

ЛАБОРАТОРНА РОБОТА №1

Тема

Відображення моделі даних в інструментальних засобах ERwin та Oracle Data Modeler

Мета роботи

Отримати загальні уявлення про ERwin та Oracle Data Modeler. Навчитися відображати моделі даних в інструментальних засобах ERwin та Oracle Data Modeler

Теоретичні відомості

Короткий огляд CA ERwin Data Modeler Standard Edition r9.5. CA ERwin Data Modeler Standard Edition - це провідне в галузі рішення для моделювання даних, що дозволяє управляти корпоративними даними за допомогою зручного графічного інтерфейсу.

Рішення CA ERwin Data Modeler допомагає організаціям керувати складною інфраструктурою даних за допомогою таких ключових функцій:

- Візуалізація складних структур даних.
 - Автоматичне створення проектів баз даних за допомогою графічних моделей даних.
 - Визначення стандартів для мінімізації надмірності. • Засоби порівняння моделей і баз даних.
 - Інтеграція і обмін метаданими з іншими засобами.
- Пов'язані продукти і рішення.
- CA ERwin Data Modeler Workgroup Edition: середа спільного моделювання даних для робочих груп.
 - CA ERwin Data Modeler Navigator Edition: доступ до моделей даних тільки для читання.
 - CA ERwin Data Modeler for MS SQL Azure: для управління середовищами MS SQL Azure CA ERwin Data Modeler Standard Edition.
 - CA ERwin Saphir Option: управління метаданими і створення моделей для ERP-систем.

- Веб-інтерфейс CA ERwin Web Portal: для обміну метаданими з широким діапазоном користувачів.

Підтримувані середовища CA ERwin Data Modeler Standard Edition працює в наступних середовищах:

- Windows 8
- Windows 10

і підтримує такі середовища баз даних:

- DB2, в тому числі DB2 для i5 / OS (System I)
- IDS (Informix)
- MySQL
- ODBC
- ORACLE
- SQL Server
- Sybase
- Sybase IQ
- Teradata.

ERwin має два рівні подання моделі - логічний і фізичний.

Логічний рівень — це абстрактний погляд на дані, коли дані представляються так, як виглядають у реальному світі, і можуть називатися так, як вони називаються в реальному світі, наприклад, "Постійний клієнт", "Відділ" або "Прізвище співробітника".

Об'єкти моделі, що представляються на логічному рівні, називаються сутностями й атрибутами. Логічна модель даних є універсальною й ніяк не пов'язана з конкретною реалізацією СУБД.

Фізична модель даних, навпроти, залежить від конкретної СУБД, фактично будучи відображенням системного каталогу. У фізичній моделі втримується інформація про всі об'єкти БД. Оскільки стандартів на об'єкти БД не існує (наприклад, немає стандарту на типи даних), фізична модель залежить від конкретної реалізації СУБД. Отже, одній логічній моделі можуть відповідати кілька різних фізичних моделей. Якщо в логічній моделі не має значення, який конкретно тип даних має атрибут, то у фізичній моделі важливо описати всю

інформацію про конкретні фізичні об'єкти - таблиці, колонки, індекси, процедури тощо.

Інтерфейс ERwin.

Інтерфейс виконано у стилі Windows-додатків, досить простий і інтуїтивно зрозумілий. Розглянемо коротко основні функції ERwin по відображенню моделі.

Кожному рівню відображення моделі відповідає своя палітра інструментів. На логічному рівні палітра інструментів має наступні кнопки:

- кнопку покажчика (режим миші) - у цьому режимі можна встановити фокус на якому-небудь об'єкті моделі;
- кнопку внесення сутності;
- кнопку категорії (категорія, або категоріальний зв'язок, - спеціальний тип зв'язку між сутностями, що буде розглянута нижче);
- кнопку внесення текстового блоку;
- кнопку перенесення атрибутів усередині сутностей і між ними;
- кнопки створення зв'язків: ідентифікуючий, "багато-до-багатьох" і неідентифікуючий.

На фізичному рівні палітра інструментів має: • замість кнопки категорій - кнопку внесення подань (view);

- замість кнопки зв'язку " багато-до-багатьох " - кнопку зв'язків подань.

Для створення моделей даних в ERwin можна використати дві нотації: IDEFIX і IE (Information Engineering). Надалі буде розглядатися нотація IDEFIX.

ERwin має кілька рівнів відображення діаграми: рівень сутностей, рівень атрибутів, рівень визначень, рівень первинних ключів і рівень іконок.

Перемкнутися між першими трьома рівнями можна з використанням кнопок панелі інструментів. Перемкнутися на інші рівні відображення можна за допомогою контекстного меню, що з'являється, якщо "кликнути" по будь-якому місцю діаграми, не зайнятому об'єктами моделі. У контекстному меню варто вибрати пункт Display Level і потім - необхідний рівень відображення.

Короткий огляд Oracle SQL Developer Data Modeler

Oracle SQL Developer Data Modeler (далі — Oracle Data Modeler) — це зручний інструмент для візуального проектування та управління модельними

структурами баз даних. Він дає змогу проєктувати, моделювати, документувати та впорядковувати бази даних на логічному та фізичному рівнях, використовуючи сучасні підходи та нотації.

Основні можливості Oracle Data Modeler

- **Візуалізація** складних структур даних у різних нотаціях (IDEF1X, Barker тощо).
- **Автоматичне створення** схеми бази даних на основі графічної моделі.
- **Визначення стандартів** та правил для атрибутів, типів даних і зв'язків, що сприяє зменшенню надлишковості даних.
- **Засоби порівняння моделей** і поточних схем баз даних, синхронізація змін.
- **Імпорт та експорт** моделей, інтеграція з іншими інструментами та обмін метаданими.

Підтримувані середовища

Oracle Data Modeler зазвичай використовується разом із **Oracle SQL Developer**, але може працювати і як окремий застосунок. Підтримуються такі операційні системи:

- Microsoft Windows
- Linux
- macOS

А також широкий спектр СУБД, зокрема:

- Oracle Database
- Microsoft SQL Server
- MySQL
- IBM DB2
- PostgreSQL
- Teradata
- І будь-які інші джерела через стандартні JDBC-драйвери

Логічна та фізична модель даних

Як і більшість професійних інструментів моделювання, Oracle Data Modeler дає змогу працювати на **логічному** та **фізичному** рівнях.

1. **Логічний рівень** — це уявлення даних з точки зору предметної області, без прив'язки до конкретної СУБД. На цьому рівні ми працюємо з *сутностями* (entity) та *атрибутами* (attribute), використовуючи зрозумілі назви (наприклад, «Факультет», «Кафедра», «Прізвище» тощо).
2. **Фізичний рівень** — це проєкція логічної моделі на конкретну СУБД, з урахуванням типів даних, підтримуваних об'єктів (таблиць, індексів, процедур тощо) та синтаксичних особливостей. Одній логічній моделі можуть відповідати декілька фізичних моделей, якщо потрібно реалізувати її в різних СУБД.

Інтерфейс Oracle Data Modeler

Oracle Data Modeler має типовий графічний інтерфейс із розподілом на панелі та вкладки:

- **Browser** (Провідник) — для огляду об'єктів моделі (від сутностей і атрибутів до конкретних типів даних, зв'язків та інших артефактів).
- **Diagram window** (Вікно діаграми) — у ньому безпосередньо розміщуються сутності, зв'язки та інші графічні елементи.
- **Properties** (Властивості) — детальні налаштування обраного об'єкта (сутності, атрибуту, зв'язку тощо).
- **Палітра інструментів** (графічні кнопки) для створення, редагування та зв'язування сутностей.

Залежно від того, у якому режимі (Logical або Relational/Physical) ви працюєте, набір інструментів та відображення можуть дещо змінюватися.

Основні інструменти створення моделі (логічний рівень)

- Додавання нової **сутності** (Entity).
- Додавання **атрибутів** (Attributes) у сутність.
- Створення **зв'язків** (Relationships) типу 1:1, 1:N (ідентифікуючий та неідентифікуючий), M:N (багато-до-багатьох).
- Встановлення **ідентифікаторів** (Primary Key) і визначення зовнішніх ключів (Foreign Key).
- Додавання **текстових пояснень** (Notes) та інших допоміжних графічних елементів.

Приклад побудови логічної моделі даних

Розглянемо аналогічний приклад, що демонструє формальне подання даних для задачі складання розкладу в університеті. Текстове подання (опис атрибутів) може бути таким:

FACULTY (#F, Name, Dean, Building, Fund)

DEPARTMENT (#D, #F, Name, Head, Building)

GROUP (#G, #D, Course, Number, Quantity)

LECTURE (#T, #G, #S, #R, Type, Day, Week)

Subject (#S, Name)

ROOM (#R, Number, Building, Seats)

TEACHER (#T, #D, Name, Post, Tel)

Де символ «#» перед назвою атрибуту вказує на ключовий атрибут (Primary Key), а посилання на ключ з іншої сутності означає зовнішній ключ.

Крок 1: Створення сутностей і їх атрибутів

- У **Browser** або за допомогою **палітри інструментів** створіть нові сутності:
 - **FACULTY (Факультет)**
 - Атрибути: F (PK), Name, Dean, Building, Fund
 - **DEPARTMENT (Кафедра)**
 - Атрибути: D (PK), F (FK), Name, Head, Building
 - **GROUP (Студентська група)**
 - Атрибути: G (PK), D (FK), Course, Number, Quantity
 - **LECTURE (Лекція)**
 - Атрибути: T (PK?), G (FK), S (FK?), R (FK?), Type, Day, Week
 - **SUBJECT (Предмет)**
 - Атрибути: S (PK), Name
 - **ROOM (Аудиторія)**
 - Атрибути: R (PK), Number, Building, Seats
 - **TEACHER (Викладач)**
 - Атрибути: T (PK), D (FK), Name, Post, Tel
- Для кожної сутності у властивостях налаштуйте **Primary Key (PK)**, як-от F для FACULTY, D для DEPARTMENT тощо.
- Інші атрибути додайте просто як *non-key attributes*.

4. Розташуйте сутності у **Diagram window** зручним для вас способом (як на рис.1 та рис.2 у прикладі).

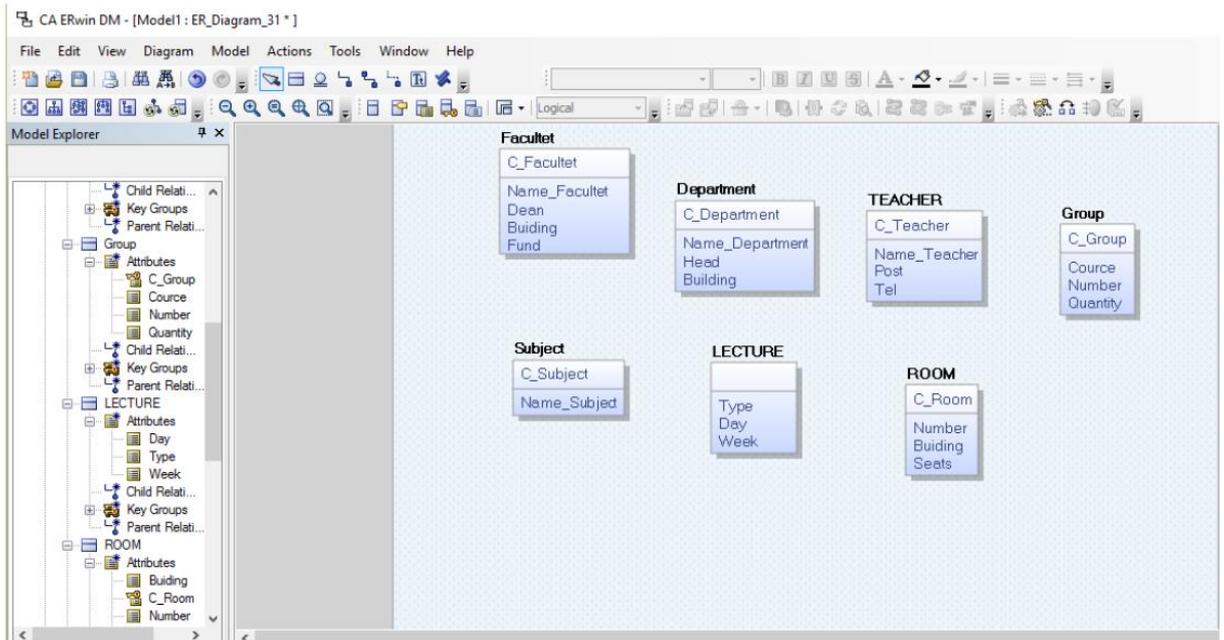


Рис. 1 Перший крок побудови логічної моделі у ERwin

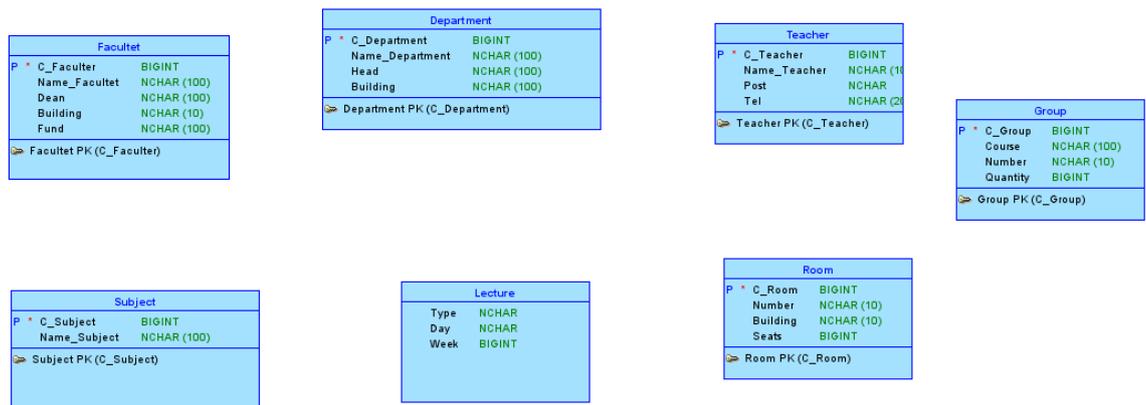


Рис. 2 Перший крок побудови логічної моделі у Oracle Data Modeler

У **ERwin** при визначенні зв'язків мігрують ключові атрибути із сутностей верхнього рівня у сутності нижнього рівня. В останніх вони стають зовнішніми атрибутами (FK). При цьому, якщо використовується ідентифікуючий зв'язок, зовнішні атрибути потрапляють у групу ключових, якщо ж використовується не ідентифікуючий зв'язок, вони потрапляють у нижню секцію відповідного прямокутника.

В **Oracle SQL Developer Data Modeler** підхід до відображення ключових атрибутів (первинних ключів) дещо інший, ніж ERwin. Залежно від обраної

нотації (IDEF1X, Barker тощо) та налаштувань відображення, первинні ключі можуть:

- відображатися вгорі списку атрибутів сутності у **спеціальному розділі** (наприклад, “Key” або “Primary UID”),
- виділятися **жирним шрифтом** або позначатися **іконою ключа**,
- мати додаткові підписи чи позначення, що вказують на їх роль як унікальних ідентифікаторів сутності.

Таким чином, у графічному поданні **логічної моделі** в Oracle Data Modeler зазвичай можна побачити, що ключові атрибути відокремлено від решти або ж позначено спеціальними позначками (ключем). Це означає, що вони відповідають первинним ключам у реляційній моделі й **однозначно ідентифікують** кожний екземпляр сутності. Усі інші (не ключові) атрибути розташовуються у наступному розділі або нижче в тому самому переліку.

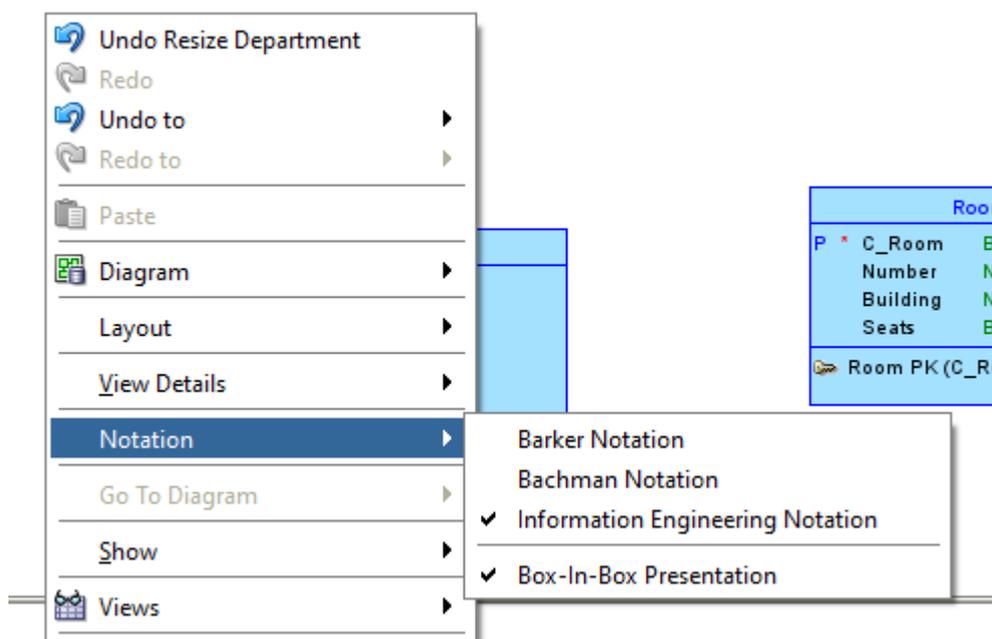


Рис. 3 Вибір нотації

Зверніть увагу, що логічна модель ще не зобов’язує вас до певних типів даних — достатньо визначити атрибути і ключі.

Крок 2: Визначення зв’язків між сутностями

1. За допомогою **інструмента зв’язків** в Oracle Data Modeler встановіть зв’язки між сутностями:

- FACULTY → DEPARTMENT (1:N): Одному факультету відповідає кілька кафедр, у DEPARTMENT з'явиться зовнішній ключ F.
 - DEPARTMENT → GROUP (1:N): Одній кафедрі відповідає кілька груп, у GROUP буде D.
 - GROUP → LECTURE (1:N) разом із SUBJECT і ROOM: у LECTURE зберігаються ключі G, S, R.
 - DEPARTMENT → TEACHER (1:N): у TEACHER зберігається D.
2. Якщо ви встановлюєте **ідентифікуючий зв'язок (Identifying relationship)**, атрибут із «батьківської» сутності автоматично стає частиною первинного ключа (PK) у «дочірній» сутності. При **неідентифікуючому зв'язку (Non-Identifying)** цей атрибут потрапляє до звичайних (Foreign Key) атрибутів.
 3. Перевірте, як ключові атрибути мігрують: якщо зв'язок 1:N, то ключ «одиночної» сутності стає зовнішнім ключем у сутності з боку «багатьох».

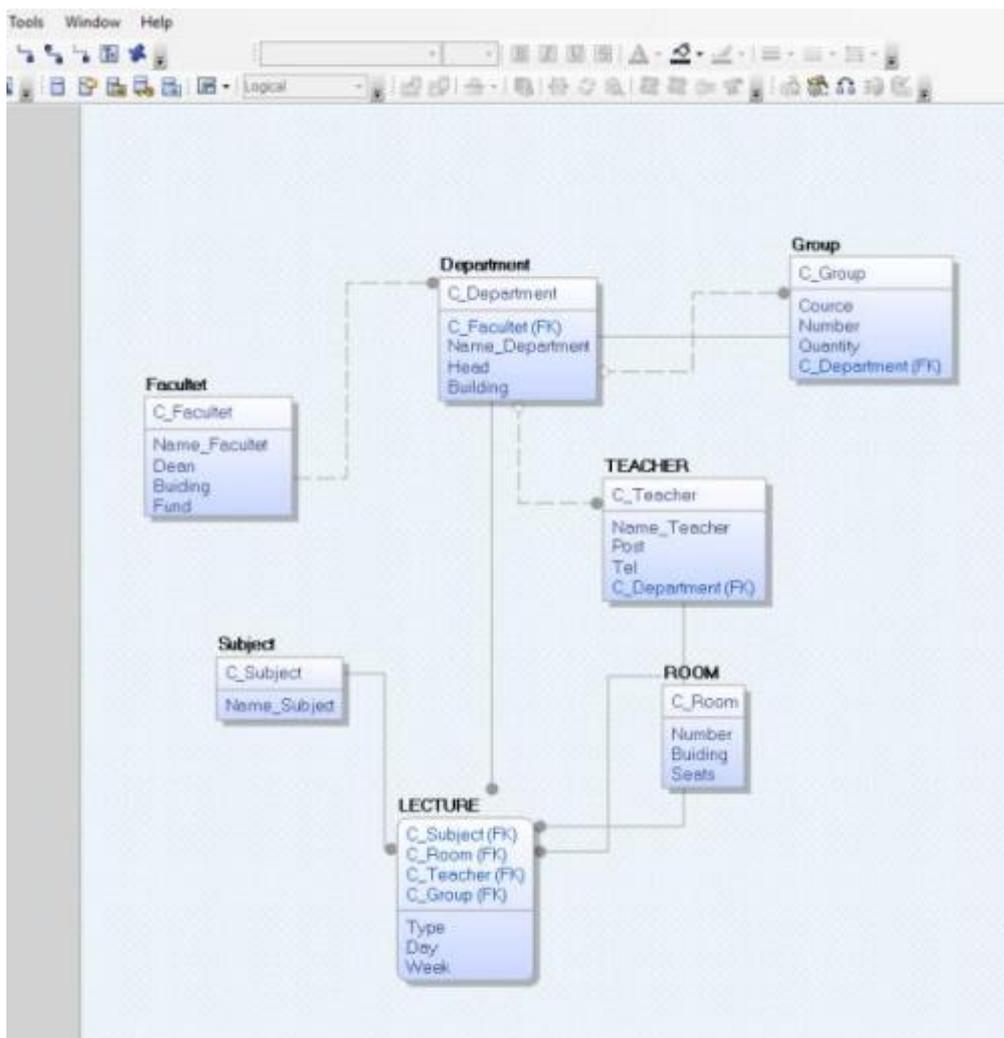


Рис. 4 Повна логічна модель – ER- діаграма у ERwin

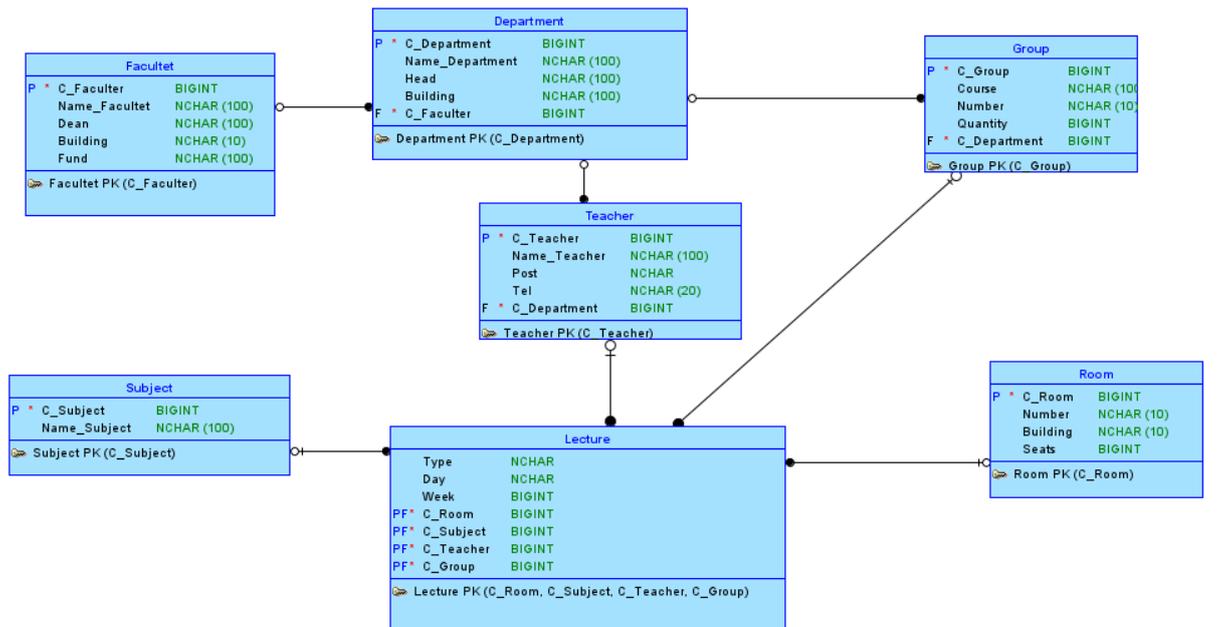


Рис. 5 Повна логічна модель – ER- діаграма у Oracle Data Modeler

У результаті маєте повну **ЛОГІЧНУ МОДЕЛЬ** (ER-діаграму), де кожна сутність має свої атрибути, а зв'язки відображають правила підпорядкованості або співвідношення між даними.

Поради та додаткові кроки

- **Назви об'єктів:** на логічному рівні використовуйте назви, зручні для розуміння бізнес-користувачами (наприклад, україномовні або англійські, залежно від команди).
- **Підготовка до фізичної моделі:** можна одразу прописати орієнтовні типи даних, але пам'ятайте, що на логічному рівні це не обов'язково. Пізніше ви зможете обрати конкретний тип при генерації фізичної моделі для цільової СУБД.

Хід виконання роботи

1. Побудуйте діаграму, яка моделює дані для задачі сертифікації нових сортів рослин:
 - **Культура** (#К, Назва_укр., Назва_лат.)
 - **Сорт** (#С, #К, Опис)
 - **Кліматична_зона** (#З, Назва, Опис)
 - **Станція** (#Ст, #З, Назва, Адреса)
 - **Випробування** (#Ст, #С, #К, Рік, Результат)
2. Опишіть всі типи зв'язків між сутностями та вкажіть їхню специфіку.