

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет      Магістерської підготовки

Кафедра      Інформаційних технологій

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему: Частина 1. Моделювання та проектування  
корпоративного сховища даних

Виконав студент 2 курсу групи  
МІС-19 спеціальності 122  
Комп'ютерні науки

Постолов Юрій Іванович

Керівник к.ф.-м.н., доцент  
Козловська Валентина Петрівна

Рецензент д.т.н, професор  
Великодний Станіслав Сергійович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської підготовки  
Кафедра інформаційних технологій

**Комплексна магістерська кваліфікаційна робота**

на тему: Моделювання та розробка корпоративного сховища даних

Склад:

**Частина 1** Моделювання та проектування корпоративного сховища даних

Виконавець: ПостоловЮрій Іванович

Керівник к.ф.-м.н., доцент

Козловська Валентина Петрівна

**Частина 2** Розробка корпоративного сховища даних

Виконавець: Постолова Ірина Василівна

Керівник к.ф.-м.н., доцент

Козловська Валентина Петрівна

Староста роботи: ПостоловВіктор Іванович

Провідний керівник: к.ф.-м.н., доцент Козловська Валентина Петрівна

Рецензент: д.т.н., професор Ведикодний Станіслав Сергійович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

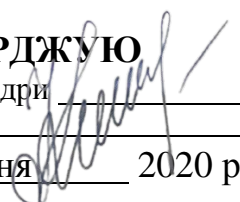
Факультет Магістерської підготовки

Кафедра інформаційних технологій

Рівень вищої освіти магістр

напрямок 122 Комп'ютерні науки

**ЗАТВЕРДЖУЮ**

Завідувач кафедри 

“ 26 ” ЖОВТНЯ 2020 р.

**З А В Д А Н Н Я**  
**НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Постолову Юрію Івановичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Моделювання та розробка корпоративного сховища даних. Частина 1. Моделювання та проектування корпоративного сховища даних»

керівник роботи Козловська Валентина Петрівна, к.ф.-м.н, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ 16 ” ЖОВТНЯ 2020 р. №235-С

2. Строк подання студентом проекту 07.12.2020 р.

3. Вихідні дані до роботи Вимоги до корпоративного СД основних користувачів

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

У вступі викладаються мета та задачі кваліфікаційної роботи

У перших розділах проводиться опис та аналіз предметної області, виконується її моделювання

Наводиться огляд та вибір існуючих моделей сховищ даних

Описується спроектоване сховище даних

У висновках підводяться підсумки виконаної роботи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

### 6. Консультанти розділів роботи


Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 26 жовтня 2020р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1	Огляд та аналіз предметної області	26.10-28.10	90	Вірк
2	Огляд існуючих моделей СД	28.10-30.10	90	Вірк
3	Моделювання СД	31.10-02.11	90	Вірк
4	Проектування СД	03.11-18.11	90	Вірк
	Рубіжна атестація	19.11	90	Вірк
5	Розробка серверного програмного забезпечення	19.11-30.11	90	Вірк
6	Оформлення пояснювальної записки	30.11-06.12	90	Вірк
	Подання роботи на кафедру	07.12		
	Перевірка на плагіат	08.12		
	Рецензування	16.12		
	Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)		90	Вірк

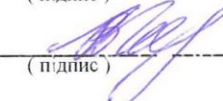
Студент

  
(підпис)

Постолов Ю.І.

(прізвище та ініціали)

Керівник роботи

  
(підпис)

Козловська В.П.

(прізвище та ініціали)

## АНОТАЦІЯ

на магістерську кваліфікаційну роботу

Проектування та розробка корпоративного сховища даних

Частина 1. Моделювання та проектування корпоративного сховища даних,

студента Постолова Юрія Івановича

Кваліфікаційна магістерська робота: 72 с., 10 рис., 14 джерел.

Ключові слова: РЕЛЯЦІЙНА БАЗА ДАНИХ, СХОВИЩЕ ДАНИХ, ТЕМПОРАЛЬНІ БАЗИ ДАНИХ, SQL, СКБД MS SQL Server.

*Мета дослідження* – моделювання корпоративного сховища даних.

*Об'єкт дослідження* – методики побудови моделі корпоративного сховища даних.

*Предмет дослідження* – моделювання корпоративного сховища даних для компанії з Інтернет-продажів.

*Задачі дослідження:* проаналізувати сучасні проектні рішення побудови корпоративних сховищ даних, виконати моделювання та проектування сховища даних у контексті предметної області, розробити фізичну модель сховища даних для заданої предметної області.

*Результати, їх новизна, теоретичне та практичне значення:* проведений аналіз існуючих моделей сховищ даних, проведене дослідження моделі темпоральної БД та анкерної моделі на відповідність їх заданій предметній області, розроблена логічна та фізична модель корпоративного сховищ даних для компанії з Інтернет-продажів, розроблені шість представлень БД та дві збережені процедури.

## **SUMMARY**

for a master's degree

‘Design and Development of a Corporate Data Warehouse’,  
Part 1. Modelling and Design of a Corporate Data Warehouse,  
student of Postolov Yurii

Qualifying master's thesis: 72 pages., 10 figures., 14 sources.

Keywords: RELATIONAL DATABASE, DATA WAREHOUSE,  
TEMPORAL DATABASES, SQL, MS SQL Server DBMS.

The aim of the research is to model a corporate data warehouse.

The object of the research is methods of building a model of a corporate data warehouse.

The subject of research is modeling a corporate data warehouse for a company with Internet sales.

Research objectives: analyze modern design solutions for building corporate data warehouses, perform modeling and design of a data warehouse in the context of a subject area, develop a physical model of a data warehouse for a given subject area.

Results, their novelty, theoretical and practical significance: the analysis of existing models of data warehouses was carried out, studies of the temporal database model and the anchor model were carried out for compliance with their given subject area, a logical and physical model of a corporate data warehouse for an Internet sales company was developed, six views DB were developed and two stored procedures.

## Передмова

Тема роботи: Моделювання та розробка корпоративного сховища даних

Метою роботи розробка моделі корпоративного сховища даних та фізична її реалізація.

Задачами даної роботи являлися аналіз існуючих моделей, методів та технологій створення корпоративного сховища даних; проектування та розробка корпоративного сховища даних компанії з Інтернет-продажів.

Частина 1: Моделювання та проектування корпоративного сховища даних

Виконавець: Постолов Юрій Іванович

Метою роботи є аналіз сучасного стану моделювання та проектування корпоративних сховищ даних.

Задачами даної роботи є аналіз моделей СД, зокрема темпоральної моделі та анкерної. Також до задач даної роботи входить проектування корпоративного сховища даних компанії з Інтернет-продажів за обраною моделлю.

Частина 2: Розробка корпоративного сховища даних

Виконавець: Постолова Ірина Василівна

Метою роботи є аналіз сучасних архітектур сховищ даних, та сучасного стану розробки корпоративних сховищ даних.

Задачею даної роботи є аналіз декількох архітектур сховищ даних, зокрема хмарних сховищ даних. Також до задач даної роботи входить розробка корпоративного сховища даних компанії з Інтернет-продажів.

Староста комплексної кваліфікаційної роботи \_\_\_\_\_ Постолов Ю.І.  
( підпис ) (прізвище та ініціали)



## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	11
ВСТУП .....	12
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ .....	15
2 МОДЕЛЮВАННЯ КОРПОРАТИВНОГО СХОВИЩА ДАНИХ .....	17
2.1 Корпоративна модель даних.....	17
2.2 Варіанти використання сховища даних .....	20
3 БАГАТОВИМІРНА МОДЕЛЬ СХОВИЩА ДАНИХ.....	23
3.1 Основні поняття методу багатовимірного моделювання .....	23
3.2 Багатовимірна модель .....	25
3.2.1 Факти.....	28
3.2.2 Ключі в таблицях фактів .....	29
3.2.3 Таблиці фактів.....	30
3.2.4 Виміри .....	32
3.3 Основні схеми багатовимірної моделі .....	33
4 МОДЕЛЮВАННЯ ТАБЛИЦЬ ВИМІРІВ .....	36
4.1 Повільно мінливі виміри.....	36
4.2 Швидко мінливі виміри .....	37
4.3 Темпоральні бази даних.....	38
5 АНКЕРНЕ МОДЕЛЮВАННЯ.....	40
6 ПРОЕКТУВАННЯ СХОВИЩА ДАНИХ КОМПАНІЇ .....	45
6.1 Доцільність використання темпоральної та анкерної моделей .....	45
6.2 Детальне проектування архітектури даних .....	46
6.3 Логічне проектування СД компанії з продажу через Інтернет.....	47
6.3.1 Сутності предметної області, що моделюється.....	50
6.3.2 Атрибути сутностей предметної області, що моделюється .....	52
6.3.3 Зв'язки між сутностями схеми .....	54
7 ВИБІР СКБД ДЛЯ РОЗРОБКИ СХОВИЩА ДАНИХ .....	56
7.1 Реляційні бази даних.....	56

7.1.1 Діалект мови SQL для СКБД SQL Server.....	57
7.1.2 Збережені процедури у мові T-SQL.....	59
7.2 Вибір СКБД для БД ІС «Особистий кабінет викладача».....	60
8 Фізичне проектування сховища даних.....	61
9 СЕРВЕРНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.....	67
9.1 Збережена процедура DepDoRequirement.....	67
ВИСНОВКИ.....	69
ПЕРЕЛІК ПОСИЛАНЬ.....	71
Додаток А. ЛОГІЧНА СХЕМА БАЗИ ДАНИХ ІС .....	73
Додаток Б. ПРЕДСТАВЛЕННЯ БАЗИ ДАНИХ ІС .....	74
Додаток В. ЗБЕРЕЖЕНІ ПРОЦЕДУРИ БД.....	76

## ПЕРЕЛІК СКОРОЧЕНЬ

БД база даних

БСД – багатовимірні сховища даних

ІС – інформаційна система

ПЗ – програмне забезпечення

РБД – реляційна база даних

РСД – реляційне СД, СД, що базується на реляційній моделі даних

СД – сховище даних

СКБД – система керування базами даних

## ВСТУП

У наш час продаж товарів через мережу Інтернет стає звичним. Епідеміологічна ситуація останнього року ще більше підвищила привабливість покупок через Інтернет.

Компанії, що використовують Інтернет для продажу своїх товарів, звичайно мають бази даних цих товарів. Але для оцінки ефективності бізнесу та перспектив розвитку компанії недостатньо мати лише транзакційну базу даних, яка зберігає інформацію про товари, їх ціну, поточні дані про їх продаж. Потрібно мати сховище даних (СД), в якому містяться як поточні, так і історичні дані, міститься вся інформація, яка дозволяє оцінювати ефективність бізнесу та перспектив його розвитку.

До середини 80-х рр. ХХ ст. практично повністю завершився перший етап оснащення бізнесу і державних структур засобами обчислювальної техніки і почався період бурхливого розвитку інформаційних систем для організації збору і зберігання великих масивів різного роду ділової та службової інформації. В основному це були корпоративні системи, призначені для оперативної обробки інформації, які обслуговували бухгалтерію, інформаційні архіви, телефонні мережі, реєстрацію документів, банківські операції і т.д. З появою персональних комп'ютерів такі системи стали доступними для безлічі дрібних і середніх фірм, підприємств і організацій. Системи оперативної обробки інформації отримали назву OLTP (On-Line Transaction Processing – оперативна, тобто в режимі реального часу, обробка транзакцій).

Типовим прикладом застосування OLTP-систем є масове обслуговування клієнтів, наприклад, бронювання авіаквитків або оплата послуг телефонних компаній. Обидві ці ситуації мають два загальних властивості: дуже велике число клієнтів і безперервне надходження інформації.

Такі транзакції виконуються тисячі разів в день в сотнях пунктів продажу. Очевидно, що основним пріоритетом в даному випадку є забезпечення мінімального часу відгуку при максимальному завантаженні системи.

Головна вимога до OLTP-систем – швидке обслуговування відносно простих запитів великої кількості користувачів, при цьому час очікування виконання типового запиту не повинно перевищувати кілька секунд. Згодом в таких системах почали акумулюватися великі обсяги даних – документи, відомості про банківські операції, інформація про клієнтів, укладених угодах, надані послуги тощо

Поступово виникло розуміння того, що збір даних не самоціль. Зібрана інформація може виявитися досить корисною в процесі управління організацією, пошуку шляхів вдосконалення діяльності та отримання за допомогою цього конкурентних переваг. Але для цього потрібні системи, які дозволяли б виконувати не тільки найпростіші дії над даними: підраховувати суми, середні, максимальні і мінімальні значення. З'явилася потреба в інформаційних системах, які дозволяли б проводити глибоку аналітичну обробку, для чого необхідно вирішувати такі завдання, як пошук прихованих структур і закономірностей в масивах даних, висновок з них правил, яким підпорядковується дана предметна область, стратегічне і оперативне планування, формування нерегламентованих запитів, прийняття рішень і прогнозування їх наслідків.

Ці завдання вирішуються у системах підтримки прийняття рішень, основою яких є корпоративне сховище даних.

Сховище даних – це різновид системи управління даними, яка забезпечує підтримку бізнес-аналітики. Сховища даних призначені тільки для виконання запитів і аналізу і зазвичай містять великі обсяги історичних даних. Дані зазвичай надходять в сховище з найрізноманітніших джерел, таких як журнали додатків і додатки транзакцій.

Сховище даних служить для централізації і консолідації великих обсягів даних з різних джерел. Аналітичні інструменти дають можливість організаціям отримувати від власних даних цінні для бізнесу відомості і підвищувати ефективність прийнятих рішень. Згодом в сховище накопичуються записи за минулі періоди, які становлять велику цінність для фахівців з ви-

вчення даних і бізнес-аналітиків. Ці можливості роблять сховища даних "єдиним джерелом перевірених даних компанії."

Метою даної комплексної магістерської роботи є моделювання, проектування та розробка сховища даних компанії по продажу товарів через мережу Інтернет.

Завданням даної частини кваліфікаційної роботи є моделювання та проектування корпоративного сховища даних.

## 1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

Проектування сховища даних для компанії слід почати з визначення конкретних бізнес-потреб, погодження сфери застосування та розробки концепції проекту [1]<sup>1)</sup>.

Найбільш важливим фактором при проектуванні є потреба кінцевих користувачів. Зазвичай користувачі використовують сховище для аналізу, і їм потрібні дані в узагальненому вигляді, а не у вигляді окремих транзакцій.

Для успішної роботи компанія по продажу товарів через Інтернет повинна мати не тільки підрозділи, що займаються власне продажем товарів. Потрібні також підрозділи, задачею яких є:

- робота з постачальниками товарів,
- облік клієнтів і їх вимог;
- маркетингові дослідження з просування товарів.

Таким чином, предметна область сховища даних повинна включати в себе щонайменше чотири предметні області окремих підрозділів компанії.

Також в сховище даних потрібно зберігати опис самих товарів. Опис товарів у сховище даних повинен дещо відрізнитись від опису цих товарів у базі даних, що використовується на сайті компанії для продажу товарів.

У сховищі даних не потрібно зберігати зображення товарів, але опис характеристик товарів бажано зберігати у максимальному об'ємі. Кількість характеристик товарів може значно покращити аналіз привабливості різних товарів однієї групи, дозволить виявити ті характеристики, які найчастіше значущі для покупців, та виявити закономірності формування у покупців критерію «ціна-якість» для різних товарів та груп товарів.

Деякі атрибути товарів можуть бути значущі в різній мірі для різних груп товарів. Тобто, для деяких груп товарів зберігання цих характеристик то-

---

<sup>1)</sup> [1] Эрик Спирли. Корпоративные хранилища данных. Планирование, разработка и реализация. Т.1. М.: Вильямс, 2001. – 400 стр.

варів призведе до зайвих витрат – витрат часу персоналу для внесення зайвих даних в базу даних, та витрат простору на диску для зберігання зайвої інформації.

Невдалим рішенням цієї проблеми буде дозвіл мати порожні значення атрибутам таблиці, що описують необов'язкові характеристики товару. При проектуванні сховища даних потрібно дотримуватись принципу, що жодні атрибути жодної таблиці не повинні мати порожніх значень (null). Такі атрибути потрібно зберігати у окремих похідних таблицях, зв'язаних з основною таблицею опису товару [2], [3]<sup>1)</sup>.

Для клієнтської бази даних також важливим є отримання щонайбільшої кількості можливих показників про клієнтів. Але збір цієї інформації обмежений. При реєстрації покупця він повинен заповнювати лише декілька обов'язкових полів, необов'язкові поля дуже часто залишаються порожніми. Але навіть велика кількість порожніх значень для необов'язкових атрибутів клієнта не може слугувати причиною відмови від використання них для опису клієнта та аналізу значень цих атрибутів на попит клієнта. Необов'язкові атрибути опису клієнта потрібно зберігати не в базовій таблиці «Клієнт», а в похідних таблицях атрибутів клієнта.

---

<sup>1)</sup> [2] Корпоративные хранилища данных. Интеграция систем. Проектная документация. URL: [https://www.prj-exp.ru/dwh/what\\_is\\_dwh.php](https://www.prj-exp.ru/dwh/what_is_dwh.php)

[3] Архипенко С., Голубев Д., Хранилища данных. От концепции до внедрения. М: Диалог-Мифи, 2002. – 528с.



## 2 МОДЕЛЮВАННЯ КОРПОРАТИВНОГО СХОВИЩА ДАНИХ

Ядром будь-якого сховища даних є його модель даних. Без моделі даних буде дуже складно організувати дані в СД. Тому розробники сховища даних повинні витратити час і сили на розробку такої моделі [4]<sup>1)</sup>.

У порівнянні з проектуванням OLTP-систем, методика проектування сховища даних має ряд відмінних рис, пов'язаних з орієнтацією структур даних сховища на вирішення завдань аналізу та інформаційної підтримки процесу прийняття рішень. Модель даних СД повинна забезпечувати ефективне рішення саме цих завдань.

Відправною точкою в проектуванні СД може служити так звана корпоративна модель даних (corporate data model або enterprise data model, EDM), яка створюється в процесі проектування OLTP-систем організації. При проектуванні корпоративної моделі даних зазвичай робиться спроба створити на основі бізнес-операцій таку структуру даних, яка б збила і синтезувала в собі всі інформаційні потреби організації .

### 2.1 Корпоративна модель даних

Під корпоративною моделлю даних розуміють багаторівневе, структуроване опис предметних областей організації, структур даних предметних областей, бізнес-процесів і бізнес-процедур, потоків даних, прийнятих в організації, діаграм станів, матриць "дані-процес" та інших модельних уявлень, які використовуються в діяльності організації

Таким чином, в широкому сенсі слова, корпоративна модель даних являє собою сукупність моделей різного рівня, які характеризують (моделюють на деякому абстрактному рівні) діяльність організації, тобто зміст корпора-

---

<sup>1)</sup> [4] Куликов Г. Г., Никулина Н. О., Речкалов А. В. Управление проектами на основе системного моделирования: Учебное пособие. Уфа: УГАТУ, 2009. – 171 с.

тивної моделі безпосередньо залежить від того, які модельні конструкції були включені в неї в даній організації.

Корпоративна модель даних містить сутності, атрибути та відношення, які представляють інформаційні потреби організації [5]<sup>1)</sup>.

Корпоративна модель даних поділяється у відповідності до предметних областей, які представляють групи сутностей, що відносяться до підтримки конкретних потреб бізнесу. Деякі предметні області можуть покривати такі специфічні бізнес-функції, як управління контрактами, інші – об'єднувати сутності, що описують продукти або послуги.

Основними елементами корпоративної моделі даних є:

- опис предметних областей організації (визначення сфер діяльності);
- взаємини між певними вище предметними областями;
- інформаційна модель даних (ERD-модель або модель "сутність-зв'язок");
- функціональна модель або модель бізнес-процесів;
- діаграми потоків даних;
- діаграми станів;
- інші моделі.

Для кожної предметної області потрібно виконати опис:

- ключів сутностей;
- атрибутів сутностей;
- підтипів і супертіпа;
- зв'язків між сутностями;
- угруповання атрибутів;
- взаємозв'язків між предметними областями;

Кожна логічна модель повинна відповідати існуючій предметній області корпоративної моделі даних. Якщо логічна модель не відповідає даній ви-

---

<sup>1)</sup> [5] Корпоративная модель данных. Лекции НОУ ИНТУИТ URL: <https://intuit.ru/studies/courses/599/455/lecture/10176>

мозі, в неї повинна бути додана модель, яка визначає предметну область, як зображено на рис. 1.



Рисунок 1 – Основні елементи корпоративної моделі даних

Корпоративна модель даних зазвичай має кілька рівнів подання. На найвищому рівні (high level) корпоративної моделі даних розташовується опис основних предметних областей організації та їх взаємозв'язків на рівні сутностей. На рис. 2 наведено фрагмент корпоративної моделі даних верхнього рівня.

На схемі, наведеній на рис. 2, представлено чотири предметних області: "Покупець" (Customer), "Рахунок" (account), "Замовлення" (Order) і "Товар" (Product). Як правило, на верхньому рівні представлення моделі вказуються тільки прямі зв'язки між предметними областями, які, наприклад, фіксують наступний факт: покупець оплачує рахунок на замовлення товарів. Детальна

інформація та непрямі взаємозв'язки на цьому рівні корпоративної моделі не наводяться.

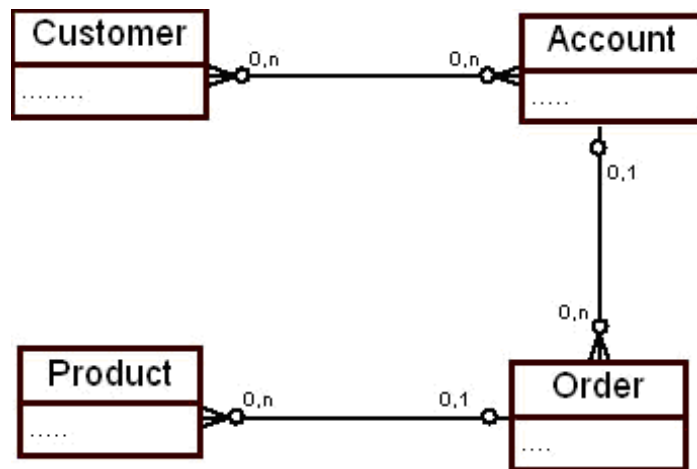


Рисунок 2 – Фрагмент корпоративної моделі даних верхнього рівня

На наступному, середньому рівні (mid level) корпоративної моделі даних показується докладна інформація про об'єкти предметних областей, тобто ключі і атрибути сутностей, їх взаємозв'язки, підтипи і супертипи, тощо.

## 2.2 Варіанти використання сховища даних

Для визначення атрибутів сутностей, значимих в контексті даної предметної області, потрібно визначити варіанти використання сховища даних при роботі з ним менеджерів і аналітиків компанії.

Розглянемо варіанти використання СД, які можуть бути потрібними для якоїсь групи користувачів:

- 1) Вибрати продукти, що дають найбільший прибуток.
- 2) Визначити комбінацію товарів, що просуває бізнес.
- 3) Визначити сегменти ринку, що дають найбільший прибуток.
- 4) Визначити клієнтів, що забезпечують найбільший прибуток.
- 5) Визначити набір продуктів, що купують найприбутковіші клієнти.

- 6) Визначити властивості, які характерні клієнтам, що забезпечують найбільший прибуток.
- 7) Визначити постачальників, що пропонують якнайкраще співвідношення ціна/якість?
- 8) Визначити вартість проведення маркетингової кампанії через заданий канал.
- 9) Визначити найефективніший канал проведення маркетингових кампаній.
- 10) Визначити, наскільки клієнти задоволені якістю товарів.
- 11) Визначити, як задоволеність клієнтів змінюється з часом.
- 12) Визначити, які продукти постачають вчасно, які - із запізненням.
- 13) Визначити, чи мають певні клієнти або продукти неприпустимо довгий термін постачання?
- 14) Визначити основні причини відмови від продукту.

На рис. 3 наведена діаграма використання корпоративного сховища даних користувачами груп «Аналітик» та «Менеджер середньої ланки».

Як видно з варіантів використання сховища даних менеджерами та аналітиками компанії, для відповіді на їх питання потрібні не деталізовані, а агреговані дані. Агреговані дані у базах даних не зберігаються, їх отримуються на вимогу користувача за допомогою запитів з групуванням даних, тобто вони є віртуальними даними.

На відміну від транзакційних баз даних користувачам сховищ даних потрібно саме агреговані дані. Тому у сховищах даних поряд з деталізованими даними зберігаються агреговані дані різного ступеня агрегації. Таблиці сховища даних не знаходяться у нормалізованому виді.

При проектуванні сховища даних потрібно виявити, які саме агрегати потрібні для основних запитів користувачів, і створити таблиці агрегованих даних. Отримувати агрегати під час виконання запиту користувача неприпустимо, оскільки у цьому випадку запити будуть виконуватись не достатньо швидко.

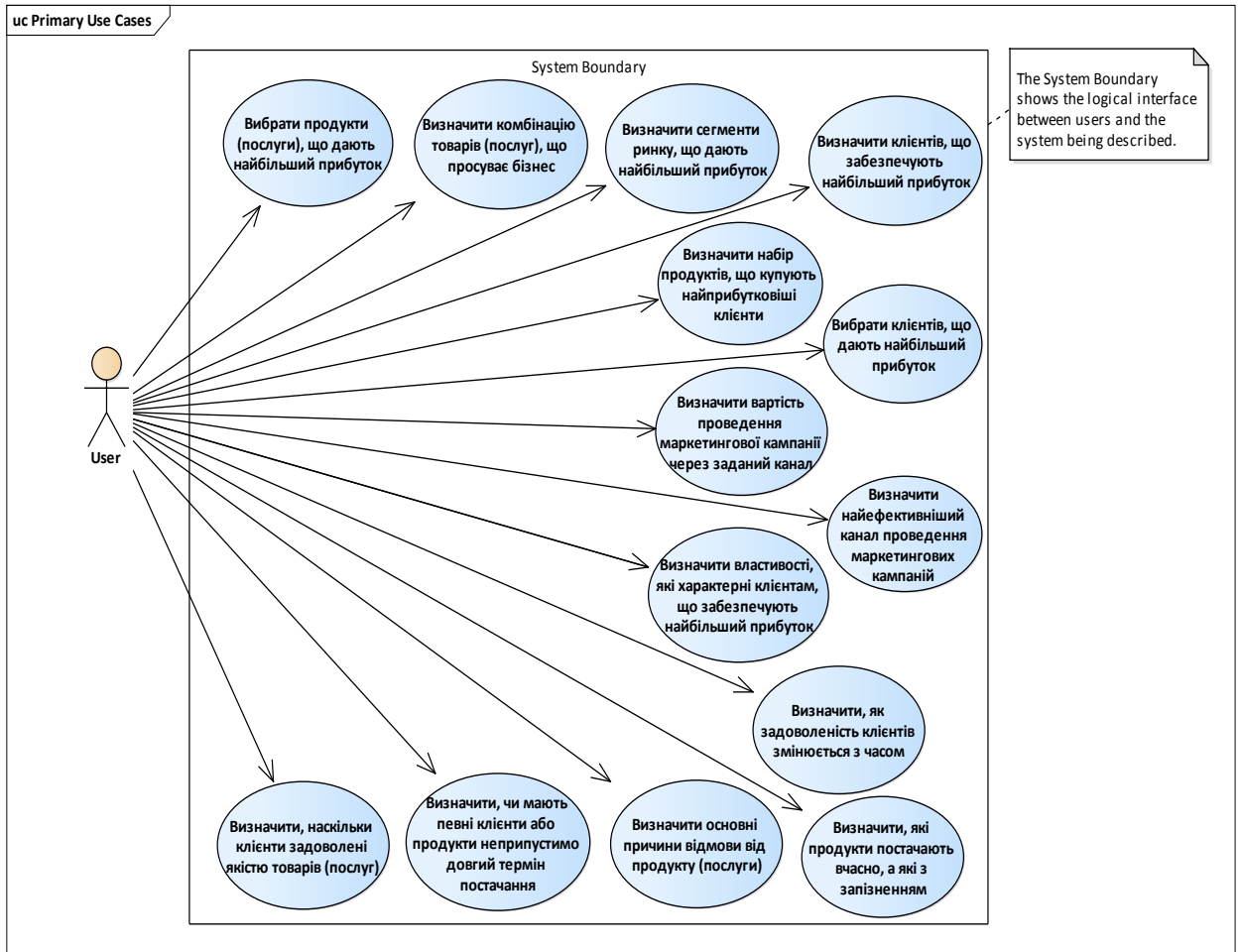


Рисунок 3 – Діаграма варіантів використання корпоративного СД

## 3 БАГАТОВИМІРНА МОДЕЛЬ СХОВИЩА ДАНИХ

Багатовимірне моделювання є методом моделювання і візуалізації даних як множини числових або лінгвістичних показників або параметрів (measures), які описують загальні аспекти діяльності організації. Як правило, при багатовимірному моделюванні основна увага фокусується на числових даних, таких як число продажів, баланс, прибуток, вага, або на об'єктах, які можна перерахувати, таких як статті, патенти, книги [6],[7]<sup>1)</sup>.

### 3.1 Основні поняття методу багатовимірного моделювання

Метод багатовимірного моделювання базується на таких основних поняттях: факти, атрибути, вимірювання, параметри (метрики), ієрархія, гранулювання.

Факт (fact) – це набір пов'язаних елементів даних, що містять метрики і описові дані. Кожен факт зазвичай представляє елемент даних, чисельно описує діяльність організації, бізнес-операцію або подію, яке може бути використане для аналізу діяльності організації або бізнес-процесів. В СД факти зберігаються в базових таблицях реляційної БД. Наприклад, вартість товару, кількість одиниць товару і т.д.

Атрибут (Attribute) – це опис характеристики реального об'єкта предметної області. Як правило, атрибут містить заздалегідь відоме значення, що характеризує факт. Зазвичай атрибути представляються текстовими полями з дискретними значеннями. Наприклад, габарити упаковки товару, запах товару.

---

<sup>1)</sup> [6] Шаламова Н.Г., Ивашечкова М.С. Определение стратегии развития организации с использованием подходов и методов многомерного моделирования. Вестник Университета №11, 2012. С.158 – 164.

[7] Метод многомерного моделирования. Лекции НОУ ИНТУИТ URL: <https://intuit.ru/studies/courses/599/455/lecture/10167>

Вимірювання (dimension) – це інтерпретація факту з деякою точки зору в реальному світі. Вимірювання, подібно атрибутам, містять текстові значення, які сильно пов'язані за змістом між собою. Зазвичай вимірювання представляються як осі багатовимірному простору, точками якого є пов'язані з ними факти. У багатовимірній моделі кожен факт пов'язаний з однією або декількома осями. Вимірювання зазвичай представляють нечислові, лінгвістичні змінні, такі як філії організації, співробітники організації, покупці, тощо.

Наприклад, при аналізі продажів продукції, виробленої або продається організацією, такими вимірами зазвичай вступають час, покупці, продавці, місце продажу або складування товару.

Вимірювання задаються перерахуванням своїх елементів (members). Елемент вимірювання (dimensional member) – унікальне ім'я або ідентифікатор (лінгвістична змінна), яка використовується для визначення позиції елемента. Наприклад, вимір "Час" може містити такі елементи: "всі місяці", "квартали", "роки".

Часто елементи вимірювання перебувають у відношенні "частина-ціле" або "батько-нащадок", що дозволяє ввести на вимірі одну або кілька ієрархій. Кожна ієрархія може мати кілька рівнів ієрархії (hierarchy levels). Кожен елемент вимірювання повинен належати тільки одному рівню ієрархії, породжуючи таким чином розбиття на підмножини. Прикладом може служити ієрархія на вимірі "Час": рік, півріччя, квартали, місяці і дні. Елемент вимірювання "тиждень" може належати двом місяцям, тому для нього слід визначити іншу ієрархію.

Параметр, метрика або показник (measure) – це числова характеристика факту, який визначає ефективність діяльності або бізнес-дії організації з точки зору вимірювання. Як правило, метрика містить заздалегідь не відоме значення характеристики факту. Конкретні значення метрики описуються за допомогою змінних. Наприклад, нехай метрикою є чисельне вираження продажів товару в грошах, кількість проданих одиниць товару і т.д. Метрика ви-



значається за допомогою комбінації елементів вимірювання і, таким чином, являє факт.

Гранулювання (*Granularity*) – це рівень деталізації даних, що зберігаються в СД. Наприклад, щоденні обсяги продажів.

### **3.2 Багатовимірна модель**

Багатовимірна модель даних, що лежить в основі побудови багатовимірних сховищ даних, спирається на концепцію багатовимірних кубів, або гіперкубів. Вони являють собою впорядковані багатовимірні масиви, які також часто називають OLAP-кубами (аббревіатура OLAP розшифровується як *On-Line Analytical Processing* – оперативна аналітична обробка). Технологія OLAP є методику оперативного вилучення потрібної інформації з великих масивів даних і формування відповідних звітів.

Сутність багатовимірного представлення даних полягає в наступному. Більшість реальних бізнес-процесів описується множиною показників, властивостей, атрибутів і т.д. Наприклад, для опису процесу продажів можуть знадобитися відомості про найменування товарів або їх груп, про постачальника і покупця, про місто, де проводилися продажі, а також про ціни, кількості проданих товарів і загальні суми. Крім того, для відстеження процесу в часі повинен бути введений в розгляд такий атрибут, як дата.

Якщо зібрати всю цю інформацію в таблицю, то вона виявиться складною для візуального аналізу і осмислення. Більш того, вона може виявитися надмірною: якщо, наприклад, один і той же товар продавався в один і той же день в різних містах, то доведеться кілька разів повторити одну й ту саму відповідність «місто – товар» із зазначенням різних суми і кількості. Все це здатне остаточно заплутати і збити з пантелику будь-якого, хто спробує витягти з такої таблиці корисну інформацію з метою аналізу поточного стану продажів і пошуку шляхів оптимізації процесу торгівлі. Зазначені проблеми

виникають з однієї простої причини: в плоскій таблиці зберігаються багатовимірні дані.

Багатовимірна модель візуально представляється за допомогою куба (або в разі більше трьох вимірів – гіперкуба). Розглянемо приклад. Нехай обсяг продажів торгової організації є функція від змінних "Товари", "Місяць" і "Регіон продажів". Тоді як вимірювань будуть виступати "Товари", "Час" і "Місце розташування". На рис. 4 наведено багатовимірний куб даних для подання даної функції.

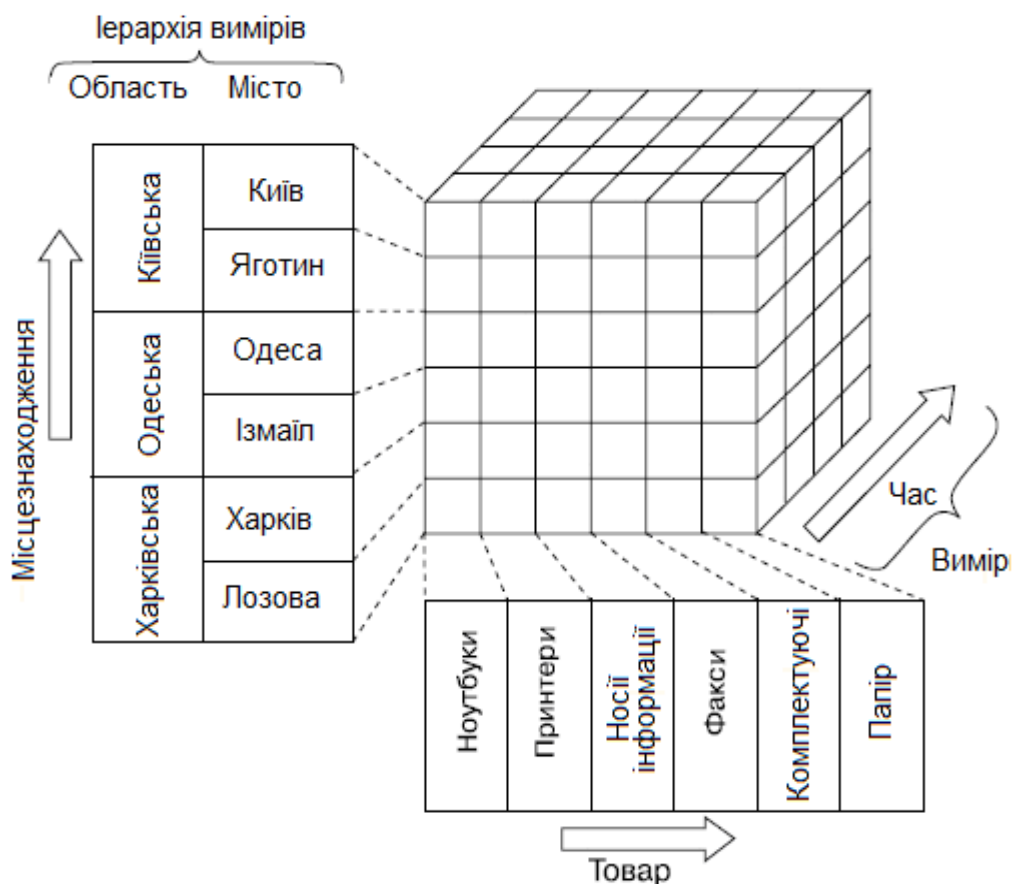


Рисунок 4 – Куб даних

На цих вимірах можуть бути задані наступні ієрархії:

- вимір «Товари» – «Виробник-Категорія-Товар»;
- вимір «Місцезнаходження» – «Область-Місто-Магазин»;
- вимір «Час» – «Рік-Квартал-Місяць» або «Тиждень-День».

Багатовимірне моделювання є основним методом логічного проектування СД для OLAP-додатків. Для таких додатків типово виконання операцій згортання і розгортання даних.

Таким чином, інформація в багатовимірному сховищі даних є логічно цілісною. Це вже не просто набори строкових і числових значень, які в разі реляційної моделі потрібно отримувати з різних таблиць, а цілісні структури типу «кому, що і в якій кількості було продано на даний момент часу». Переваги багатовимірного підходу очевидні.

- Представлення даних у вигляді багатовимірних кубів наочніше, ніж сукупність нормалізованих таблиць реляційної моделі, структуру якої представляє тільки адміністратор БД.
- Можливості побудови аналітичних запитів до системи, що використовує БСД, більш широкі.
- У деяких випадках використання багатовимірної моделі дозволяє значно зменшити тривалість пошуку в БСД, забезпечуючи виконання аналітичних запитів практично в режимі реального часу. Це пов'язано з тим, що агреговані дані обчислюються попередньо і зберігаються в багатовимірних кубах разом з деталізованими, тому витрачати час на обчислення агрегатів при виконанні запиту вже не потрібно.

В принципі, OLAP-куб може бути реалізований і за допомогою звичайної реляційної моделі. У цьому випадку має місце емуляція багатовимірного представлення сукупністю плоских таблиць. Такі системи отримали назву ROLAP – Relational OLAP.

Використання багатовимірної моделі даних пов'язане з певними труднощами. Так, для її реалізації потрібно більший обсяг пам'яті. Це пов'язано з тим, що при реалізації фізичної багатовимірності використовується велика кількість технічної інформації, тому обсяг даних, який може підтримуватися багатовимірні сховища даних (БСД), зазвичай не перевищує кількох десятків гігабайт. Крім того, багатовимірна структура важче піддається модифікації;

при необхідності вбудувати ще один вимір потрібно виконати фізичну перебудову всього багатовимірного куба. На підставі цього можна зробити висновки, що застосування систем зберігання, в основі яких лежить багатовимірне представлення даних, доцільно тільки в тих випадках, коли обсяг використовуваних даних порівняно невеликий, а сама багатовимірна модель має стабільний набір вимірювань.

### 3.2.1 Факти

З точки зору взаємозв'язку вимірів і фактів останні можна розбити на наступні класи:

- адитивні факти (параметри) (Additive facts). Факт називається адитивним, якщо його має сенс використовувати з будь-якими вимірами для виконання операцій додавання з метою отримання будь-якого значимого результату. Наприклад, має сенс підсумовувати загальний обсяг продажу для продукту, місцезнаходження і часу, оскільки це не викликає накладання серед явищ реального світу, які генерують кожне з цих значень;
- напівадитивні факти (Semiadditive facts). Факт називається напівадитивним, якщо його має сенс використовувати спільно з деякими вимірами для виконання операцій додавання з метою отримання будь-якого значимого результату. Наприклад, підсумовування запасів за різними товарами і складами має сенс, але підсумовування запасів товарів у різний час недоцільне, оскільки одне і те ж фізичне явище може враховуватися кілька разів;
- неадитивні факти (Non-additive facts). Факт називається неадитивним, якщо його не має сенсу використовувати спільно з будь-яким виміром для виконання операцій додавання з метою отримання будь-якого значимого результату. Неадитивні факти не комбінуються в будь-якому вимірі зазвичай тому, що вибрана формула не дозволяє

об'єднати середні значення низького рівня в середньому значенні вищого рівня. Наприклад, вимір кімнатної температури;

- числові заходи інтенсивності (Numerical Measures of Intensity). Факт називається числовий мірою інтенсивності, якщо він, будучи неадитивним за часом, допускає агрегацію і підсумовування по деякому числу часових періодів. Наприклад, залишок на рахунку.

Розглянемо приклади неадитивних фактів. Факти, що представляють атрибути тексту, є неадитивними. Підсумувати такі факти сенсу не має. Для таких фактів має сенс підрахунок їх кількості в таблиці фактів. Ціна одиниці товару також є неадитивним фактом. Не має сенсу підсумовувати ціну клавіатури або собівартість одиниці продукції. Однак якщо зберігається добуток вартості одиниці товару на кількість проданого товару, то ця величина буде вже адитивним фактом.

Відсотки і відносини величин (валовий прибуток) є неадитивними фактами. Можна зберігати як окремі параметри чисельник і знаменник відносини, коли їх роздільне підсумовування має сенс. І це будуть вже адитивні факти. До неадитивних фактів відносяться також статистичні середні суми, такі як, наприклад, середня температура за день. Сума середніх денних температур за тиждень не має ніякого сенсу.

### **3.2.2 Ключі в таблицях фактів**

Первинний ключ в таблиці фактів є, як правило, складовим первинним ключем. Він складається з множини зовнішніх ключів, які є первинними ключами вимірювань, пов'язаних з фактами.

Слід зауважити, що іноді кілька комбінацій зовнішніх ключів представляють унікальний первинний ключ таблиці фактів. А з іншого боку, не завжди комбінація всіх зовнішніх ключів вимірювань гарантує унікальність первинного ключа таблиці фактів.

Розглянемо докладніше питання унікальності первинного ключа таблиці фактів.

Факти, як правило, гранульовані. Гранулювання фактів визначає семантичний сенс значення факту з точки зору рівня деталізації, яку пов'язують з фактом інформації. Наприклад, загальний обсяг продажів по даному магазину на вказаний день по даному виду товару.

Часто гранулювання визначають на рівні окремої бізнес-операції. Наприклад, продаж товару. Іноді потрібно зберігати інформацію про продаж даного товару за день (тут вже присутній деякий рівень агрегації даних).

Унікальність первинного ключа таблиці фактів гарантується визначенням гранулювання фактів, а не умовою унікальності зовнішніх ключів від таблиць вимірів, якщо гранулювання фактів не визначене належним чином.

### **3.2.3 Таблиці фактів**

Факти в багатовимірної моделі прийнято представляти у вигляді таблиці фактів. У логічної моделі «сутність-зв'язок» таблиця фактів представляється сутністю, атрибутами якої є факти (метрики або опису) і складовою ключ, що зв'язує таблицю фактів з таблицями вимірювань взаємозв'язком «один до багатьох».

Таблиці фактів поділяють на три основні категорії, в залежності від рівня деталізації фактів (гранулювання).

- Транзакційна таблиця фактів. У такій таблиці фактів зберігають факти, які фіксують певні події (транзакції). Це факти, що описують кожен подію бізнесу. Наприклад, продаж товару.
- Таблиця фактів періодичних моментальних знімків У такій таблиці збирають факти, здатні фіксувати поточний стан певного напрямку бізнесу. Це факти, які описують поточний стан певного напрямку бізнесу для будь-якої комбінації значень вимірів за даний період часу. Наприклад, продаж організації на певну дату (щодня).
- Таблиця фактів кумулятивних моментальних знімків У такій таблиці збирають факти, здатні фіксувати деякий підсумковий стан певного напрямку бізнесу на поточний момент часу. Це факти, які описують

проміжні підсумки діяльності організації за певним напрямом бізнесу для будь-якої комбінації значень вимірів за даний період часу. Наприклад, продажі цього року на певну дату.

Основними характеристиками таблиці фактів є такі.

1. Таблиця фактів містить числові параметри (метрики).
2. Кожна таблиця фактів має складовою ключ, що складається з первинних ключів таблиць вимірів. Первинний ключ таблиці вимірювань є зовнішнім ключем в таблиці фактів.
3. Таблиця фактів має, як правило, невелика кількість полів, не більше 20-ти.
4. Дані в таблиці фактів мають такі властивості:
  - числові параметри використовуються для агрегації і підсумовування;
  - значення даних повинні мати властивості адитивності або полуадитивності по відношенню до вимірювань, для того щоб їх можна було підсумувати;
  - всі дані таблиці фактів повинні бути однозначно ідентифіковані через ключі таблиць вимірів, щоб забезпечити доступ до них через таблиці вимірювань.

Таким чином, таблицю фактів можна розділити на дві частини. Перша частина складається з первинних ключів вимірювань, а друга – з числових параметрів функціонально залежать від ключів таблиць вимірів. Часто таблиця фактів має сурогатний первинний ключ, згенерований системою, від якого за визначенням функціонально повно залежать всі параметри.

Приклад таблиці фактів наведено на рис. 5.

В таблиці фактів «Продажі» перші 3 атрибути є зовнішніми ключами, що зв'язують цю таблицю з таблицями вимірів «Час», «Місцезнаходження» та «Товар» відповідно.

У сукупності ці три зовнішні ключа утворюють первинний ключ таблиці фактів «Продажі».

Продажі
<u>Ідентифікатор часу</u>
<u>Ідентифікатор місцезнаходження</u>
<u>Ідентифікатор товару</u>
Кількість покупців
Кількість продажів
Сумарний прибуток

Рисунок 5 – Приклад таблиці фактів

### 3.2.4 Виміри

Основними характеристиками таблиці вимірювань є наступні:

1. Таблиці вимірювань містять дані про деталізацію фактів.
2. Таблиці вимірювань містять описову інформацію про числових значеннях в таблиці фактів, тобто вони містять атрибути фактів.
3. Як правило, денормалізовані таблиці вимірювань містять велику кількість полів.
4. Таблиці вимірювань містять зазвичай значно менше рядків, ніж таблиці фактів.
5. Атрибути таблиць вимірів зазвичай використовуються при візуалізації даних у звітах і запитах.

На рис. 6 наведено приклад таблиці виміру «Час». В цій таблиці атрибут «Ідентифікатор часу» є первинним ключем таблиці вимірювань. Решта поля є атрибутами параметрів таблиці фактів, що залежать від часу.

Час
<u>Ідентифікатор часу</u>
День тижня
День місяця
Місяць
Квартал
Рік
Звітний період
Відмітка свята

Рисунок 6 – Приклад таблиці виміру



### 3.3 Основні схеми багатовимірної моделі

Застосування реляційної моделі при створенні СД в ряді випадків дозволяє отримати переваги над багатовимірною технологією, особливо в частині ефективності роботи з великими масивами даних і використання пам'яті комп'ютера. На основі реляційних сховищ даних (РСД) будуються ROLAP-системи, і ця ідея належить Кодду.

В основі технології РСД лежить принцип, відповідно до якого вимірювання зберігаються в плоских таблицях так само, як і в звичайних реляційних СУБД, а факти (дані, що агрегуються) – в окремих спеціальних таблицях цієї ж бази даних. При цьому таблиця фактів є основою для пов'язаних з нею таблиць вимірів. Вона містить кількісні характеристики об'єктів і подій, сукупність яких передбачається в подальшому аналізувати.

На логічному рівні розрізняють дві схеми побудови РСД – «зірка» і «сніжинка». При використанні схеми «зірка» центральною є таблиця фактів, з якою пов'язані всі таблиці вимірювань. Таким чином, інформація про кожен вимір розташовується в окремій таблиці, що спрощує їх перегляд, а саму схему робить логічно прозорою і зрозумілою користувачеві ( рис. 7).

Однак розміщення всієї інформації про вимірювання в одній таблиці виявляється не завжди виправданим. Наприклад, якщо товари, що продаються, об'єднані в групи (має місце ієрархія), то доведеться в той чи інший спосіб показати, до якої групи належить кожен товар, що призведе до багаторазового повторення назв груп. Це не тільки викличе зростання надмірності, але і підвищить ймовірність виникнення протиріч (якщо, наприклад, один і той же товар помилково віднесуть до різних груп).

Для більш ефективної роботи з ієрархічними вимірами була розроблена модифікація схеми «зірка», яка отримала назву «сніжинка». Головною особливістю схеми «сніжинка» є те, що інформація про один вимір може зберігатися в декількох зв'язаних таблицях ( рис. 8).

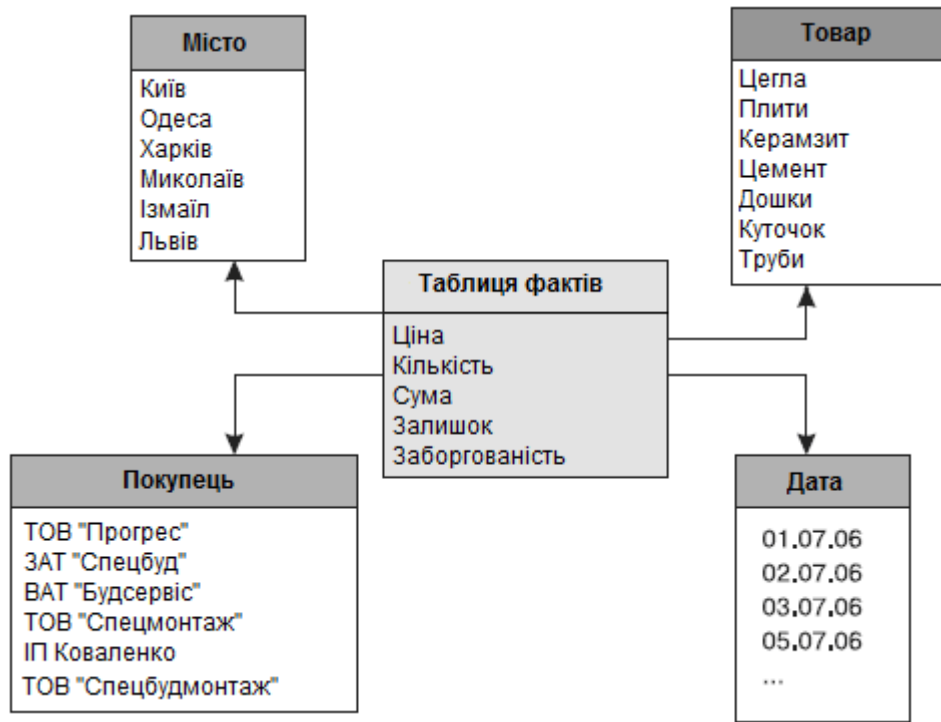


Рисунок 7 – Приклад схеми «зірка»

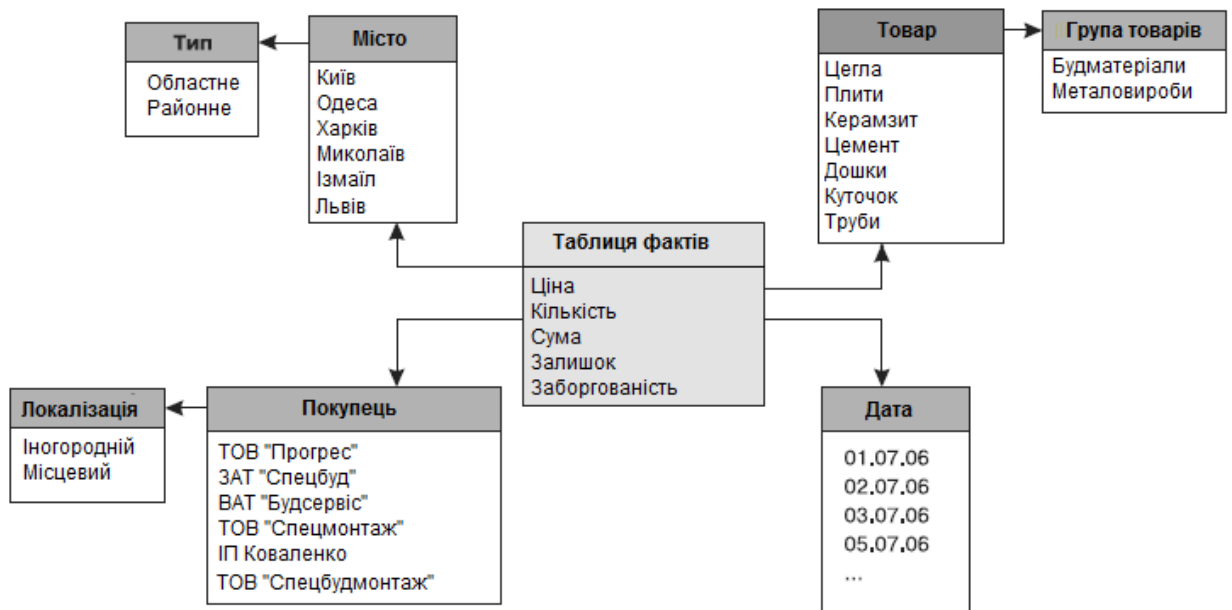


Рисунок 8 – Приклад схеми «сніжинка»

На рис. 7 – 8 зображені концептуальні моделі, на яких відсутні первинні та зовнішні ключі. Щоб отримати логічну модель потрібно додати ключі:

- у таблицю «Місто» – первинний ключ «ідентифікатор міста»;

- у таблицю «Товар» – первинний ключ «ідентифікатор товару»;
- у таблицю «Дата» – первинний ключ «ідентифікатор дати»;
- у таблицю «Покупець» додати первинний ключ «ідентифікатор покупця»;
- у таблицю фактів додати у якості зовнішніх ключів всі 4 ідентифікатори, які у сукупності будуть первинним ключем таблиці фактів.

Вимір «Час» є одним з критичних елементів моделі СД. Якщо дані в OLTP-системах запити фокусуються на поточному моменті часу, то в системах підтримки прийняття рішень, для яких проектується і створюються СД, запити фокусуються на задачах аналізу даних, а саме – як дані змінювалися в різні періоди часу. Наприклад, який був обсяг продажів торгової компанії за останній квартал, місяць або які тенденції в покупках товарів протягом останнього кварталу.

Вимірювання «Місто» дозволяє згрупувати операції продажів по містах країни з урахуванням їх географічного положення.

Вимірювання «Товари» дозволяє аналізувати типові схеми закупівель товарів і відповідати на питання, які товари, як правило, купуються одночасно покупцями.

Вимірювання «Покупці» дозволяє аналізувати покупки з урахуванням їх частоти, географічного положення та кількості.

Таблиці фактів мають велику кількість рядків. Таблиці вимірювань, як правило, мають набагато менше число рядків. Одним з головних переваг використання такої схеми є те, що продуктивність операції СУБД з'єднання таблиці фактів і таких таблиць з'єднань буде близька до оптимальної.

## 4 МОДЕЛЮВАННЯ ТАБЛИЦЬ ВИМІРІВ

Значення полів таблиць вимірів може змінюватися з часом. Такі зміни призводять до проблем в проектуванні таблиць вимірів. Це пов'язано з тим, що за визначенням та призначенням сховища даних призначені для зберігання історичних даних і надання користувачам доступу до них. Виміри, що змінюються, умовно можна розділити на дві групи: повільно мінливі дані та швидко мінливі дані.

### 4.1 Повільно мінливі виміри

Повільно мінливими вимірами (Slowly Changing Dimensions) називаються таблиці вимірювань, в яких деякі атрибути можуть змінити свої значення після закінчення деякого періоду часу, причому частота таких змін є невеликою.

Як приклад розглянемо таблиці вимірювань "Покупець" або "Товари". Припустимо, що первинні ключі таблиць не змінюються, а атрибути таблиці вимірювань можуть змінюватися з часом. В такому випадку виділяють три типи дій.

- Тип 1. Змінити значення атрибута таблиці вимірювань з новим значенням. При цьому буде втрачена хронологія.
- Тип 2. Створити новий рядок в таблиці вимірювань з новим значенням сурогатного ключа; у новому рядку буде нове значення первинного ключа таблиці, нове значення змінного атрибута, та копії інших атрибутів.
- Тип 3. Створити додатковий атрибут таблиці вимірювань з новим значенням та з атрибутом, що фіксує час, коли відбулась зміна значення атрибуту.

Для першого випадку втрачається попереднє значення атрибуту таблиці вимірів. Наприклад, у виміру «Покупець» є атрибут «Сімейний стан». Цей

атрибут може використовуватись для аналізу, як сімейний стан покупців впливає на попит за різними групами товарів та на обсяги покупок. Покупці з часом можуть одружуватись, розводитись, становитись вдівцем/вдовою.

Якщо проектувати сховище даних за типом 1, історія зміни сімейного стану буде втрачена і описаний аналіз попиту покупців буде некоректним.

Якщо проектувати сховище даних за типом 2, історія зміни сімейного стану не буде втрачена тільки у випадку, якщо буде додаткова таблиця історії змін ідентифікаторів таблиць вимірів. Але і в цьому випадку схема буде неефективна, тому що буде присутня зайва надмірність даних – будуть зберігатись копії значень всіх атрибутів, які не змінювались у часі.

При проектуванні СД за типом 3, історія зміни сімейного стану покупця зберігається, якщо ця зміна відбулась лише один раз за досить тривалий період часу. Якщо ці зміни торкнулись хоча б одного покупця декілька разів, ця схема працювати не буде. Крім того, достатня кількість покупців може на протязі досить тривалого періоду часу не змінювати показник свого сімейного стану, в записах цих покупців два атрибути будуть мати порожнє значення (null). Наявність порожніх значень є ознакою погано спроектованого СД.

## **4.2 Швидко мінливі виміри**

Мінливими вимірами (Rapidly Changing Dimensions) називаються таблиці вимірювань, в яких деякі атрибути можуть часто міняти свої значення в короткі періоди часу.

Моделі для управління такими вимірами залежать від кардинальності таблиць вимірів. Якщо кардинальність таблиць вимірів є невеликий (до 10000 записів), то може бути використаний такий же підхід, як у випадку повільно мінливих вимірювань.

У разі дуже великих таблиць вимірів (до мільйона записів) слід уникаати дублювання записів і не створювати нові додаткові записи. Оскільки не-

обхідно пам'ятати про продуктивність виконання запитів з такими вимірами, підходи до моделювання таких вимірювань є важливими.

Перераховані підходи не є оптимальними навіть для повільно мінливих вимірів. Для швидко мінливих вимірів використання будь-якого підходу з перерахованих 3 типів може викликати суттєві проблеми при використанні сховища даних.

### 4.3 Темпоральні бази даних

Можна навести багато прикладів бізнес-додатків, які використовують дані, що залежать від часу. Наприклад, фінансові програми, інформаційні системи управління готелями, наукові додатки (моніторинг навколишнього середовища) оперують даними, які змінюються в часі. Список може бути продовжений. СКБД в таких системах управляють даними, які посилаються на час, і отже, час зв'язується з сутностями, які зберігаються в базах даних.

Що ж таке темпоральні дані? У дуже широкому сенсі – це довільні дані, які явно або неявно пов'язані з певними датами або проміжками часу. Під таке визначення потрапляють майже будь-які дані та інформація. Наприклад, навіть якщо немає явної залежності від часу для будь-якого факту або події, то все одно для нього є неявна залежність від часу, так як колись нам (або системі) стало відомо, що такий факт існує. Крім того, є ймовірність, що в майбутньому факт буде переглянутий або на умови його використання будуть накладені певні обмеження; тому не можна вже буде розглядати його як деяку абсолютну істину, вірну у всіх ситуаціях і в будь-який момент часу [8]<sup>1)</sup>.

Темпоральні бази даних – це бази даних, що зберігають темпоральні дані. Такі БД і містяться в них дані можуть розглядатися як темпоральні тільки в тому випадку, якщо відомо правило інтерпретації часових міток і інтервалів для конкретної системи керування базами даних (СКБД). Щоб визначи-

---

<sup>1)</sup> [8] Темпоральные данные и базы данных. URL: <https://intuit.ru/studies/courses/599/455/lecture/10165>.

ти, чи є розглянута СКБД темпоральною в повному сенсі цього слова, необхідно зрозуміти, чи можна окремо виділити і спеціальним чином інтерпретувати дані атрибута «час». У категорію темпоральних СКБД не потраплятимуть звичайні реляційні СКБД, в яких підтримуються пов'язані з часом типи даних, але інтерпретацією і зв'язком даних (або подій) між собою з урахуванням часу доводиться займатися розробнику. У темпоральній СКБД враховуються специфічна природа часу і мінливість даних з плином часу.

Для мінливих вимірів, значення атрибутів яких можуть досить швидко змінюватись, а також для вимірів, у яких може змінюватись кількість атрибутів, потрібен новий підхід для проектування та експлуатації сховища даних.

Таким відносно ним підходом к проектуванню сховищ даних є анкерне (якірне) моделювання.

## 5 АНКЕРНЕ МОДЕЛЮВАННЯ

Анкерне (Anchor) моделювання – це гнучкий метод моделювання бази даних, що підходить для інформації, яка з часом змінюється як за структурою, так і за змістом. Він надає графічну нотацію, яка використовується для концептуального моделювання, аналогічну моделювання сутності-відносини, з розширеннями для роботи з часовими даними. Техніка моделювання включає чотири моделюють конструкції: якор, атрибут, зв'язок і вузол (the anchor, attribute, tie, knot), кожна з яких відображає різні аспекти області, що моделюється [9]<sup>1)</sup>.

На рис. 9 показаний типовий приклад анкерної моделі бази даних.

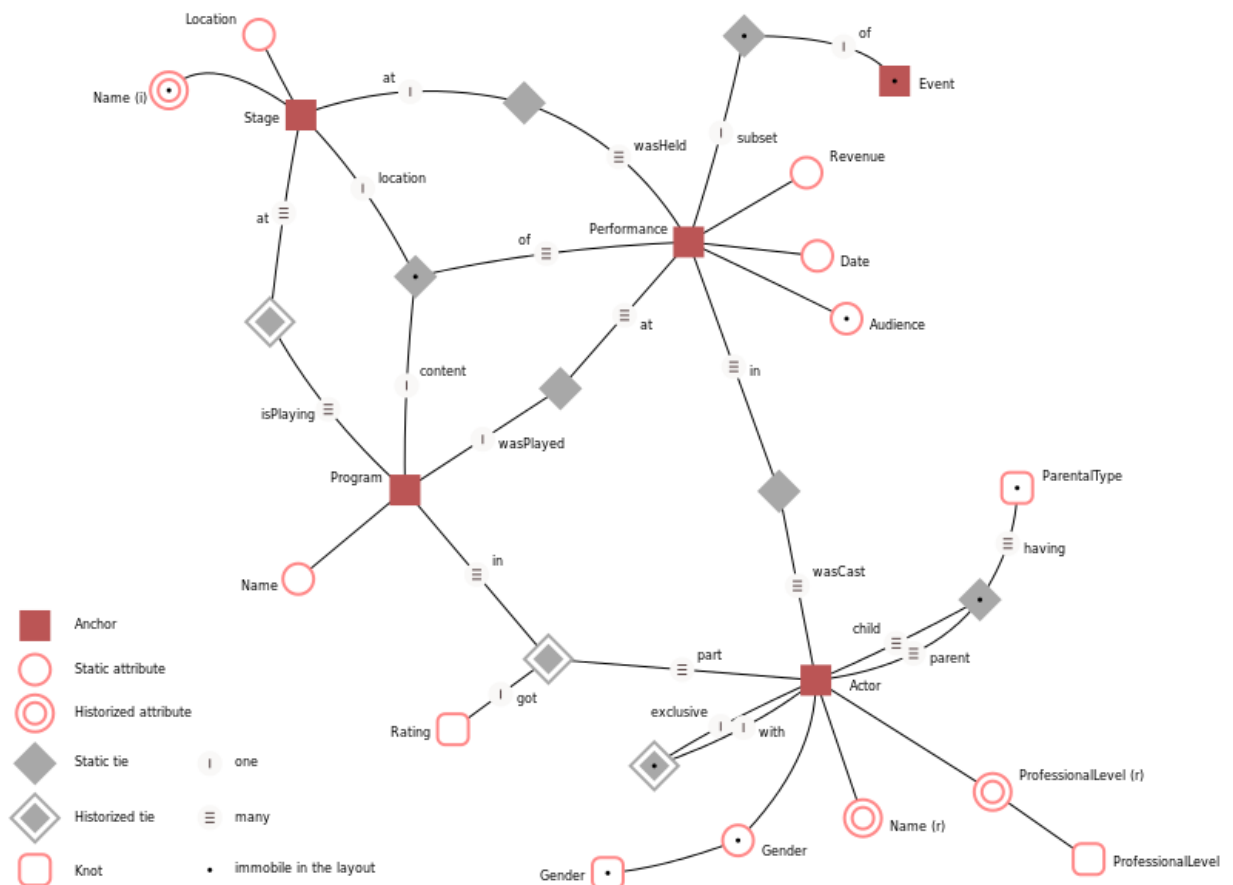


Рисунок 9 – Приклад анкерної моделі БД

<sup>1)</sup> [9] Anchor modeling. URL: [https://en.wikipedia.org/wiki/Anchor\\_modeling](https://en.wikipedia.org/wiki/Anchor_modeling)



Якоря використовуються для моделювання сутностей і подій, атрибути використовуються для моделювання властивостей якорів, зв'язки моделюють відносини між якорями, а вузли використовуються для моделювання загальних властивостей, таких як стани. Атрибути і зв'язки можуть бути витлумачені, коли зміни в інформації, яку вони моделюють, необхідно зберегти.

На малюнку 9 наведена орієнтовна модель, що показує різні графічні символи для всіх концепцій. Символи нагадують ті, що використовуються при моделюванні відносин сутностей, з декількома розширеннями. Подвійний контур по атрибуту або зв'язку вказує, що зберігається історія змін, і також доступний символ вузла (обведений квадрат із закругленими краями).

Отримані моделі можна перетворити в фізичні проекти баз даних за допомогою формалізованих правил. Коли таке перетворення буде виконане, таблиці в реляційній базі даних будуть в основному в шостий нормальній формі [10]<sup>1)</sup>.

Анкерне моделювання було створено для того, щоб скористатися перевагами високого ступеня нормалізації, уникаючи при цьому її недоліків. Отримано такі переваги, як можливість неруйнівного розвитку моделі, щоб уникнути нульових значень і збереження інформації без дублювання. Проблеми з продуктивністю через додаткових об'єднань в значній мірі можна уникнути завдяки функції сучасних механізмів баз даних, званої «виключення об'єднання» або «виняток таблиці». Щоб обробляти зміни в інформаційному змісті, моделювання прив'язки імітує аспекти тимчасової бази даних в результуючій схемою реляційної бази даних. орієнтована, документо-орієнтована.

Проблеми з продуктивністю через додаткові об'єднання в значній мірі можна уникнути завдяки функції сучасних механізмів баз даних, званої «виключення об'єднання» або «виключення таблиці». Щоб обробляти зміни в

---

<sup>1)</sup> [10] 6th TDWI European Conference / TDWI homepage Archived July 20, 2011, at the Wayback Machine

інформаційному змісті, моделювання прив'язки імітує аспекти часової бази даних в результуючій схемі реляційної бази даних.

Найбільш ранні установки з використанням якірного моделювання були зроблені в Швеції, а перша була побудована в 2004 році, коли з використанням цієї техніки було побудовано сховище даних для страхової компанії. У 2007 році цей метод використовувався в декількох сховищах даних і одній системі OLTP, і він був представлений на міжнародному рівні Ларсом Рённбеком на конференції 2007 Transforming Data with Intelligence] (TDWI) в Амстердамі. Це викликало досить інтересу до техніки, щоб дати більш формальний опис. З тих пір дослідження, що стосуються моделювання якоря, проводяться у співпраці між творцями Олле Регардтом і Ларсом Рённбеком і командою факультету комп'ютерних і системних наук Стокгольмського університету. Перша стаття, в якій формалізовано якірне моделювання, була представлена в 2009 році на 28-й Міжнародній конференції з концептуального моделювання і отримала нагороду за кращу роботу [11]<sup>1)</sup>.

Якірне моделювання обробляє два типи інформаційної еволюції: структурні зміни та зміни змісту. Зміни в структурі інформації представлені через розширення. Високий ступінь нормалізації дозволяє неруйнівним чином додавати необхідні концепції моделювання, які необхідні для фіксації змін, таким чином, що кожна попередня схема завжди залишається підмножиною поточної схеми. Оскільки існуюча схема не зачіпається, це дає перевагу можливості розвивати базу даних дуже ітеративно і не спричиняючи простою.

Зміни у змісті інформації здійснюються шляхом наслідування подібних особливостей часової бази даних у реляційній базі даних. При анкерному моделюванні частину інформації можна прив'язати до моментів часу або до інтервалів часу (як відкритих, так і закритих). Часові моменти, коли відбуваються події, моделюються за допомогою атрибутів, наприклад, дати народження людей або часу покупки. Інтервали часу, протягом яких значення є

---

<sup>1)</sup> [11] 28th International Conference on Conceptual Modeling – GRAMADO, BRAZIL.  
URL: <http://www.inf.ufrgs.br/er2009/>

дійсним, фіксуються через історизацію атрибутів та зв'язків, наприклад, зміни кольору волосся людини або проміжок часу, протягом якого людина перебуває у шлюбі. У реляційній базі даних це досягається додаванням до таблиці, що відповідає історизованому атрибуту або зв'язку, одного стовпця з достатньо детальним типом даних, щоб зафіксувати швидкість змін. Це додає невелику складність, оскільки потрібно вивчити більше одного рядка в таблиці, щоб знати, закритий інтервал чи ні.

При анкерному моделюванні існує співвідношення один до одного між символами, що використовуються в концептуальній моделі, і таблицями в реляційній базі даних. Кожен якір, атрибут, зв'язок і вузол мають відповідну таблицю в базі даних з однозначно визначеною структурою. Таким чином, концептуальна модель може бути перетворена в схему реляційних баз даних за допомогою простих автоматизованих правил, і навпаки. Це відрізняється від багатьох інших методів моделювання, в яких між концептуальним, логічним та фізичним рівнями існують складні, а іноді і суб'єктивні етапи перекладу.

Якірні таблиці містять один стовпець, в якому зберігаються ідентичності. Прийнято, що ідентичність є єдиною властивістю сутності, яка завжди присутня і незмінна. Оскільки ідентифікаційні дані рідко доступні з модельованого домену, вони натомість технічно генеруються, наприклад, із послідовності великих чисел [12]<sup>1)</sup>.

Прикладом якоря для особи племінників Дональда Дака є набір кортежів з одним полем:

```
{{#42}, {#43}, {#44}}
```

Вузли можна розглядати як поєднання якоря і одного атрибута. Таблиці вузлів містять два стовпці, один для ідентичності та другий для значення. Через спільне зберігання особистостей та цінностей вузли неможливо історизувати. Їх корисність полягає в можливості зменшити вимоги до сховища та

---

<sup>1)</sup> [12] [https://en.wikipedia.org/wiki/Anchor\\_modeling](https://en.wikipedia.org/wiki/Anchor_modeling)

покращити продуктивність, оскільки таблиці, що посилаються на вузли, можуть зберігати коротке значення, а не довгий рядок.

Прикладом вузла для гендерної ознаки є набір кортежів з двома полями:

```
{{#1, 'Чоловік'}, {#2, 'Жінка'}}
```

Статичні таблиці атрибутів містять два стовпці, один для ідентичності сутності, якій належить значення, а другий для фактичного значення властивості. Історизовані таблиці атрибутів мають додатковий стовпець для зберігання початкової точки інтервалу часу. У таблиці атрибутів, що зав'язується, стовпець значень – це ідентичність, яка посилається на таблицю вузлів.

Прикладом статичного атрибута для імен племінників є набір кортежів з двома полями:

```
{{#42, 'Huey'}, {#43, 'Dewey'}, {#44, 'Louie'}}
```

Прикладом зав'язаного статичного атрибута для їх статі є набір кортежів з двома полями:

```
{{#42, #1}, {#43, #1}, {#44, #1}}
```

Прикладом історичного атрибута (зміни) кольорів їхнього вбрання є набір кортежів з трьома полями:

```
{{#44, 'Orange', 1938-04-15}, {#44, 'Green', 1939-04-28}, {#44, 'Blue', 1940-12-13}}
```

Статичні таблиці зв'язків пов'язують два або більше якорів один до одного і містять два або більше стовпців для зберігання ідентифікаційних даних. Історизовані таблиці зв'язку мають додатковий стовпець для зберігання початкової точки інтервалу часу. Вузлові таблиці з зв'язками мають додатковий стовпець для кожного вузла, на який посилаються.

Прикладом статичного зв'язку для стосунків між братами чи сестрами є набір кортежів з двома полями:

```
{{#42, #43}, {#42, #44}, {#43, #42}, {#43, #44}, {#44, #42}, {#44, #43}}
```

Усі отримані таблиці будуть у шостій нормальній формі, за винятком зв'язків, у яких не всі стовпці є частиною первинного ключа.

## 6 ПРОЕКТУВАННЯ СХОВИЩА ДАНИХ КОМПАНІЇ

### 6.1 Доцільність використання темпоральної та анкерної моделей

Анкерне моделювання баз даних до 6NF та використання темпоральних баз даних має сенс у випадку існування вимірів з мінливими атрибутами, зміни яких у часі потрібно відстежувати. Якщо є мінливі атрибути, зміна значення яких не впливає на бізнес-процеси компанії, використання складних моделей невиправдане.

Наприклад, зміна атрибуту Married (Одружений) для виміру «Покупец» не призведе до зміни у агрегованих даних, які використовуються в сховищах даних. Аналізувати попит покупців за цією ознакою навряд чи має сенс.

Покупців можна групувати за віковими групами, рівнем освіти, соціальними групами, місцем проживання. Так, всі ці ознаки людини є змінними у часі, але немає сенсу зберігати історію зміни цих ознак – крім вікової групи, яка розраховується при зберіганні дати (або року) народження покупця.

Якщо потрібен аналіз попиту на різні групи товарів та обсягів купівлі товарів за ознаками цих мінливих атрибутів, створювати агрегати за цими ознаками необхідно одразу при отриманні необхідних деталізованих даних, і вважати значення мінливого атрибуту таблиці-виміру постійним протягом періоду, що визначає гранульованість даних, потрібних для аналізу.

Наприклад, потрібно провести аналіз зміни попиту та обсягів покупок для у різні місяці року для покупців з різним освітнім рівнем. У цьому випадку вважається, що протягом одного місяця всі покупці не змінюють свою групу: без середньої освіти; середня освіта; незакінчена вища освіта(коледж, молодший бакалавр); вища освіта (бакалавр, магістр); науковий ступінь.

Наприкінці місяця у цьому випадку бажано проводити соціологічне опитування покупців з проханням вказати значення мінливих атрибутів, за якими може проводитись аналіз.

Але найчастіше такий аналіз зовсім не буває мати сенсу. Вказані атрибути покупця є необов'язковими при заповненні картки реєстрації на веб-

сайті компанії для можливості здійснення покупок. Крім того ніхто не перевіряє істинність особистих даних, що вводить при реєстрації покупець. Більшість людей не любляють соціологічні опитування, тому не будуть приймати в них участь щомісяця. Таким чином, проводити аналіз продажів товарів за цими ознаками недоцільно.

## **6.2 Детальне проектування архітектури даних**

Детальне проектування включає в себе етапи логічного та фізичного проектування.

Цей етап – один з найважливіших для проекту створення сховища даних. Помилки і прорахунки, допущені на цьому етапі, буде дуже складно виправити на наступних етапах. Крім того, подібні прорахунки можуть призвести до перегляду всього проекту і, отже, до його фактичного провалу.

При розробці логічної моделі даних дуже важливим є взаємодія проєктувальників і бізнес-аналітиків предметної області. Не можна покладатися на думку тільки однієї з цих груп: проєктувальники зазвичай недостатньо обізнані про предметну область СД, а бізнес-аналітики предметної області можуть не надати повної інформації про функціональні залежності в даних. В результаті недостатня взаємодія цих груп розробників сховища даних призводить до проектування частково неправильної схеми даних

При виконанні цього етапу спочатку будується логічна модель даних, яка згодом перетвориться в фізичну модель.

На етапі логічного проектування виконується розробка логічних моделей даних для сховища даних та вітрин даних в робочому просторі бази даних. Потім виконується відображення логічних моделей даних джерел даних в логічні моделі сховища даних та вітрин даних.

На етапі фізичного проектування виконується логічні об'єкти перетворюються в фізичні структури сховища даних. Проводиться денормалізація логічної моделі. Проводиться оцінка необхідності додаткових індексів крім

тих, що автоматично створюються засобами СКБД при створенні таблиць БД. Розробляються необхідні індекси та обмеження.

На стадії логічного проектування розглядаються логічні взаємовідносини між об'єктами предметної області. Особливістю логічного проектування СД є повна орієнтація на потреби і завдання кінцевого користувача – бізнес-аналітиків і менеджерів компанії різних рівнів. Кінцеві користувачі будуть аналізувати дані і працювати з агрегованими даними, а не з даними конкретної бізнес-операції. Цей клас користувачів, як правило, зазвичай не знає точно, що їм все-таки потрібно від даних. Вони знають, відповіді на які питання потрібно отримати при експлуатації системи, а не те, які саме дані при цьому використовуються. І тут важливо оцінити рівень претензій до даних і окреслити коло завдань, які можуть виникнути.

В процесі логічного проектування виділяється набір об'єктів предметної області – типів сутностей – з їх атрибутами. Об'єкт являє собою фрагмент інформації про предметну область. У реляційних базах даних тип сутності відображається в відношення – плоску таблицю, а його атрибути відображаються в колонки наступній таблиці.

У логічному проектуванні СД для створення ER-діаграм використовується багатовимірне моделювання, яке, в основному, зводиться до ідентифікації інформації про об'єкт у вигляді фактів (таблиці фактів) та ідентифікації інформації, за допомогою якої ці факти можна переглянути (таблиці вимірювань). Результатом стадії логічного проектування є логічна схема СД.

### **6.3 Логічне проектування СД компанії з продажу через Інтернет**

Одним з ключових моментів побудови архітектури даних є ступінь деталізації інформації при перетворенні її в елементи даних. Процес такого перетворення називають структуризацією даних.

Рівень структуризації (деталізації або гранулювання) даних (Data granularity) є однією з найважливіших характеристик СД. Рівень структуризації даних – це ступінь деталізації даних, що зберігаються, оптимальна з то-

чки зору вирішення інформаційно-аналітичних завдань в рамках предметної області СД.

Підтримка високого рівня структуризації даних призводить до необхідності зберігати і супроводжувати великі обсяги даних, що може негативно позначатися на продуктивності системи зберігання даних, зокрема, на часі відповіді на запит. З іншого боку, низький рівень структуризації даних призводить до того, що система зберігання даних може відповідати на строго обмежене коло запитів. Тому одним із завдань проектувальника СД на рівні логічного моделювання є прийняття рішення про оптимальний рівень структуризації даних в СД.

Структуризація даних передбачає розбиття всього набору даних на певні класи з метою подальшої деталізації всередині виділеного класу. Для СД характерні три основні види даних (класів).

- Фактичні (або деталізовані) дані (Real-time data) дані представляють собою поточний стан кількісних і якісних показників діяльності організації. Джерелом таких даних є зазвичай OLTP-системи. Таким даними притаманний високий рівень структуризації. Для того щоб використовувати такі дані в СД, їх потрібно попередньо обробити за допомогою процедур очищення.
- Похідні (або агреговані) дані (Derived data) дані представляють собою дані, які отримані в результаті підсумовування, агрегації і усереднення фактичних даних.
- Консолідовані дані (Reconciled data) – це фактичні дані, які були очищені і являють собою інтегрований джерело даних для вирішення задач аналізу. Основна вимога до таких даних – їх узгодженість (consistency).

Фактичні дані будуть постувати з різними інтервалами під час прийому замовлення та купівлі товарів. Ці дані через OLTP-системи повинні потрапляти у БД компанії. У БД компанії фактичні дані є мінливими – клієнт може



відмовитись від замовлення, або змінити деякі параметри замовлення. У СД повинні потрапляти тільки остаточні, очищені фактичні дані.

У більшості комерційних СКБД є вбудовані засоби ETL (Extract-Transform-Load) – засоби завантаження даних у СД з попереднім їх очищенням та перетворенням у необхідні структури даних. Тому для побудови логічної моделі даних перш за все потрібно вирішити питання з гранулюванням даних.

Найбільш природним для компаній роздрібної торгівлі через мережу Інтернет є фіксація факту продажу товару покупцю на протязі одного дня. Будемо вважати такі дані детальними для сховища даних.

Агреговані дані повинні включати в себе групування даних за декількома критеріями:

- угруповання тільки по часу: сумарні витрати кожного клієнта протягом заданого часового інтервалу: місяця, кварталу, року;
- угруповання за часом і додатковим фактором: місце розташування покупця; сезон року; вікова група покупця; група товарів; тощо.

Найчастіше для логічного моделювання предметної області використовують модель «сутність – зв'язок». Застосування методу створює модель предметної області, яка не залежатиме від фізичної реалізації. Метод застосовується як при моделюванні предметних областей OLTP-систем, так і при моделюванні предметних областей систем бізнес-аналітики (BI-system). Знання цього методу допомагає проектувальнику СД швидко встановити логічні зв'язки між моделями БД OLTP-систем масштабу організації та моделями СД BI-систем.

Результатом моделювання методом "сутність – зв'язок", або ER-моделювання, є ER-модель. ER-модель представляється за допомогою ER-діаграм, які є графічної нотації для абстрагування даних у вигляді сутностей, взаємозв'язків і атрибутів.

Таким чином, семантика предметної області представляється в ER-моделі в термінах суб'єктивних засобів опису – сутностей, атрибутів, ідентифікаторів сутностей, супертіпа, підтипів і т.д.

### **6.3.1 Сутності предметної області, що моделюється**

Сутність предметної області є результатом абстрагування реального об'єкта шляхом виділення і фіксації набору його властивостей. Сутність є абстракцією реального об'єкта, тому що серед всієї сукупності характеристик реального об'єкта для опису сутності вибираються тільки ті властивості об'єкту, які є значущими в контексті даної предметної області. Таким чином, сутність представляє клас об'єктів, який є результатом абстрагування реального об'єкта. Зазвичай сутності позначаються іменником природної мови.

Сутність описується за допомогою даних, іменованих властивостями або атрибутами (attributes) сутності. Сутності вступають в зв'язки один з одним через свої атрибути. Кожна група атрибутів, що описують один реальний прояв сутності, являє собою екземпляр сутності (instance). Іншими словами, екземпляр сутності – це реалізації сутності, що відрізняються один від одного і допускають однозначну ідентифікацію. Іменування суті в однині полегшує в подальшому читання моделі. Фактично, ім'я сутності дається по імені її примірника.

Одним з основних комп'ютерних способів розпізнавання сутностей в ІС є присвоєння сутностям ідентифікаторів (Entity identifier). Часто ідентифікатор сутності називають ключем. Завдання вибору ідентифікатора суті є семантично суб'єктивної завданням. Оскільки сутність визначається набором своїх атрибутів, для кожної сутності доцільно виділити таку підмножину атрибутів, яке однозначно ідентифікує дану сутність.

Деякі суті мають природні ідентифікатори. Наприклад, природним ідентифікатором сучасної людини є його ідентифікаційний номер. Ідентифікатори суті можуть бути складовими – що складаються з декількох атрибутів, і атомарними – що складаються з одного атрибута сутності.

Унікальний ідентифікатор сутності – це атрибут сутності, що дозволяє відрізнити одну сутність від іншої. Якщо сутність має кілька унікальних ідентифікаторів, так званих можливих ключів, то проектувальник повинен вибрати первинний ключ сутності.

Розрізняють однозначні і багатозначні атрибути. Однозначними є атрибути, які в межах конкретного екземпляра сутності мають тільки одне значення. В іншому випадку вони вважаються багатозначними.

Для предметної області «Продаж товарів через Інтернет» можна одразу виділити дві сутності: «Товар» і «Покупець». Для предметної області «Роздрібна торгівля через магазин» виділяють також сутність «Продавець». Але у продажу товару через мережу Інтернет є свої особливості.

У цьому випадку продаж товару знеособлена. Клієнт сам вибирає товар, система автоматично генерує бланк замовлення і виставляє рахунок покупцеві. Покупець може відразу оплатити замовлення банківською картою або вибрати оплату готівкою при отриманні замовлення. Продавець відсутній в ланцюжку «вибір товару – оформлення замовлення – оплата покупки – отримання покупки». У компаніях з продажу товарів через Інтернет присутні співробітники, які можуть допомогти клієнтам Інтернет-магазину у пошуку та виборі товару. Але ці працівники не є продавцями, вони є менеджерами з продажу.

Але в предметній області компанії з продажу товарів через Інтернет повинна бути ще одна сутність – «Постачальник». Ця сутність може бути відсутня, якщо компанія сама виробляє товари, які продає. Вірніше, ця сутність буде присутня для такого виду компаній в частковій предметній області «Виробництво». Щоб виробити власний товар потрібно закупити матеріали та комплектуючі у інших виробників.

Сутності «Покупець», «Товар», «Постачальник» для багатовимірної моделі є таблицями вимірів. Таблицею фактів у логічній моделі буде асоціативна сутність «Продаж».

Компанії з продаж товарів через Інтернет дуже часто використовують акції для підвищення попиту покупців. Аналітикам компанії потрібно аналізувати, як впливає акція на попит покупців, та як це відображається на прибутках компанії. Тому наступною сутністю в логічній моделі СД повинна бути «Акція».

### **6.3.2 Атрибути сутностей предметної області, що моделюється**

У сутності «Товар» обов'язковим атрибутом буде «Назва». Також для полегшення ідентифікації товару покупцем потрібен атрибут «Опис товару». Для полегшення класифікації товарів потрібні атрибути «Фірма» (фірма-виробник), «Категорія».

Колір товару можна включити як необов'язковий атрибут, але це призведе до виникнення значень null для товарів, колір яких не вказують. Зазвичай моделі одного товару з різним кольором мають різну назву, наприклад, «Product XXX-Red», «Product XXX-Black», тому додавати окремий атрибут кольору у сутність «Товар» не потрібно.

Назва товару буде унікальним ключем сутності «Товар».

У сутності «Покупець» обов'язковим атрибутом буде «Ім'я». Більшість компаній має на увазі, що цей атрибут зберігає ім'я та прізвище покупця. Ім'я людини не є унікальним ідентифікатором, навіть поєднання імені і прізвища не є унікальним. У більшості Інтернет-магазинів для ідентифікації покупців використовується номер мобільного зв'язку. Тому другим обов'язковим атрибутом сутності «Покупець» буде номер мобільного телефону.

Компанії з продаж через Інтернет воліють знати електронні адреси своїх покупців, хоча і не оголошують цю характеристику обов'язковою для реєстрації. Таким чином, третім атрибутом буде електронна адреса.

Щоб доставити товар покупцеві, потрібно знати його місцезнаходження, отже наступний атрибут сутності «Покупець» – місто проживання. Зазвичай повна адреса не є обов'язковим атрибутом, тому що більшість товарів

поставляються у поштові відділення, або у пункти отримання товарів, а повідомлення про доставку приходять на телефон як SMS.

Але атрибут «Адреса» може бути необов'язковим для можливості доставки товару кур'єром. Необов'язковими атрибутами також будуть стать покупця та дата його народження.

У сутності «Постачальник» обов'язковим атрибутом буде «Назва». Також обов'язковими атрибутами сутності «Постачальник» будуть «Телефон» та «Електронна адреса». Всі ці атрибути будуть унікальними ключами.

Також атрибутом сутності «Постачальник» повинна бути фізична адреса та банківський рахунок, на який поступають виплати за доставку товарів.

У асоціативної сутності «Продаж» повинні бути атрибутами ключові атрибути сутностей «Товар» та «Покупець». При купівлі товару покупець не зв'язаний безпосередньо з постачальником цього товару, тому у сутності «Постачальник» буде зв'язок тільки з сутністю «Товар».

Таблиця «Товар» є таблицю вимірів. Таким чином, у схемі є ієрархія вимірів, і у моделі використовується не схема «зірка», а схема «сніжинка».

Атрибутами-фактами в таблиці «Продаж» будуть кількість товару та ціна одиниці товару.

Ще одною таблицею-виміром буде таблиця «Час». У цієї таблиці повинен бути атрибут-ідентифікатор та атрибути, що є частками визначення дати: день місяця, номер місяця, номер кварталу, номер року. Номер кварталу не є обов'язковим атрибутом, тому що він отримується з номеру місяця. Можна зберігати лише дату продажу, але наявність серед атрибутів таблиці «Час» вказаних часток дати полегшує зв'язок між агрегованими даними різного ступеня агрегації за часом.

Атрибут-ідентифікатор таблиці «Час» також є серед зовнішніх ключів таблиці фактів «Продаж». Унікальним ключем цієї таблиці є поєднання атрибутів «День» – «Місяць» – «Рік».

У сутності «Акція» складніше виявити значущі атрибути. Унікальним атрибутом буде назва акції. Але деякі акції проводяться періодично, напри-

клад, «Чорна п'ятниця». Тому або назва акції значення атрибуту таблиці повинна бути повністю унікальною, наприклад «Чорна п'ятниця 27-11-2020», або унікальний ключ повинен бути складовим. Додатковими атрибутами акції будуть дата початку та дата закінчення акції.

Деякі акції можуть відноситись лише для заданих груп товарів заданих фірм-виробників. Тому серед атрибутів таблиці «Акція» повинні бути «Група товарів» і «Фірма».

### **6.3.3 Зв'язки між сутностями схеми**

Більшість зв'язків між таблицями вже виявлені вище. Але не зрозумілими залишились взаємовідносини між таблицею «Акція» та іншими таблицями.

Таблицю існування (Existence Table), зазвичай використовують, коли для таблиць вимірювання типу часу може існувати перекривання періодів – наприклад, в маркетинговій акції з просування товару на ринок, пов'язаної з гнучкою сіткою цін на товари. При цьому не всі товари братимуть участь в даній маркетингової акції.

Тому таблиця існування «Товари акції» буде пов'язана з таблицями «Товар» і «Продаж» і серед її атрибутів будуть дати початку і кінця акції.

Вимір «Товар» дозволяє аналізувати типові схеми закупівель товарів і відповідати на питання, які товари, як правило, купуються одночасно покупцями.

Вимір «Покупець» дозволяє аналізувати покупки з урахуванням їх частоти, географічного положення та кількості.

Таблиці вимірювань є своєрідним путівником при вибірці рядків з таблиці фактів.

Таблиці фактів мають велику кількість рядків. Таблиці вимірів, як правило, мають набагато менше число рядків. Одним з головних переваг використання такої схеми є те, що продуктивність операції СКБД з'єднання таблиці фактів і таких таблиць з'єднань буде близька до оптимальної.

На рис. 10 представлена логічна модель СД.

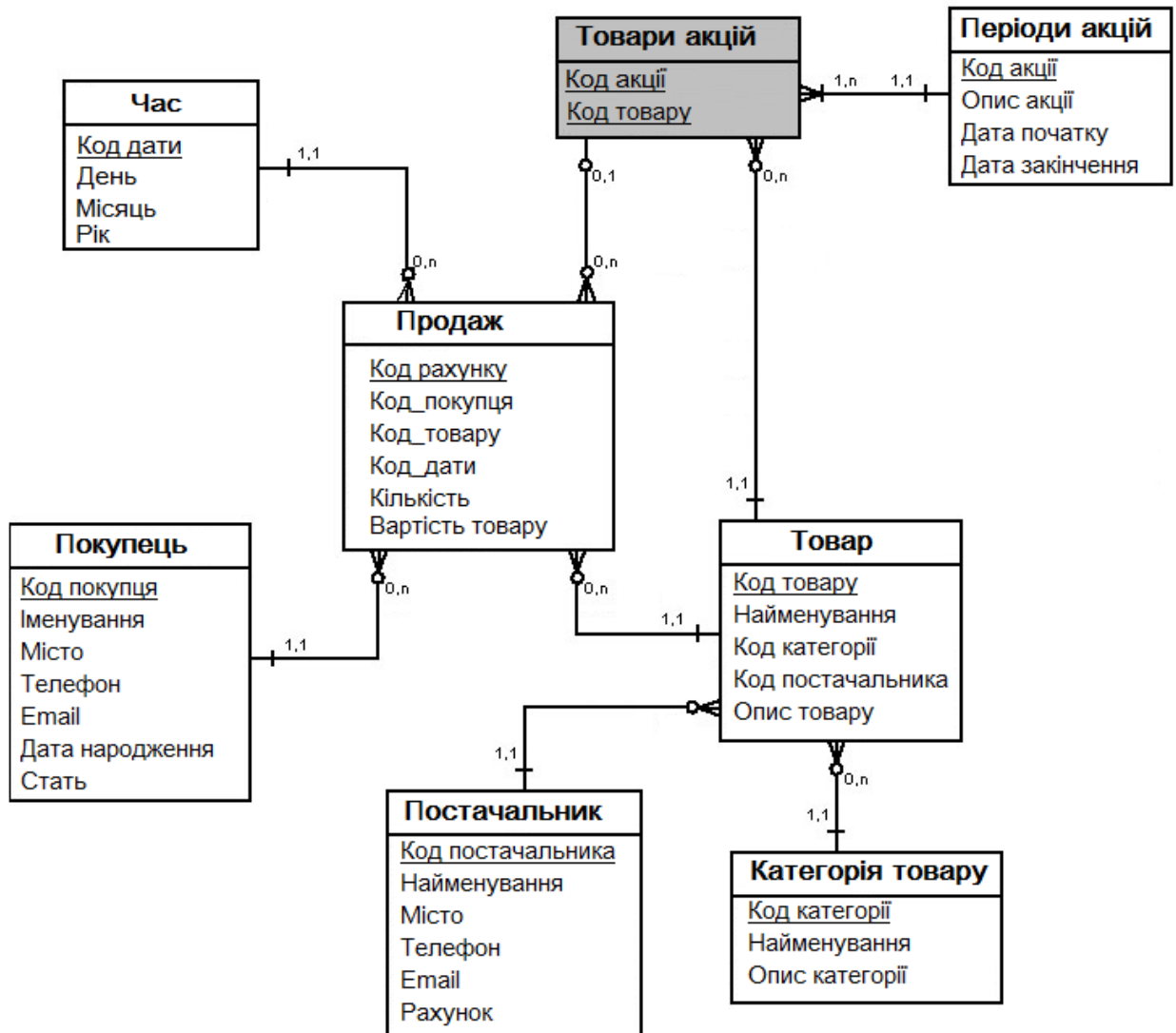


Рисунок 10 – Схема СД компанії з Інтернет-продажів

## 7 ВИБІР СКБД ДЛЯ РОЗРОБКИ СХОВИЩА ДАНИХ

### 7.1 Реляційні бази даних

Однією з головних переваг реляційних систем керування базами даних (РСКБД) над системами, заснованими на інших моделях даних є мова структурованих запитів SQL (Structured Query Language). Мову SQL називають мовою реляційних баз даних.

Завдяки цій мові можна виконувати вибірки з баз даних як завгодно складної структури. Мова SQL дозволяє знаходити узагальнені (агреговані) дані, які використовуються, наприклад, в корпоративних сховищах даних.

Переваги SQL:

1. Наявність міжнародних стандартів.
2. Незалежність від конкретної СКБД. Незважаючи на наявність діалектів і відмінностей в синтаксисі, в більшості своїй тексти SQL-запитів можуть бути досить легко перенесені з однієї СКБД в іншу.
3. Підтримка архітектури клієнт-сервер.
4. Декларативність. За допомогою SQL програміст описує тільки те, які дані потрібно витягнути або модифікувати. Яким чином це зробити, вирішує СКБД безпосередньо при обробці SQL-запиту.
5. Поширеність.
6. Швидке навчання.

Основні конструкції мови SQL затверджені міжнародними стандартами і однакові в усіх СКБД незалежно від компанії-розробника, так само, як і в відкритих СКБД. До стандартів SQL відносяться і конструкції мови, які дозволяють створювати агрегати – групувати дані за деякими критеріями. Цей факт важливий для розробки сховищ даних, оскільки при експлуатації СД основні групи користувачів – керівники різних рівнів компанії та аналітики – використовують у своїх запитах саме агреговані дані.



Комерційні СКБД мають більше можливостей для розробки сховищ даних та застосувань для бізнес-аналітики. Використання вільних СКБД призведе до необхідності розробляти власні засоби для інтеграції даних та застосування для бізнес-аналітики, або купувати відповідні програмні продукти на ринку програмного забезпечення. Але ці продукти коштують досить дорого.

У останні роки компанія Microsoft випускає безкоштовні редакції своїх комерційних продуктів, наприклад, СКБД SQL Server та середовище швидкої розробки проектів і додатків Visual Studio. Звичайно, безкоштовна редакція має дещо обмежений функціонал у порівнянні з комерційними редакціями, а також призначена для використання у навчальних закладах та для малого бізнесу. Але якщо компанія входить у ці обмеження і її задовольняє наявний функціонал програмних систем, то вибір цих продуктів для створення СД є цілком виправданим.

### **7.1.1 Діалект мови SQL для СКБД SQL Server**

Незважаючи на наявність стандартів мови SQL, практично в кожній СКБД застосовується свій діалект мови. Одним зі діалектів SQL є мова T-SQL.

Мова T-SQL – це власний діалект мови структурованих запитів, який застосовується в СКБД SQL Server. Мова T-SQL призначена виключно для роботи з СКБД SQL Server, але основна частина операторів, що застосовується у ній, має більш широке використання.

Мова T-SQL сумісна зі стандартом ANSI SQL-92 на початковому рівні. Це означає, що в певному обсязі T-SQL сумісна з дуже широким відкритим стандартом. Слід враховувати, що в кожній реляційної СКБД застосовуються різні розширення і способи підвищення продуктивності, що доповнюють зазначений стандарт ANSI і виходять за його рамки.

Основна відмінність між SQL та T-SQL є те, що SQL – це мова, яка використовується для зберігання і управління даними в СКБД, а T-SQL є роз-

ширеною версією SQL і призначена для виконання операцій на сервері MS SQL Server.

Додатки можуть зв'язуватися з SQL Server, виконуючи оператори T-SQL. Розробник може писати запити для виконання операцій над таблицями, об'єднання таблиць і додавання обмежень. Він також може виконувати транзакції, писати збережені процедури, подання, індекси і багато іншого. Існують різні числові, рядкові функції, функції дати. Крім того, існують функції агрегування для виконання операції з набором значень.

При підготовці випуску СКБД SQL Server 2005 мова T-SQL була в значній мірі доопрацьована, і в неї були додані багато нових програмних конструкцій. Крім усього іншого, вона була перетворена в мову, сумісну із загальним середовищем виконання (Common Language Runtime CLR) операційної системи Windows; таким чином, починаючи з цього випуску T-SQL стала однією з мов .NET.

В усіх версіях СКБД SQL Server, починаючи з версії SQL Server 2005, передбачена можливість використовувати для доступу до бази даних будь-яку мову .NET, але в кінцевому підсумку в основі звернення безпосередньо до самих даних завжди лежить якийсь код SQL, тому мова T-SQL залишається базовою мовою при виконанні будь-яких дій в СКБД SQL Server. З кожним випуском SQL Server багато змінюється, але найбільш фундаментальні оператори доступу до даних є практично такими ж, як і раніше.

T-SQL – це процедурне розширення мови SQL компаній Microsoft. SQL була розширена такими додатковими можливостями як:

- керуючі оператори;
- локальні і глобальні змінні;
- різні додаткові функції для обробки рядків, дат, математики тощо;
- підтримка аутентифікації Microsoft Windows.

Мова T-SQL є ключем до використання SQL Server. Всі додатки, які взаємодіють з екземпляром SQL Server, незалежно від їх реалізації і призна-

ченого для користувача інтерфейсу, відправляють серверу інструкції T-SQL.

### 7.1.2 Збережені процедури у мові T-SQL

Збережені процедури і функції представляють собою набір SQL-операторів, які можна зберігати на сервері. Якщо сценарій збережений на сервері, то клієнтам не доведеться повторно ставити одні й ті ж окремі оператори, замість цього вони зможуть звертатися до збереженій процедурі [13] – [14]<sup>1)</sup>.

Ситуації, коли збережені процедури особливо корисні:

- Численні клієнтські програми написані на різних мовах або працюють на різних платформах, але повинні виконувати однакові операції з базами даних.
- Безпека грає першорядну роль. Збережені процедури використовуються для всіх стандартних операцій, що забезпечує сумісність і безпеку середовища, а процедури гарантують належну реєстрацію кожної операції. При такому типі установки додатка, і користувачі не отримують безпосередній доступ до таблиць бази даних і можуть виконувати тільки конкретні процедури, що зберігаються.
- Необхідно знизити мережевий трафік між клієнтом і сервером. Обсяг інформації, що пересилається між сервером і клієнтом істотно знижується, але збільшується навантаження на систему сервера баз даних, так як в цьому випадку на стороні сервера виконується велика частина роботи по обробці даних.

---

<sup>1)</sup> [13] Хранимые процедуры. URL: <https://metanit.com/sql/sqlserver/11.1.php>

[14] CREATE PROCEDURE (Transact-SQL). Microsoft. Документация по SQL. URL: <https://docs.microsoft.com/ru-ru/sql/t-sql/statements/create-proceduretransact-sql?view=sql-server-ver15>

## 7.2 Вибір СКБД для побудови сховища даних

При описі архітектури сховищ даних відмічалось, що реляційна і багатовимірна архітектура сховища даних мають свої недоліки. Головним недоліком для реляційних сховищ даних (РСД) є порівняно великий час виконання аналітичних запитів. Але у сховищ даних цієї архітектури є і вагомі переваги. Для сховища даних, яке тільки розробляється, головною перевагою буде досить проста процедура реструктуризації СД – зміна її структури: кількості вимірів та їх атрибутів, можливість зміни зв'язків між таблицями.

Тому для сховища даних, що розробляється, вибирається у якості архітектури РСД.

Для фізичної реалізації БД була вибрана реляційна СКБД Microsoft SQL Server 2017.

## 8 Фізичне проектування сховища даних

На стадії фізичного проектування розглядаються завдання розміщення даних в БД для ефективної їх вибірки. Для реляційних сховищ даних на цій стадії пишуться SQL-оператори. Дані, зібрані на стадії логічного проектування, перетворюються в опис фізичної БД, включаючи таблиці, унікальні ідентифікатори об'єктів – в первинні ключі, обмеження за значеннями даних, взаємини між об'єктами – в зовнішні ключі, індекси.

Призначення табличного простору полягає в відособленні таблиць від їх індексів, маленьких таблиць від великих таблиць, тобто є механізмом для вирішення завдання оптимального розміщення даних (деякі СКБД вирішують цю задачу самі, без втручання користувача).

Секціонування великих таблиць необхідно для збільшення продуктивності обробки запитів, тобто є одним з механізмів вирішення задачі оптимізації вибірки, так само, як і створення індексів.

Обмеження в СД відрізняються від обмежень в OLTP-системах. У системах зберігання даних цілісність і перевірка даних забезпечується на стадії завантаження даних. Тому роль обмежень в СД не настільки вже й велика. Типовим обмеженням в СД є обмеження NOT NULL.

### 8.1 Визначення доменів та типів даних таблиць

При створенні логічної моделі сховища даних у 6-му розділі були виявлені атрибути таблиць вимірів та таблиці фактів. Серед цих атрибутів більшість атрибутів є рядками символів різної довжини, тобто належать до домену символічних рядків.

Атрибути, які є ідентифікаторами таблиць:

- Код\_товару;
- Код\_покупця;
- Код\_постачальника;
- Код категорії;

- Код\_акції;
- Код\_дати;
- Код\_рахунку.

Всі ці атрибути належать до домену цілих чисел. Але не оптимальним буде вибирати для всіх перелічених атрибутів один тип даних.

В SQL Server тип `smallint` займає 2 байти і описує цілі числа в діапазоні від -32768 до 32767. Цього діапазону цілком достатньо для кодування екземплярів таких сутностей, як «Час» (вистачить на 89 років), «Категорія товару», «Періоди акцій» (акції починаються не щодня, навіть якщо паралельно можуть проходити декілька акцій).

Але вже для сутності «Постачальник» діапазону значень типу `smallint` може виявитись замало. Ще в більшій мірі це стосується сутностей «Товар», «Покупець», а тим більше – «Продаж».

Для кодування екземплярів цих сутностей потрібен тип довжиною у 4 байти – тип `int`.

Атрибут «Вартість товару», належить до домену грошових коштів. В SQL Server значення цього домену описуються типом `decimal`.

В таблиці вимірів «Час» атрибути, що описують складові календарної дати описуються за типом `smallint`.

Серед не ключових атрибутів інших таблиць присутній лише один атрибут, який належить до домену цілих чисел – атрибут «Кількість» таблиці фактів «Продаж». Цей атрибут описаний у контексті кількості екземплярів одного товару, що закупив один покупець за один раз. Тому для цього атрибуту буде достатньо типу `smallint`.

Ще три атрибути належать до домену календарних дат:

- Дата\_народження;
- Дата\_початку;
- Дата\_закінчення.

Ці атрибути можуть бути описані типом `date` або `datetime`.

Обмеження є важливим інструментом проектувальника, за допомогою якого він підтримує цілісність БД. Їх можна використовувати для того, щоб бути впевненим в тому, що колонка первинного ключа таблиці є унікальною і завжди містить значення. Обмеження застосовуються також для підтримки посилальної цілісності. Останнє означає, що значення в колонці зовнішнього ключа повинні існувати як деяке значення в колонці первинного ключа іншої таблиці.

Обмеження представляють собою спосіб застосування бізнес-правил предметної області на рівні БД і гарантують сумісність даних, що вводяться з тими, які вже знаходяться в таблицях. У реляційній БД під обмеженням розуміється правило (умова), якому повинен задовольняти певний елемент в БД.

Наприклад, умови, яким повинні додатково задовольняти значення колонки таблиці в рамках певного для неї типу даних (тобто тип даних плюс правило), повністю втілюють концепцію домену в фізичній моделі даних БД.

Для виключення можливості додавання декілька разів у таблицю одного і того ж запису потрібно додати для кожної таблиці обмеження UNIQUE, яке визначає атрибути таблиці, в яких значення не повинні повторюватись.

За звичай таким унікальним ключем є атрибут «Назва» або «Найменування». Але якщо в таблиці немає жодного атрибуту, значення якого не повинні повторюватись, потрібно вибрати складовий унікальний ключ з двох атрибутів.

Кількість атрибутів унікального ключа може бути більше 2.

Серед інших обмежень цілісності слід відмітити, що дуже не бажане знаходження у таблицях сховища даних порожніх значень, які позначаються як null. Тому для всіх атрибутів слід використовувати обмеження NOT NULL.

## 8.2 Створення таблиць бази даних

При вирішенні задачі завдання об'єктів БД для СД проєктувальник має на вході фізичну модель даних, таблиць вимірів і фактів, а на виході повинен створити набір команд CREATE TABLE, які будуть використовуватися для створення таблиць і інших об'єктів БД.

Створення таблиць вимірів.

Для створення таблиці вимірювання «Покупець» (Customer) команда CREATE TABLE має вид:

```
create table Customer (
  Cust_ID int identity,
  FName    varchar(20)    not null,
  LName    varchar(20)    not null,
  EMail    varchar(40) not null unique,
  Phone    varchar(13)   not null unique,
  Cust_City    varchar(20) not null,
  Cust_birtdate date null,
  constraint PK_CUSTOMER primary key (Cust_ID)
)
GO
```

Ніяких додаткових обмежень цілісності не потрібно.

Для створення таблиці вимірювання «Категорія» (Category) команда CREATE TABLE має вид:

```
create table Category(
  Cat_ID smallint identity,
  Name_cat varchar(50) not null UNIQUE NONCLUSTERED,
  Text_cat varchar(300) not null,
  constraint PK_CATEGORY primary key (Cat_ID)
)
GO
```



Ніяких додаткових обмежень цілісності не потрібно.

Для створення таблиці вимірювання «Постачальник» (Provider) команда CREATE TABLE має вид:

```
create table Provider (
  Prov_ID smallint identity
  Prov_FName      varchar(20)      not null,
  Prov_LName      varchar(20)      not null,
  City            varchar(20)      not null,
  Prov_Address    varchar(40)      null,
  EMail          varchar(40) not null unique,
  Phone          varchar(13) not null unique,
  Company        varchar(40)      not null,
  constraint PK_PROVIDER primary key (Prov_ID)
)
GO
```

Ніяких додаткових обмежень цілісності не потрібно.

Для створення таблиці вимірювання «Товар» (Product) команда CREATE TABLE має вид:

```
create table Product (
  Prod_ID int identity,
  Name_prod varchar(100) not null UNIQUE NONCLUSTERED,
  Cat_ID smallint REFERENCES Category(Cat_ID),
  Prov_ID smallint REFERENCES Provider(Prov_ID),
  Text_prod varchar(300) not null,
  Unit_Price numeric(8,2) not null,
  constraint PK_PRODUCT primary key (Prod_ID)
)
GO
```

Ніяких додаткових обмежень цілісності не потрібно.

Для створення таблиці вимірювання «Час» (Time) команда CREATE TABLE має вид:

```
CREATE TABLE Time (
    Time_ID    smallint identity,
    TDay       tinyint    not null,
    TMonth     tinyint    not null,
    TYear      smallint    not null,
    TQuarter   tinyint    not null,
    constraint PK_TIME primary key (Time_ID)
)
GO
```

Ніяких додаткових обмежень цілісності не потрібно.

Створення таблиці фактів.

Команда CREATE TABLE для створення таблиці фактів «Продаж» (Sale) на діалекті SQL сімейства СУБД MS SQL Server має вигляд:

```
CREATE TABLE Sale (
    Sale_ID    int identity,
    Time_ID    smallint REFERENCES Time(Time_ID),
    Cust_ID    int REFERENCES Customer(Cust_ID),,
    Prod_ID    int REFERENCES Product(Prod_ID),
    Amount     numeric(9,2)    not null,
    Quantity   integer        not null,
    constraint PK_SALE primary key (Sale_ID)
)
GO
```

## 9 СЕРВЕРНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Збережені процедури дозволяють проводити різноманітну обробку даних. В коді збережених процедур можуть використовуватись всі різновиди мови T-SQL:

- мова визначення даних DDL (Data Definition Language);
- мова маніпулювання даними DML (Data Manipulation Language);
- мова керування даними DCL (Data Control Language);
- мова запитів DQL (Data Query Language).

Мова запитів DQL є декларативною, а не процедурною. У запиті до БД вказується, що вибирається (які дані) і звідки (з яких таблиць та представлень), але не визначається, як саме виконується ця вибірка.

У збережених процедурах СКБД поряд з декларативними запитами мови DQL використовуються процедурні конструкції мови T-SQL: цикли, умовний оператор, оператор створення віртуальних таблиць – курсорів, та інші.

Курсор (CURSOR) створюється з використанням оператора SELECT, який формує структуру цієї віртуальної таблиці та заповнює її вмістом. Курсори можуть існувати тільки на час роботи процедури, потім вони повинні знищуватись, а пам'ять, що виділяється для них, повинна звільнюватись. Якщо потрібно, щоб результати обробки даних залишались після закінчення роботи збереженої процедури, потрібно створити у базі даних робочу таблицю необхідної структури, яка буде заповнюватись даними під час роботи збереженої процедури.

### 9.2 Збережена процедура DoAgregats

Збережена процедура DoAgregats заповнює таблиці фактів різного ступеня агрегації даними, отриманими з функцій БД, що виконують запити з групування по різним вимірам. У якості параметрів у функції передаються номер року, для якого розраховуються агреговані дані; для меншого ступеня агрегації параметром функцій є також номер місяця.

Процедура DoAgregats заповнює таблиці агрегованих фактів SaleGrCustjmerMonth, SaleGrProdMonth, SaleGrCustomerYear, SaleGrProdYear, SaleGrCategoryMonth, SaleGrCategoryYear.

Ці таблиці фактів використовуються для отримання бізнес-звітів.

## ВИСНОВКИ

Продаж товарів через мережу Інтернет в останні роки стає все більш популярним. Для показу товарів можливим покупцям на сайті компанії мають базу даних цих товарів.

Але для оцінки ефективності бізнесу та перспектив розвитку компанії потрібно мати сховище даних (СД), в якому містяться як поточні, так і історичні дані, міститься вся інформація, яка дозволяє оцінювати ефективність бізнесу та перспектив його розвитку: база даних покупців, постачальників товарів, продажів товарів, маркетингових акцій, тощо.

Це сховище даних повинне містити агреговані дані по продажу товарів, які використовуються аналітиками компанії для аналізу попиту на товари та впливу на попит та продаж товарів різноманітних факторів: сезонів року, місце проживання покупця, віку покупця, проведення маркетингових компаній, тощо.

За звітами аналітиків керівництво нижньої та середньої ланки розробляє поточну політику компанію, направлену на підвищення прибутку та розвиток компанії. Керівництво верхньої ланки розробляє стратегію розвитку компанії.

Метою даної комплексної кваліфікаційної роботи було моделювання, проектування та розробка сховища даних для компанії з Інтернет-продаж.

Завдання даної частини комплексної роботи було моделювання та проектування сховища даних.

При виконанні кваліфікаційної роботи були отримані наступні результати:

- було проведено дослідження сучасного стану у технології моделювання та проектування сховищ даних;
- були проаналізовані можливі моделі сховищ даних;

- були проаналізовані переваги та недоліки багатовимірних та реляційних сховищ даних;
- була досліджена та проаналізована анкерна модель сховища даних, яка розвивається в останні роки;
- анкерна модель сховища даних була визнана недоцільною через відсутність мінливих атрибутів у таблицях-вимірах сховища даних;
- для втілення сховища даних була обрана архітектура реляційного сховища даних;
- були розроблені логічна та фізична моделі сховища даних;
- для фізичної реалізації сховища даних була вибрана СКБД Microsoft SQL Server 2017 (безкоштовна редакція).

Задачі, поставлені в технічному завданню на виконання кваліфікаційної роботи повністю виконані.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Эрик Спирли. Корпоративные хранилища данных. Планирование, разработка и реализация. Т.1. М.: Вильямс, 2001. – 400 стр.
2. Корпоративные хранилища данных. Интеграция систем. Проектная документация. URL: [https://www.prj-exp.ru/dwh/what\\_is\\_dwh.php](https://www.prj-exp.ru/dwh/what_is_dwh.php) (Дата звернення 25.11.2020)
3. Архипенко С., Голубев Д., Хранилища данных. От концепции до внедрения. М: Диалог-Мифи, 2002. – 528с.
4. Куликов Г. Г., Никулина Н. О., Речкалов А. В. Управление проектами на основе системного моделирования: Учебное пособие. Уфа: УГАТУ, 2009. – 171 с.
5. Корпоративная модель данных. Лекции НОУ ИНТУИТ URL: <https://intuit.ru/studies/courses/599/455/lecture/10176> (Дата звернення 14.11.2020)
6. Шаламова Н.Г., Ивашечкова М.С. Определение стратегии развития организации с использованием подходов и методов многомерного моделирования. Вестник Университета №11, 2012. С.158 – 164.
7. Метод многомерного моделирования. Лекции НОУ ИНТУИТ URL: <https://intuit.ru/studies/courses/599/455/lecture/10167> (Дата звернення 14.11.2020)
8. Темпоральные данные и базы данных. URL: <https://intuit.ru/studies/courses/599/455/lecture/10165?page=1> (Дата звернення 08.12.2020)
9. Anchor modeling. URL: [https://en.wikipedia.org/wiki/Anchor\\_modeling](https://en.wikipedia.org/wiki/Anchor_modeling) (Дата звернення 08.12.2020)
10. 6th TDWI European Conference / TDWI homepage Archived July 20, 2011, at the Wayback Machine
11. 28th International Conference on Conceptual Modeling – GRAMADO, BRAZIL. URL: <http://www.inf.ufrgs.br/er2009/> (Дата звернення 03.12.2020)
12. [https://en.wikipedia.org/wiki/Anchor\\_modeling](https://en.wikipedia.org/wiki/Anchor_modeling) (Дата звернення 15.11.2020)
13. Хранимые процедуры. URL: <https://metanit.com/sql/sqlserver/11.1.php> (Дата звернення 08.12.2019)

14.CREATE PROCEDURE (Transact-SQL). Microsoft. Документация по SQL. URL: <https://docs.microsoft.com/ru-ru/sql/t-sql/statements/create-procedure-transact-sql?view=sql-server-ver15> (Дата звернення 08.12.2020)



## ДОДАТОК А. ЛОГІЧНА СХЕМА СХОВИЩА ДАНИХ

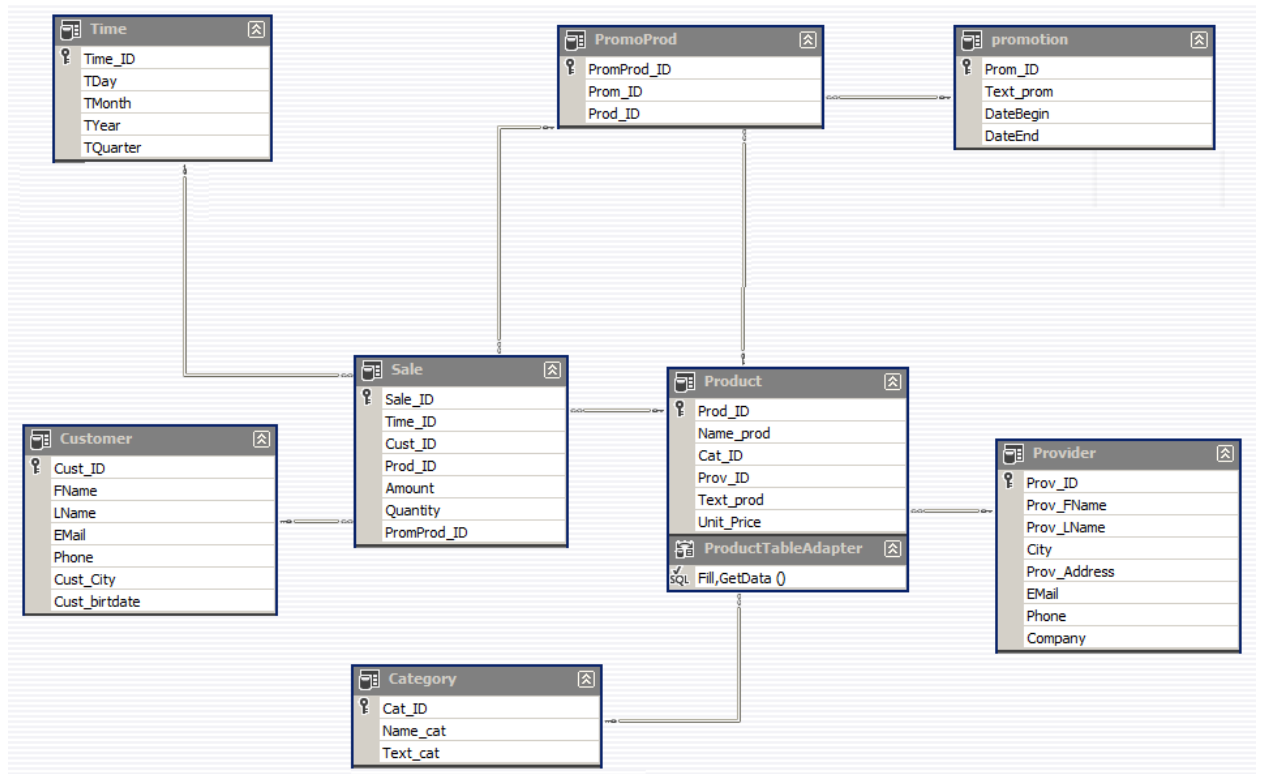


Рисунок А – Логічна схема сховища даних

## Додаток Б. ПРЕДСТАВЛЕННЯ БАЗИ ДАНИХ

```

create view TeachSumBallPartYear as
select bal11=v.SumBalPart,bal12=0,bal2=0,bal3=0,bal4=0,bal5=0,bal6=0,
v.year_exit,v.id_teach
from ViewRatSumNoName v
where id_part=1
union
select bal11=0,bal12=v.SumBalPart,bal2=0,bal3=0,bal4=0,bal5=0,bal6=0,
v.year_exit,v.id_teach
from ViewRatSumNoName v
where id_part=2
union
select bal11=0,bal12=0,bal2=v.SumBalPart,bal3=0,bal4=0,bal5=0,bal6=0,
v.year_exit,v.id_teach
from ViewRatSumNoName v
where id_part=3
union
select bal11=0,bal12=0,bal2=0,bal3=v.SumBalPart,bal4=0,bal5=0,bal6=0,
v.year_exit,v.id_teach
from ViewRatSumNoName v
where id_part=4
union
select bal11=0,bal12=0,bal2=0,bal3=0,bal4=v.SumBalPart,bal5=0,bal6=0,
v.year_exit,v.id_teach
from ViewRatSumNoName v
where id_part=5
union
select bal11=0,bal12=0,bal2=0,bal3=0,bal4=0,bal5=0,bal6=v.SumBalPart,
v.year_exit,v.id_teach
from ViewRatSumNoName v
where id_part=6
GO

CREATE view PerCat as
select p.*, categ =case when categ is not null then categ else '-' end
from Periodical p left outer join (select id_per,categ from PeriodicCategory c
where categ in ('A','B','A','B') and (dateFrom is null or (DateTo is null or
Year(dateTo)<Year(getdate()))))pc
on p.id_per=pc.id_per
GO

CREATE view PerScopus as
select p.*, IsS=case when categ='S' then '+' else ' ' end
from Periodical p left outer join (select id_per,categ from PeriodicCategory c
where categ='S' and (dateFrom is null or (DateTo is null or Year(dateTo)<Year(getdate()))))pc
on p.id_per=pc.id_per
GO

CREATE view PerWoS as
select p.*, IsWoS=case when categ='W' then '+' else ' ' end
from Periodical p left outer join (select id_per,categ from PeriodicCategory c
where categ='W' and (dateFrom is null or (DateTo is null or Year(dateTo)<Year(getdate()))))pc
on p.id_per=pc.id_per
GO

create view PublTypeInRating as
select title, number, bal, id_rat from Rating r
where id_rat in(select id_rat from PublType)
GO

create view ViewPublTypeRequir as
select pt.*, number
from PublType pt left outer join Requirements rq on pt.id_req=rq.id_req
GO

```

```
Create View ViewSpec_PerSpec as
SELECT      code, name_spec, id_per, Speciality.id_spec
FROM        PerSpec INNER JOIN
            Speciality ON PerSpec.id_spec = Speciality.id_spec
GO
```

```
create view ViewRatSumNoName as
select id_teach, number, SumBalPart=SUM(SumBal),
       year_exit,rp.id_part
from RatingPart rp join ViewGroupPart vg on rp.id_part=vg.id_part
group by id_teach, number, year_exit,rp.id_part
GO
```

## ДОДАТОК В. ВИХІДНИЙ КОД ЗБЕРЕЖЕНИХ ПРОЦЕДУР ТА ФУНКЦІЙ

```

CREATE proc DoAgregats
  @year_do smallint,
  @m bit,
  @k tinyint output,
  @msg varchar(200) output
AS
begin
  declare @@n smallint, @@number tinyint

  set @@n =(select k=count(*) from Sale_Cust where Time2_ID in
  (select Time2_ID from Time2 where TYear= @year_do)
  if @@n=0
  begin
    insert into SaleCustMonth
    select Cust_ID, TMonth, TYear, SumAmount=SUM(Amount*Quantity)
    from Sale
    where TYear=@year_do
    group by Cust_ID, TMonth, TYear
  end
  else
  begin
    set @@number=(select m=MAX(SMonth) from SaleCustMonth where TYear=@year_do)

    insert into SaleCustMonth
    select Cust_ID, TMonth, TYear, SumAmount=SUM(Amount*Quantity)
    from Sale
    where TYear=@year_do and TMonth>@@number
    group by Cust_ID, TMonth, TYear
  end
end
GO

```