SQL-код для створення потрібних сутностей буде:

CREATE TABLE Supplier(

ID INT IDENTITY NOT NULL PRIMARY KEY,

Address NVARCHAR(50) NOT NULL,

Telephone NVARCHAR(9),

Name NVARCHAR(40) NOT NULL

);

CREATE TABLE Product(

ID INT IDENTITY NOT NULL PRIMARY KEY,

Name NVARCHAR(50) NOT NULL,

Type NVARCHAR(20)

);

CREATE TABLE Store(

ID INT IDENTITY NOT NULL PRIMARY KEY,

Address NVARCHAR(50) NOT NULL,

Telephone NVARCHAR(9),

Name NVARCHAR(40) NOT NULL

);

CREATE TABLE Catalog(

ID INT IDENTITY NOT NULL PRIMARY KEY,

Supplier\_ID INT,

--Product\_ID INT,

Price INT,

Warrant\_ID INT

);

CREATE TABLE Warrant(

ID INT IDENTITY NOT NULL PRIMARY KEY,

Product\_ID INT,

Store\_ID INT,

Size INT

);

ALTER TABLE Catalog

ADD CONSTRAINT CatalogForeign FOREIGN KEY (Supplier\_ID) REFERENCES Supplier(ID) ON DELETE CASCADE;

ALTER TABLE Catalog

ADD CONSTRAINT CatalogForeign2 FOREIGN KEY (Warrant\_ID) REFERENCES Warrant(ID) ON DELETE CASCADE;

ALTER TABLE Warrant

ADD CONSTRAINT WarrantForeign FOREIGN KEY (Product\_ID) REFERENCES Product(ID) ON DELETE CASCADE;

ALTER TABLE Warrant

ADD CONSTRAINT WarrantForeign2 FOREIGN KEY (Store\_ID) REFERENCES Store(ID) ON DELETE CASCADE;

--INSERT

INSERT INTO Supplier(Address,Telephone,Name) VALUES

('Kiev','4131109','SONY') , ('New York','222bbb22','HTC'), ('Moscow','222323','AKG'),

('Kharkiv','44422111','Sennheiser'),('Paris','222323','Bayerndynamics')

INSERT INTO Store(Address,Telephone,Name) VALUES

('Kiev','4131109','IKEA') , ('New York','222bbb22','EBAYSTORE'), ('Moscow','222323','WALLMART'),

('Kharkiv','44422111','247SHOP'),('Paris','222323','SHOPSHOPSHOP')

INSERT INTO Product(Name,Type) VALUES

('AMG221','Audio'),('PROGJ100','Audio'),('BB99','Audio'),('FF99','Video'),('HUHU2','Video'),('A3','Video')

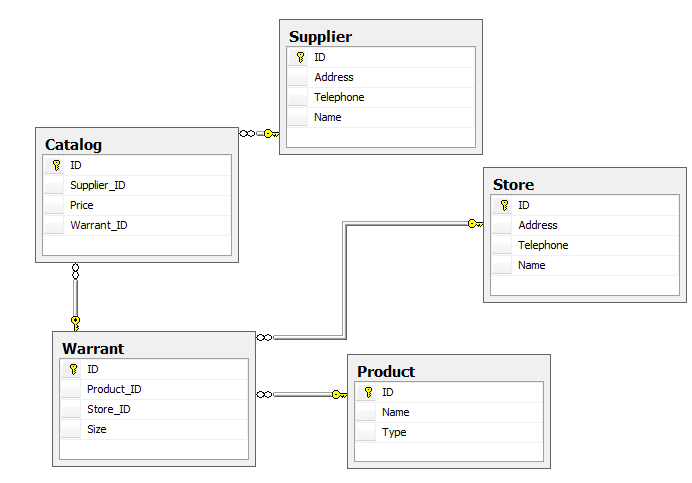
INSERT INTO Warrant(Product\_ID,Store\_ID,Size) VALUES

(1,1,12),(1,3,12),(1,4,12),(1,5,12),(1,4,6),(2,1,12),(3,1,12),(4,1,12),(5,1,12),(6,1,12)

INSERT INTO Catalog(Supplier\_ID,Price,Warrant\_ID) VALUES

(1,1000,3),(1,1500,2),(1,2500,4),(1,100,1),(1,1000,6),(2,1000,3),(3,1000,3),(4,1000,7),(5,5000,3)

Якщо побудувати діаграму створеної бази даних, то можемо побачити таблиці та зв’язки між ними (мал. 7).



Мал. 7. Діаграма усіх таблиць бази даних AudioVideoStore

# Створення необхідного функціонування

За допомогою представлень, збережених процедур і функцій, створюємо необхідний функціонал.

## Створення впорядкованих списків

Створюємо списки, згідно з варіантом завдання.

А Прайс-лист: Товар – фирма - поставщик, цена товара;

CREATE VIEW CatalogView

AS

SELECT Product.Name as Name,Product.Type,Supplier.Name as Supplier,Catalog.Price

FROM Catalog,Supplier,Product,Warrant

WHERE Catalog.Supplier\_ID = Supplier.ID AND Catalog.Warrant\_ID = Warrant.ID AND Warrant.Product\_id = Product.ID

Б) -- аудиомагнитофоны по цене от X до Y, фирмы ХХ;

CREATE PROCEDURE ViewCatalogFiler @Firm nvarchar(30) = NULL,@low INT, @high INT

AS

SELECT \* FROM CatalogView

WHERE TYPE = 'Audio' AND Supplier = @Firm AND Price > @low AND Price < @high

В) данные о фирме ХХ:

CREATE PROCEDURE ViewFirmINFO @FIRM nvarchar(30) = NULL

AS

SELECT Supplier.Name,Supplier.Telephone,Supplier.Address

FROM Supplier

## Створення обчислень

А) Количество типов товаров, поставляемых фирмой ХХ;

CREATE PROCEDURE getNumProducts @Firm nvarchar(30) = NULL

AS

SELECT COUNT(\*) AS RESULT

FROM Supplier,Catalog,Warrant,Product

WHERE Supplier.Name = @Firm AND Catalog.Supplier\_id = Supplier.ID AND Catalog.Warrant\_ID = Warrant.ID AND

Warrant.Product\_ID = Product.ID

--EXEC getNumProducts @firm = 'AKG'

Б) Средняя цена на телевизоры фирмы ХХ

CREATE PROCEDURE getAveragePriceTV

AS

SELECT Product.Name,Product.Type,AVG(Catalog.Price)

FROM Supplier,Catalog,Warrant,Product

WHERE Product.Type = 'TV' AND Catalog.Supplier\_id = Supplier.ID AND Catalog.Warrant\_ID = Warrant.ID AND

Warrant.Product\_ID = Product.ID

GROUP BY Product.Name,Product.Type

## Корегування даних

А ) Удаление сведений о поставщике и всех его товарах;

CREATE PROCEDURE deleteSuplier @Name nvarchar(40) = NULL

AS

DELETE FROM Supplier

WHERE

Supplier.Name = @Name

Б) изменение цены на товары заданного поставщика на ХХ % (например, телевизоры фирмы SONY на 5 %);.

CREATE PROCEDURE raiseBy @Name nvarchar(40) = NULL, @factor INT = 1

AS

UPDATE CATALOG

SET Price = Price\*@factor

WHERE Catalog.Supplier\_ID = (SELECT TOP 1 Supplier.ID FROM Supplier WHERE Supplier.Name = @NAME)

## Створення звіту

А) Информация о товаре (марка, цена, название, адрес и телефон фирмы -поставщика), группировка по полю "фирма";

CREATE VIEW fullInfo

AS

SELECT Product.Name as Name,Product.Type,Supplier.Name as Supplier,Supplier.Address,Supplier.Telephone,Catalog.Price

FROM Catalog,Supplier,Product,Warrant

WHERE Catalog.Supplier\_ID = Supplier.ID AND Catalog.Warrant\_ID = Warrant.ID AND Warrant.Product\_id = Product.ID

CREATE PROCEDURE viewFullInfo

AS

SELECT \* FROM fullinfo

Б) Гарантийный талон на товар (марка и цена товара, фирма-производитель, срок гарантии, название и адрес магазина ).

CREATE VIEW WarrantyView

AS

SELECT Product.Name,Product.Type,Supplier.Name as Supplier,Warrant.Size,Store.Name as Store,Store.Address

FROM Catalog,Supplier,Product,Warrant,Store

WHERE Catalog.Warrant\_ID = Warrant.ID AND Store.ID = Warrant.Store\_ID AND Product.ID = Warrant.Product\_ID

AND Catalog.Supplier\_ID = Supplier.ID

CREATE PROCEDURE ViewWarrantyView

AS

SELECT \* FROM WarrantyView

# Перевірка нормалізації розробленої моделі

Для того, щоб це перевірити, перевіримо чи належить дана модель до третьої нормальної форми. Для цього модель повинна належати до першої і другої форми, а також не повинна мати транзитивних відношень, тобто залежностей не ключевого атрибуту від іншого не ключового.

Для початку перевіримо належність до першої нормальної форми. Для цього усі атрибути повинні буди атомарні і повинні бути відсутні групи, які повторюються. Це виконується, виходячи з того, що в таблицях немає повторювальних даних чи груп.

Далі перевіримо належність до другої нормальної групи. Це буде виконуватись, якщо модель належить до першої нормальної групи (це вже перевірено), а також будуть відсутні неповні функціональні залежності не ключових атрибутів первинного ключа., тобто будь-яке не ключеві поле має однозначно ідентифікуватись повним набором ключів. Це виконується, тому що будь-яке поле унікальне і визначається набором ключів.

Залишилось перевірити відсутність транзитивних відношень, тобто ні одне з ключових полів не має ідентифікуватися за допомогою іншого ключового поля. У побудованій моделі немає полів, які ідентифікуються в залежності від іншого ключового поля. Значить розроблена модель належить до третьої нормальної форми.

З цього слідує, що процес нормалізації чи приведення до третьої нормальної форми не потрібен, бо він вже досягнуто.

**Висновок**

У ході виконання розрахунково-графічної роботи, я дослідив методи проектування баз даних, навчився проектувати власну базу даних у програмі Enterprise Erchitect, а також створювати різного роду функціонування для потреб користувача.