

Лабораторна робота № 1

Тема. Проектування власної БД на зовнішньому, інфологічному та даталогічному рівнях.

Мета. Систематизація та застосування знань з теорії проектування БД. Усвідомлення ролі кожного рівня проектування БД. Формування вмінь та навичок проектування власної БД на різних рівнях.

Теоретичні відомості

База даних (БД) – це поіменована структурована сукупність відомостей про об'єкти певної предметної області. БД розробляють для прискорення задоволення інформаційних потреб користувача (скажімо, визначення вільних місць на конкретний сеанс в кінотеатрі чи на обраний рейс потяга Укрзалізниці). Концепція застосування БД базується на однократному введенні даних і багатократному їх використанні. Існують не лише комп'ютерні БД (наприклад, прайс-листи чи бібліотечні каталоги), але саме електронні БД дають змогу максимально швидко перетворювати вхідну інформацію у вихідну.

Об'єкти предметної області характеризуються своїми властивостями (наприклад, для кожного співробітника організації можна вказати прізвище, ім'я, по батькові, ідентифікаційний номер, дані паспорта, домашню адресу). У свою чергу, властивості об'єкта складаються з неподільних атрибутів (наприклад, з властивості *Дані паспорта* доцільно виділити атрибути *Серія паспорта*, *Номер паспорта*, *Дата видачі паспорта*, *Ким видано паспорт*). В БД зберігаються значення лише тих атрибутів, які дають змогу відрізнити один об'єкт від іншого. Це зменшує розмір БД і, як наслідок, прискорює їх обробку.

На практиці найчастіше використовують реляційні (табличні) БД. В них дані однотипних об'єктів групують по окремих таблицях (сутностях), а логічно пов'язані атрибути різних таблиць поєднують зв'язками. Кожен рядок (запис, кортеж) реляційної таблиці описує один однотипний об'єкт, а кожен стовбець (поле) містить значення одного атрибута для всіх таких об'єктів. Приклад фрагменту реляційної таблиці наведено на рис. 1, а перелік її атрибутів міститься у заголовках полів.

Чітка структуризація даних реляційної таблиці з одного боку прискорює їх обробку, а з іншого – дає змогу отримати однозначні результати опрацювання. Наприклад, при пошуку співробітників, які проживають на вулиці *Кавказька*, співставлення значень буде здійснюватися лише для поля *Вулиця проживання* кожного запису (а не для всіх полів) і тому не буде віднайдено співробітниць з прізвищем *Кавказька*.

КодТовару	Назва товару	Марка товару	Одиниця вимірювання
1	згущене молоко 'Бахмач'	Бахмач	шт.
2	майонез 'Делікатесний'	Торчин продукт	шт.
3	цукерки 'Червоний мак'	Світоч	кг
4	цукерки 'Столичні'	ЕКО продукт	кг
5	цукерки 'Зоряне сйво'	ЕКО продукт	кг
6	кава 'Галка'	Львівська кава	шт.

Рис. 1. Фрагмент реляційної таблиці

Для організації взаємодії з електронними БД використовують системи управління базами даних (СУБД), наприклад, такі, як Oracle, MS Access чи SQL Server. **СУБД** – це комплекс програмних і мовних засобів загального і спеціального призначення необхідний для створення БД, підтримування її в актуальному стані, організації маніпулювання даними та доступу до них різних користувачів. Використання СУБД дає змогу розробнику зосередитися на створенні прикладних програм і обробляти дані реляційних таблиць на рівні окремих полів, не замислюючись над тим, де і як ці дані фізично зберігаються у файлах БД. Структура даних реляційних таблиць разом з даними зберігається в БД і автоматично використовується СУБД, що дозволяє вносити в неї зміни без внесення корегувань в прикладні програми, а новим програмам – використовувати накопичені раніше дані (рис. 2). Структура даних входить у **словник даних**, де також можуть міститися відомості про предметну область, користувачів, їх права доступу та журнал внесення змін. Фактично, опрацювання записів БД виконується СУБД в два проходи: на першому – зчитується структура даних, а на другому – відбувається їх обробка згідно зчитаної структури.

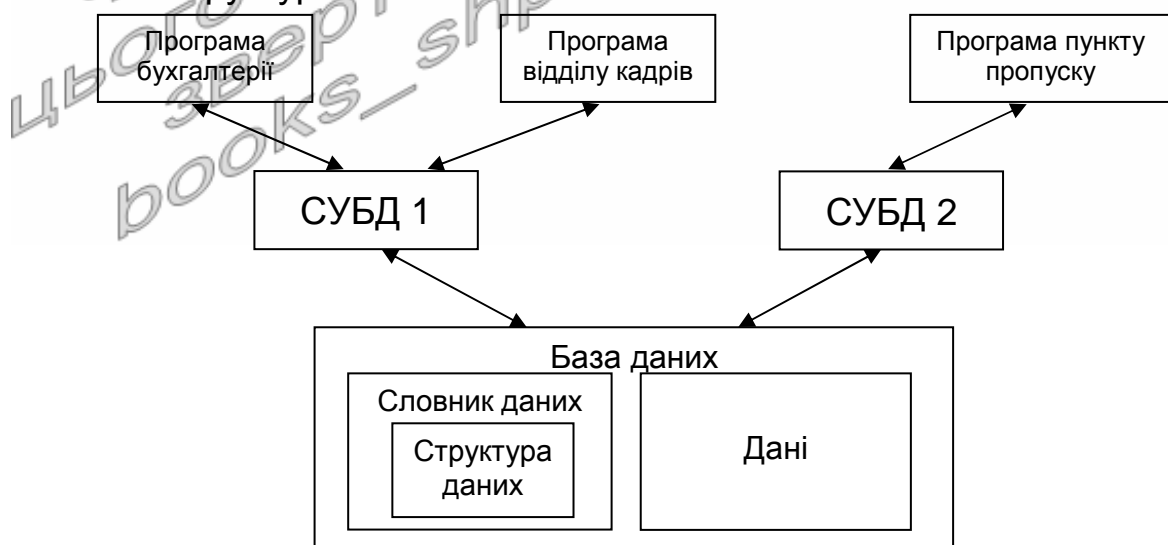


Рис. 2. Організація обміну даними при незалежній структурі БД від прикладних програм

Ми будемо розробляти електронні БД та прикладні програми для їх обробки за допомогою СУБД MS Access, оскільки вона дозволяє опрацьовувати дані в інтерактивному режимі без безпосереднього застосування програмних кодів.

Але виділення окремих атрибутів об'єктів, зведення їх до табличного вигляду та опису фізичного розміщення даних недостатньо для забезпечення однократного введення всіх даних. Розглянемо, наприклад, фрагмент даних таблиці обліку замовлених товарів (рис. 3). При її використанні у випадку замовлення клієнтом декількох товарів (записи 2-5) крім самого товару, його кількості та ціни доводиться щоразу вказувати відомості про того самого клієнта, співробітника, який прийняв замовлення та дату замовлення. Таке **дублювання** даних призводить до **надлишкових** витрат пам'яті і може призвести до **неузгодженості** інформації у різних записах таблиці (наприклад, дані клієнта у п'ятому записі відрізняються від трьох попередніх). Тому під час проектування БД після виділення окремих атрибутів необхідно усунути ці **три аномалії ненормалізованих відношень**.

Співробітник	Клієнт	Дата зам	Назва товару	Кільк	Ціна
Мулявка Снепа	Листопад Євге	03.01.2016	згущене молоко	3	12,50 грн.
Кравцов Петро	Кликов Михай.	05.01.2016	цукерки 'Столич	1	100,00 грн.
Кравцов Петро	Кликов Михай.	05.01.2016	кава 'Галка'	3	50,00 грн.
Кравцов Петро	Кликов Михай.	05.01.2016	морозиво 'Плом	5	5,00 грн.
Кравцов Петро	Дудяк Олекса	05.01.2016	цукерки 'Черво	1	80,00 грн.
Андрюшенко Л	Тітова Тетяна А	06.01.2016	цукерки 'Столич	1	100,00 грн.

Рис. 3. Структура та фрагмент даних реляційної таблиці з дублюваннями, надлишковостями та неузгодженостями

Отже, проектування БД загалом виконують на чотирьох рівнях:

1. На **зовнішньому** рівні визначають перелік задач, котрі будуть розв'язуватися з використанням прикладної програми, та перелік атрибутів, які необхідно зберігати в БД. Таке проектування найчастіше виконує постановник.
2. На **інфологічному** рівні виділяють типові інформаційні об'єкти (сутності) та встановлюють зв'язки між ними з метою мінімізації дублювань, надлишковостей та неузгодженостей ще на етапі проектування, до використання СУБД для створення електронної БД. Найчастіше проектування на цьому рівні виконують проектні менеджери.
3. При проектуванні на **даталогічному** рівні на основі інфологічної моделі будують інформаційно-логічну модель предметної області з урахуванням особливостей та обмежень обраної СУБД. Тут формують концептуальну модель даних, визначають їх типи та характеристики. Як правило, виконують таке проектування розробники програмного забезпечення.

4. На **внутрішньому** рівні формують опис фізичного розміщення даних в пам'яті ПК: визначають формати записів, порядок їх фізичного впорядкування, характеристики та шляхи доступу до даних. Від параметрів фізичної моделі залежить обсяг пам'яті, який займає БД та час реакції системи. Параметри фізичної моделі можуть змінюватися в процесі експлуатації, але ці зміни не повинні зумовлювати зміни на вищих рівнях. І взагалі, зміни в проектах нижчих рівнів не повинні викликати зміни проектів вищих рівнів. Проектування на внутрішньому рівні виконують розробники СУБД та адміністратор. В Access БД та засоби для їх обробки найчастіше зберігаються в одному файлі, тому нам виконувати проектування БД на внутрішньому рівні не потрібно.

В процесі проектування БД розробляють лише структуру таблиць та визначають зв'язки між ними, а дані в таблиці вносять лише після фізичного створення БД в середовищі обраної СУБД. Для прискорення проектування БД на сьогодні широко використовують спеціалізовані інструментальні засоби (наприклад, ERwin), які дають змогу зберігати проміжні результати проектування та автоматизують перехід між його рівнями. Детально основні етапи проектування на перших трьох рівнях описані в завданнях та прикладі виконання цієї лабораторної роботи. Тут наголосимо лише на принципах виділення окремих таблиць (сутностей) і встановлення зв'язків між ними на інфологічному рівні та на виборі типів даних для окремих полів на даталогічному рівні проектування.

В основі процедури агрегації атрибутів по окремих сутностях на інфологічному рівні лежить аналіз співвідношень між ними. Розрізняють чотири типи співвідношень між атрибутами:

1. Атрибути $A1$ і $A2$ перебувають у співвідношенні $1:1$ (один до одного), якщо кожному представнику атрибута з сукупності $A1$ відповідає не більше одного представника атрибута з сукупності $A2$ і навпаки (рис. 4а). Якщо між атрибутами виявлено співвідношення $1:1$, то такі атрибути відносять до однієї таблиці та присвоюють їй назву. В кожній таблиці виділяють **первинний ключ** – це поле (рідше – сукупність полів), яке однозначно ідентифікує описаний об'єкт предметної області. Таке поле має обов'язково бути унікальним і непорожнім. При використанні об'єкта вказують лише значення з його ключового поля. Наприклад, в таблиці *Товари* (див. рис. 1) ключовим є поле *КодТовару*, яке однозначно ідентифікує окремий товар. Тому для опису замовлень вказаного товару в таблицях обліку замовлень достатньо зберігати лише код цього товару.

Наголосимо, що співвідношення встановлюються між кількостями, а не між значеннями атрибутів, адже, наприклад, декілька співробітників можуть народитися в один і той самий день, але кожен з них має одне прізвище і одну дату народження, тому й співвідношення між цими атрибутами $1:1$.

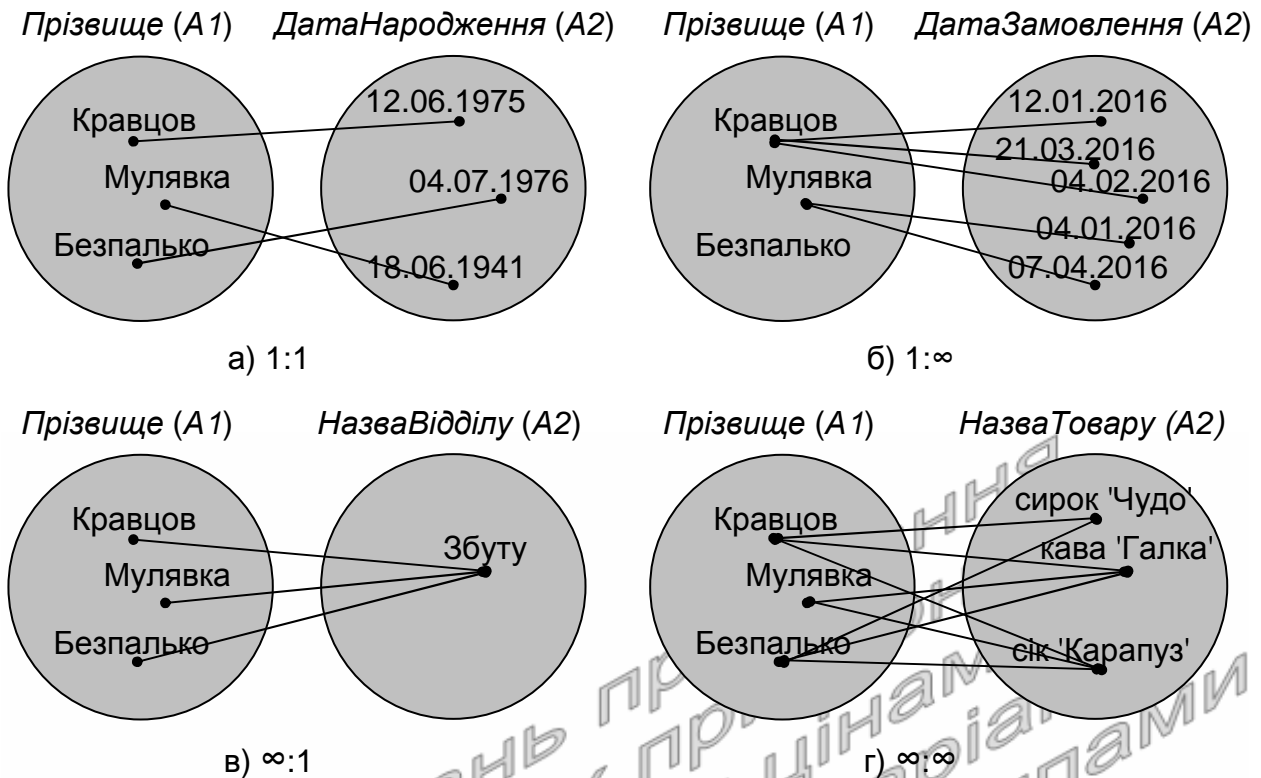


Рис. 4. Різновиди співвідношень між атрибутами

2. Атрибути $A1$ і $A2$ перебувають у співвідношенні $1:\infty$ (один до багатьох), якщо кожному представнику атрибута з сукупності $A1$ відповідає нуль або декілька представників з сукупності $A2$, але кожному представнику атрибута з сукупності $A2$ відповідає не більше одного представника з сукупності $A1$. Наприклад, один співробітник може обслуговувати багато замовлень (рис. 4б), але за кожне замовлення відповідає лише один визначений співробітник. У випадку виявлення між атрибутами співвідношення $1:\infty$ їх розміщують в різних таблицях і між цими таблицями встановлюють структурний зв'язок. Зв'язок встановлюється між полями з однаковим змістовим навантаженням, причому на стороні 1 , як правило, знаходиться первинний ключ, який ідентифікує об'єкт, а на стороні ∞ – числове поле, яке вказує на використання обраного об'єкта (**вторинний ключ**, значення якого в таблиці на стороні ∞ може повторюватися). Тобто для опису взаємодії об'єкта з іншим об'єктом в таблиці на стороні ∞ зберігають лише значення ключового поля цього об'єкта з таблиці на стороні 1 . Наприклад, в таблиці *ОблікЗамовлень* (див. рис. 3) атрибути *КодКлієнта* і *КодТовару* перебувають у співвідношенні $1:\infty$, оскільки в одному замовленні клієнта може міститися декілька замовлених товарів. Тому цю таблицю необхідно розбити на дві таблиці – *ЗаголовкиЗамовлень* і *ПунктиЗамовлень* (рис. 5). Перша з них має фіксувати факт оформлення замовлення, а друга – містити дані про товари кожного замовлення. Інший приклад: як зазначалося вище,

атрибути *ПІБ* (співробітника) та *КодЗамовлення* перебувають у співвідношенні 1:∞ (див. рис. 4б). Тому ці атрибути мають міститися в різних таблицях (на рис. 5 – відповідно *Співробітники* та *ЗаголовкиЗамовлень*). Ці таблиці пов'язані по полях *КодСпівробітника*. В таблиці *Співробітники* це поле ідентифікує кожного співробітника, а в таблиці *ЗаголовкиЗамовлень* – вказує на того співробітника, який оформив конкретне замовлення.

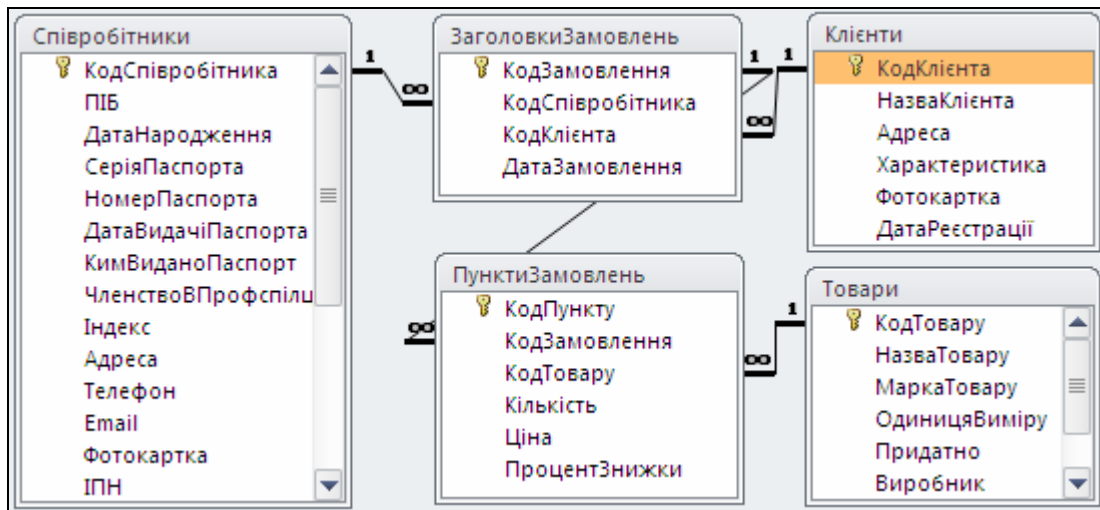


Рис. 5. Структура таблиць для обліку замовлень та пов'язаних об'єктів

3. Співвідношення $\infty:1$ (рис. 4в) формулюється як обернене до попереднього. Воно реалізується структурним зв'язком $\infty:1$ між двома таблицями (на рис. 5 такий зв'язок встановлений між таблицями *ЗаголовкиЗамовлень* та *Клієнти* і *ПунктиЗамовлень* та *Товари*).
4. Атрибути *A1* і *A2* перебувають у співвідношенні $\infty:\infty$ (багато до багатьох), якщо кожному представнику атрибута з сукупності *A1* відповідає нуль або декілька представників з сукупності *A2* і кожному представнику атрибута з сукупності *A2* відповідає нуль або декілька представників з сукупності *A1*. При виявленні між атрибутами співвідношення $\infty:\infty$ їх розміщують в різних таблицях і створюють ще й одну чи декілька проміжних таблиць, між якими встановлюють структурні зв'язки $1:\infty$. Наприклад, між атрибутами *Прізвище* (співробітника) і *НазваТовару* існує співвідношення $\infty:\infty$, оскільки кожен співробітник може оформляти замовлення на різні товари і кожен товар може замовлятися у різних співробітників (рис. 4г). Тому ці атрибути розміщують в різних таблицях (на рис. 5 – відповідно *Співробітники* та *Товари*), які пов'язуються через дві проміжні таблиці *ЗаголовкиЗамовлень* та *ПунктиЗамовлень*.

Після виділення сутностей та структурних зв'язків між ними (як на рис. 5) виконують їх перевірку на відповідність умовам нормалізації, з основними положеннями якої ми пропонуємо читачу ознайомитися самостійно [4, с. 60-67].

Вслід за проектуванням на інфологічному рівні основним етапом проектування на даталогічному рівні є визначення **типу даних** для кожного поля з переліку, пропонованого обраною СУБД. Саме тип даних в основному визначає дані, які можуть зберігатися в полі. В Access застосовуються типи даних, наведені в табл. 1.

Таблиця 1

Типи даних СУБД MS Access та їх призначення

Тип даних	Характеристика даних
Счетчик	Числове унікальне і непорожнє поле-лічильник, значення якого генеруються автоматично. Найчастіше використовується для первинних ключів
Текстовый / Короткий текст	Короткий текст (до 255 символів), який зберігається безпосередньо в таблиці
Поле MEMO / Длинный текст	Довгий текст, що зберігається поза таблицею, а в самій таблиці міститься лише посилання на нього
Числовой	Як цілі, так і дійсні числа
Денежный	Дійсні числа з максимум чотирма цифрами після коми
Дата/время	Дійсне число, ціла частина якого вказує на кількість днів, які минули до введеної дати, а дробова – на частину доби, що минула до вказаного часу
Логический	Може містити одне з двох значень – <i>Істинно</i> (-1) чи <i>Хибно</i> (0)
Поле объекта OLE	Вбудований OLE-об'єкт (наприклад, графічний, аудіо чи відео фрагмент). Зберігається за принципом MEMO-поля
Гиперссылка	Містить текст для користувача, шлях та місце у файлі чи на сторінці, на яке вказує посилання
Вложение	Один чи декілька вкладених файлів. Зберігаються за принципом MEMO-поля
Вычисляемый	Містить результати обчислень над іншими полями таблиці, застосовується для прискорення розрахунків

Лише після визначення типу даних для кожного атрибута всіх таблиць доцільно приступати до створення БД в середовищі обраної СУБД згідно розробленого проекту (як на рис. 5).

Література: [4, С. 5-80; 5, С. 464-471]

Підготовчий етап заняття

1. Ознайомтеся з завданнями та прикладом виконання лабораторної роботи. Оберіть тему з наведеної нижче тематики для проектування власної БД та зареєструйте її у викладача.

Тематика для розробки власного додатку обробки БД

Створення інформаційної системи...

- | | |
|---------------------------|--|
| 1. Служби зайнятості. | 9. Автозаправної станції. |
| 2. Абонементу бібліотеки. | 10. Деканату ВУЗу. |
| 3. Бібліотечних фондів. | 11. Кафедри ВУЗу. |
| 4. Кадрової агенції. | 12. Гуртожитку. |
| 5. Музею. | 13. Музичного відділу радіостанції. |
| 6. Центру статистики. | 14. Автостоянки. |
| 7. Архіву. | 15. Агентства з торгівлі цінними паперами. |
| 8. Проведення аукціонів. | |

16. Залізничних перевезень.
17. Автовокзалу.
18. Вантажних перевезень.
19. Авіаперевезень.
20. Метрополітену.
21. Рекламної агенції.
22. Страхової компанії.
23. Банківської установи.
24. Лікарні.
25. Поліклініки.
26. Стоматологічної поліклініки.
27. Аптеки.
28. Готелю.
29. Кінотеатру.
30. Воєнкомату.
31. Хімчистки.
32. Відділення зв'язку.

33. Поштового відділення.
34. Кур'єрської пошти.
35. Податкової адміністрації.
36. Пенсійного фонду.
37. Автомобільного заводу.
38. Диспетчерської радіотаксі.
39. Заводу з виробництва продовольчої продукції.
40. Друкарні.
41. Продовольчого магазину.
42. Промислового магазину.
43. Магазину з торгівлі косметикою.
44. Магазину з торгівлі оргтехнікою.
45. Перукарні.
46. Фотоцентру.
47. Ринку.
48. Туристичної фірми.

Проектування БД на зовнішньому рівні

2. Здійсніть проектування на зовнішньому рівні БД обраної предметної області, виконуючи наступні дії:
 - 2.1. Визначте перелік задач, що мають вирішуватися ІС обробки БД;
 - 2.2. Вивчіть та проаналізуйте первинні документи. Сформууйте перелік реквізитів документів для таблиць оперативної інформації;
 - 2.3. Вивчіть та проаналізуйте нормативно-довідкові документи. Сформууйте перелік реквізитів документів для таблиць умовно-постійної інформації;
 - 2.4. Вивчіть процес перетворення вхідної інформації у вихідну.

Проектування БД на інфологічному рівні

3. Виконайте проектування БД обраної предметної області на інфологічному рівні. Для цього:
 - 3.1. Ліквідуйте синонімію (один і той же атрибут в різних таблицях має різну назву) та омонімію (різні атрибути в різних таблицях мають однакову назву) між атрибутами. Узагальніть окремі атрибути;
 - 3.2. Виконайте агрегацію атрибутів для виділення інформаційних об'єктів (не менше восьми таблиць);
 - 3.3. Сформулюйте запити системи та опишіть хід їх розв'язання;
 - 3.4. Побудуйте графічне зображення інфологічної моделі. Зобразіть та обґрунтуйте письмово кожен структурний зв'язок;

Проектування БД на даталогічному рівні

4. Здійсніть часткове проектування БД обраної предметної області на даталогічному рівні, вказуючи для кожного атрибута, виділеного на інфологічному рівні, тип даних поля СУБД MS Access.

5. Захистіть розроблені інфологічну і даталогічну моделі перед викладачем.

Приклад проектування БД гуртового складу

1. Аналіз предметної області.

Проектування БД на зовнішньому рівні

Як відомо, основними видами діяльності будь-якої торгівельної організації, у тому числі і гуртового складу, є отримання та реалізація товарів. Завдяки різниці між цінами продажу та купівлі товарів утворюються прибутки таких організацій. Отримані в результаті діяльності оборотні кошти та чисті прибутки торгівельної організації мають бути спрямовані на придбання тих товарів, запасів яких недостатньо для задоволення реального та прогнозованого попиту. В умовах розмаїтого асортименту товарів гуртового складу ведення обліку обігу продукції без ІС займає багато часу. Ще складніше в цих умовах спрогнозувати попит клієнтів та розрахувати необхідні обсяги закупок. Саме тому при проектуванні БД та створенні ІС гуртового складу слід насамперед автоматизувати облік обігу товарів.

З метою створення проекту БД на зовнішньому рівні спочатку проаналізуємо типову накладну постачання товарів (рис. 6).

від «___» _____ 200__ р.					
Накладна № _____					
Кому _____		_____			
<small>посада</small>		П., І., Б.			
Від кого _____		_____			
<small>посада</small>		П., І., Б.			
№ з-п	Найменування	Од. вим.	К-ть	Ціна	Сума
Всього:					
Здав: _____		Прийняв: _____		_____	
<small>підпис</small>		<small>підпис</small>		П., І., Б.	

Рис. 6. Типовий бланк накладної постачання товару

Як видно з наведеного документа, в накладній постачання насамперед фіксуються реквізити постачальника та отримувача продукції, дата постачання, назва, одиниця виміру, кількість, ціна та сума кожного поставленого товару. В процесі діяльності гуртового складу від кожного постачальника може бути здійснено декілька закупок, тому їх дані доцільно зберігати в окремій таблиці та використовувати при потребі. Крім того, в накладній постачання фіксуються дані співробітника, відповідального за отримання товару. У кожному торгівельному складі є визначене коло співробітників, відповідальних за отримання товарів. Кожен з них приймає декілька постачань в день, тому їх дані доцільно зберігати в окремій таблиці та використовувати

при обліку постачань. Окремий товар постачається на гуртовий склад після того, як сукупний реальний та прогнозований попит перевищить запаси складу, тобто в процесі діяльності кожен товар може надходити декілька разів. Саме тому характеристики товарів доцільно зберігати в окремій таблиці і посилатися на них при потребі.

Так само, аналізуючи накладну відвантаження товарів, яка має вигляд, аналогічний до рис. 6, приходимо до необхідності зберігання даних клієнтів в окремій таблиці. При цьому дані співробітників та товарів мають вказуватися з тих самих таблиць, що й при формуванні постачань. Для аналізу ефективності роботи окремих відділів гуртового складу необхідно також створити окрему таблицю для зберігання їх даних і для кожного співробітника вказати посилання на відділ, де він працює.

Отже, в ІС обліку обігу товарів гуртового складу мають автоматизуватися наступні задачі:

- облік відділів;
- облік співробітників;
- облік постачальників;
- облік клієнтів;
- облік товарів;
- облік постачань товарів;
- облік замовлень товарів.

< Далі вказується перелік атрибутів для зберігання даних кожної задачі і описується процес перетворення вхідної інформації у вихідну. >

2. Проектування БД на інфологічному рівні

В процесі проектування БД обліку обігу товарів гуртового складу на інфологічному рівні нами було виділено дев'ять таблиць-сутностей. Графічне зображення структури виділених таблиць та зв'язків між ними наведено на рис. 7. При виділенні таблиць було ліквідовано синонімію і омонімію атрибутів, атрибути *Прізвище*, *Ім'я* та *ПоБатькові* в таблиці *Співробітники* узагальнено в атрибуті *ПІБ*, оскільки їх окреме опрацювання не передбачається.

Між атрибутами кожної окремої таблиці встановлено співвідношення 1:1. Окремі атрибути інформаційних об'єктів потрапили не в одну, а в декілька таблиць, але в кожній з них вони мають різне функціональне призначення:

- атрибут *КодСпівробітника* в таблиці *Співробітники* містить унікальний внутрішній ідентифікаційний номер кожного співробітника, в таблиці *ЗаголовкиПостачань* цей самий атрибут містить код того співробітника, що приймав постачання, а в таблиці *ЗаголовкиЗамовлень* – код співробітника, що виконав замовлення;
- < Аналогічно обґрунтовуються інші дублювання назв атрибутів різних таблиць >.

Для зберігання даних постачань товарів в БД виділено дві таблиці: *ЗаголовкиПостачань* та *ПунктиПостачань*. Перша з них містить дані про сам факт постачання, а друга зберігає відомості про окремі поставлені товари. Між таблицями *ЗаголовкиПостачань* та

ПунктиПостачань встановлено співвідношення 1:∞, оскільки за одне постачання може надійти багато товарів.

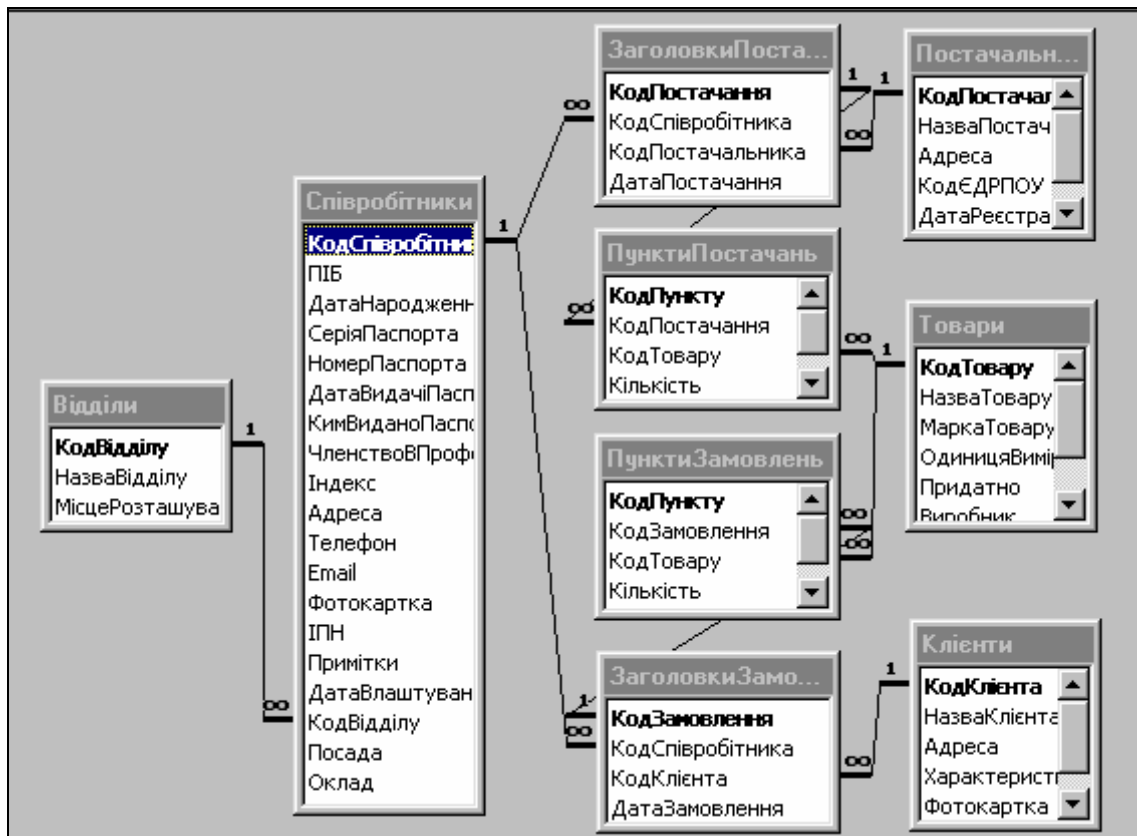


Рис. 7. Схема даних БД обліку обігу товарів гуртового складу на інфологічному рівні

Дані замовлень товарів в БД також зберігаються в двох таблицях: *ЗаголовкиЗамовлень* та *ПунктиЗамовлень*. Перша з них містить дані про сам факт замовлення, а друга зберігає відомості про окремі замовлені товари. Між таблицями *ЗаголовкиЗамовлень* та *ПунктиЗамовлень* встановлено співвідношення 1:∞, оскільки за одне замовлення клієнт може закупити багато товарів.

< Аналогічно обґрунтовуються інші структурні зв'язки між таблицями схеми даних. Далі описуються запити системи, хід їх розв'язання та атрибути таблиць, які при цьому використовуються. >

3. Проектування БД на даталогічному рівні

Для створення ІС згідно розробленого проекту БД на інфологічному рівні нами було обрано СУБД MS Access, оскільки вона дозволяє:

- розробляти нескладні додатки обробки БД в інтерактивному режимі без створення програмного коду;
- ефективно модифікувати створені об'єкти в режимі конструктора;
- копіювати, імпортувати та експортувати дані таблиць;
- перевіряти і забезпечувати цілісність реляційних відношень між таблицями та даними окремих записів таблиць;
- створювати ефективні засоби для редагування даних таблиць, здійснення пошуку, сортування та фільтрування даних, організації взаємодії між об'єктами.

Недоліки СУБД MS Access пов'язані з проблемами надійного захисту даних та продуктивної одночасної роботи з ІС багатьох користувачів. Але для ІС гуртового складу вони не мають особливого значення, оскільки з такою системою працює обмежене коло співробітників з декількох осіб.

Таблиці ІС розроблені згідно інформаційно-логічної моделі БД (див. рис. 7). Структура та типи даних полів таблиці *Співробітники* наведені в табл. 2.

Таблиця 2

Структура та типи даних полів таблиці *Співробітники*

Ім'я поля	Тип даних
КодСпівробітника	Счетчик
ПІБ	Текстовый
ДатаНародження	Дата/время
СеріяПаспорта	Текстовый
НомерПаспорта	Числовой
КимКолиВиданоПаспорт	Текстовый
ЧленствоВПрофспілці	Логический
Індекс	Числовой
Адреса	Текстовый
Телефон	Текстовый
Email	Текстовый
Фотокартка	Поле OLE
ІПН	Текстовый
Примітки	Поле MEMO
ДатаВлаштування	Дата/время
КодВідділу	Числовой
Посада	Текстовый
Оклад	Денежный

< Аналогічно описуються структури та типи даних полів інших таблиць ІС >.

Контрольні запитання

- Що таке БД? СУБД? Чому на практиці найчастіше використовують реляційні БД?
- На яких рівнях виконують проектування БД? З чим це пов'язано? Як узгоджуються між собою моделі кожного рівня проектування? Який результат проектування БД на кожному рівні?
- Які кроки необхідно виконати при проектуванні БД на зовнішньому рівні?
- Які кроки необхідно виконати при проектуванні БД на інфологічному рівні?
- Чому атрибути предметної області, як правило, неможливо розмістити в одній таблиці? На що вказують структурні зв'язки між таблицями?
- Що таке первинний і вторинний ключ? Які типи даних обирають для таких полів?
- На якому рівні проектування БД враховують особливості та обмеження обраної СУБД? Який найважливіший етап цього рівня проектування?