**6 Засоби забезпечення цілісності даних. Ключі, індекси. Посилальна цілісність**

**6.1 Ключі та індекси**

*Потенційний ключ* (*primary key*) сутності - це один або кілька атрибутів, чиї значення однозначно визначають кожен екземпляр сутності. При існуванні декількох потенційних ключів один з них позначається як *первинний ключ*, а інші - як *альтернативні* або *вторинні* ключі. В СКБД Access (і в багатьох інших теж) по первинному ключу автоматично створюється індекс (див. нижче). Індексування по вторинних ключах є справою розробника бази даних. Потенційний ключ може ще бути *унікальним* (*unique*). Від первинного він відрізняється тим, що серед його значень може бути значення „ніщо” (null). В СКБД Access первинний ключ таблиці може бути створений в режимі конструктора, для цього потрібно виділити відповіді поля (поле), після чого встановити цю властивість, використовуючи панель інструментів або контекстне меню. Унікальний ключ може бути створений при індексації вторинних ключів.

Нижче наведений приклад встановлення первинного ключа – *primary key* (рис. 6.1).

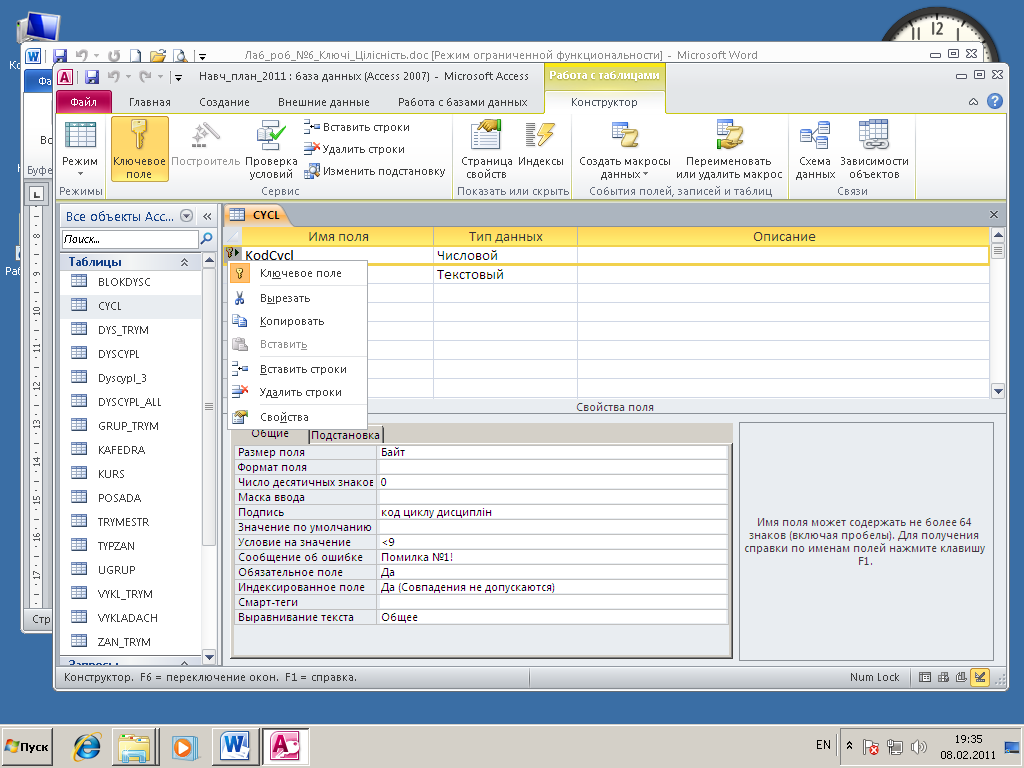


Рис. 6.1 Встановлення поля у якості первинного ключа

Первинний ключ може бути простим, який складається з одного поля, або складеним – з декілька полів. Для встановлення складеного ключа ці поля треба виділити. Приклад встановлення складеного ключа наведено на рис. 6.2.

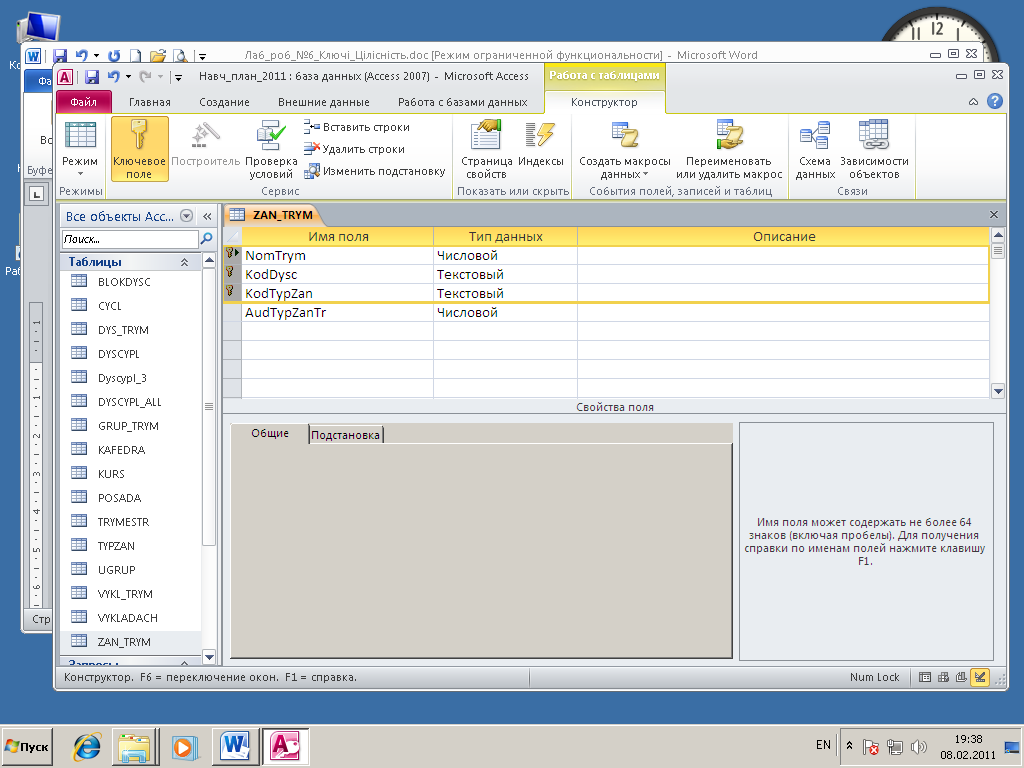
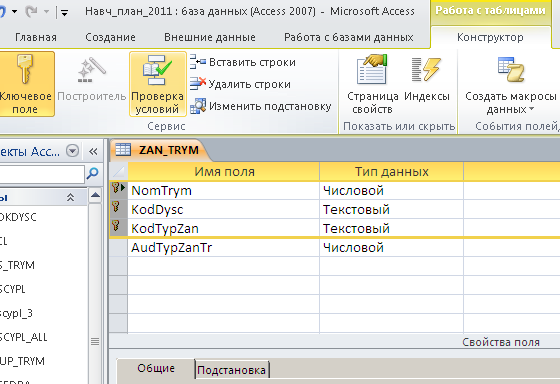


Рис. 6.2 Визначення складеного ключа

Змінити порядок полів у ключі можна через вкладку **Работа с таблицями** групу **– Индексы/Indexies**,перетягуючи рядки з полями у вікні індексів (рис.6.3).



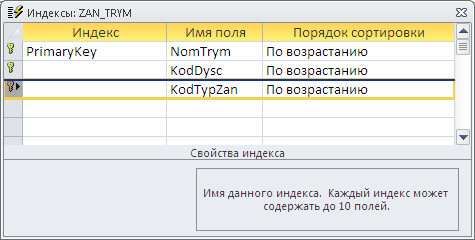


Рис.6.3 Зміна порядку полів складеного ключа

*Індексація* таблиці по полю (сукупності полів) – це засіб прискорення пошуку даних за рахунок створення на фізичному рівні бази даних додаткової таблиці (індексного файлу), у якій знаходяться тільки значення індексованих полів та покажчики на кортеж в основній таблиці з тим же значенням індексованих полів. Одна таблиця може бути проіндексована по різним полям та по різним сукупностям полів. Таким чином, індексація дозволяє прискорити пошук інформації, але вона потребує додаткової пам’яті. При наявності індексного файлу пошук спочатку здійснюється в ньому, а потім, при наявності запису із шуканим значення, іде звернення до основної таблиці за адресою, що задана покажчиком. Таким чином, поняття ключа є поняттям логічного рівня БД, а індексу – фізичного. В СКБД Access індекс поля таблиці може бути створений в режимі конструктора, для цього потрібно при визначенні властивості поля Indexed задати значення Yes, після чого встановити одну із опцій Duplicates OK (це і буде унікальний ключ) або No Duplicates. Як уже відмічалося, у більшості сучасних СКБД при визначенні поля (сукупності полів) у ролі первинного ключа це поле або сукупність полів автоматично індексуються з властивістю Indexed →Yes→ No duplicate.

**6.2 Цілісність даних в базах даних**

*Цілісність* (*integrity*) – це комплекс вимог до підтримання узгодженості та несуперечності даних при виконанні операцій над ними. Неузгодженість та суперечність між даними може виникати з цілого ряду обставин.

Вони можуть бути наслідками

* апаратного збою,
* помилок у системному програмному забезпеченні
* або логічних помилок в прикладному програмному забезпеченні.

Реляційні СКБД захищають дані від такого роду неузгодженостей шляхом використання такого механізму, як *транзакція*, який тут не розглядається*.*

*Другий тип цілісності, що зветься об’єктною* *цілісністю* (*entity* *integrity*), зв’язаний з коректним проектуванням бази даних. У попередніх роботах використовувалися такі засоби підтримання об’єктної цілісності: тип, розмір, вхідна маска, правило обмеження значень атрибута, обов’язковість значення (Data type, Field Size, Input Mask, Validation Rule, Required). В п.6.1 було розглянуто ще один засіб підтримання об’єктної цілісності – первинний ключ. Визначення первинного ключа відносять до засобів об’єктної цілісності тому, що СКБД не дозволить увести запис з дублюючим значення ключа.

*Третій тип цілісності*, називаний *посилальною цілісністю* (*referential* *integrity*), означає несуперечність між частинами інформації, що повторюється в різних таблицях. Посилальна цілісність визначається при встановлення зв’язків між таблицями. У середовищі СКБД Access зв’язки між таблицями можна задавати „динамічно” при створенні запитів (виконували в роботах №4, №5), форм і звітів, а можна одразу встановити усі взаємозв’язки (на вкладці DatabaseTools натискується кнопка  - Relationship). Другий шлях треба використовувати, коли вже остаточно визначена структура БД та усі запити, форми і звіти, що увійдуть до проекту. А якщо ви знаходитися у процесі експериментів зі структурою БД, то взаємозв’язки краще встановлювати динамічно. В лабораторних роботах спочатку будемо використовувати перший спосіб, а потім обидва способи. Посилальна цілісність встановлюється тільки при встановленні схеми взаємозв’язків (Relationship).

В СКБД Access ця властивість зв’язку може бути задана шляхом встановлення прапорця на пункті „Enforced Integrity ” (рис. 6.4).

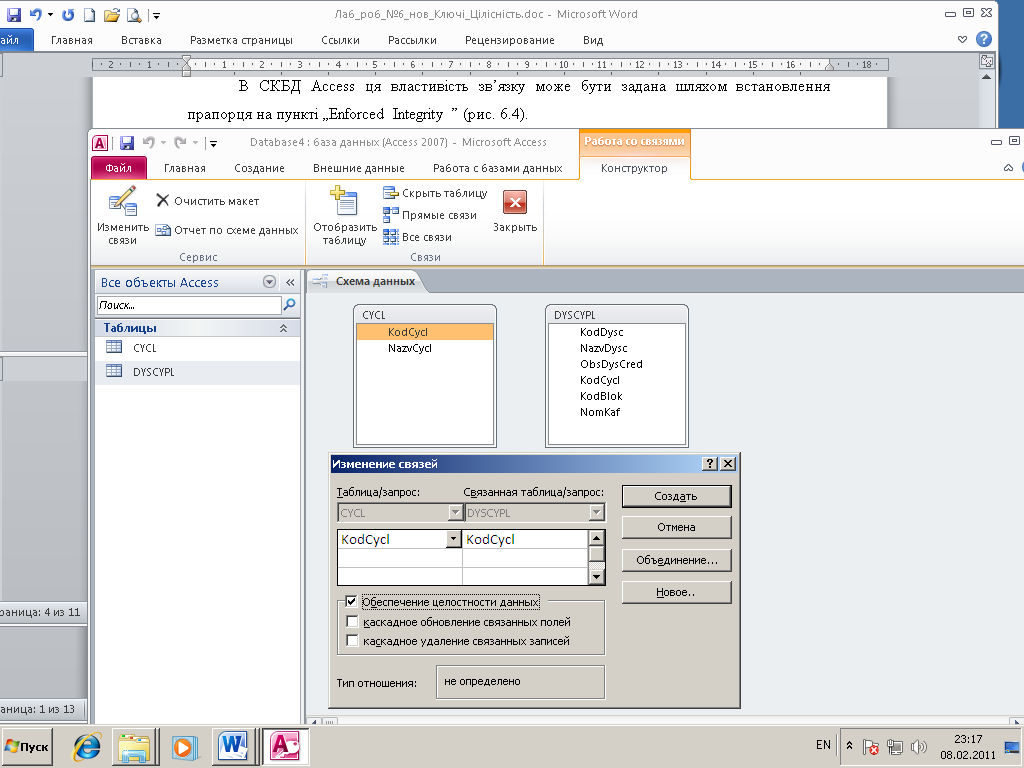
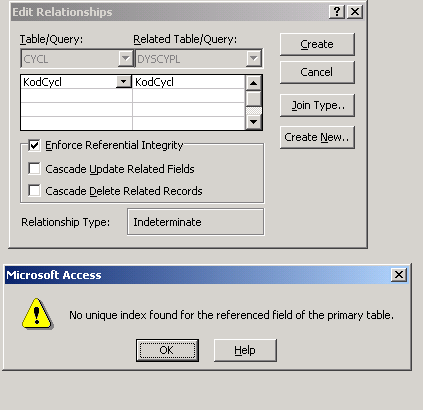


Рис. 6.4 Завдання режиму посилальної цілісності при встановленні зв’язків

між таблицями CYCL i DYSCYPL по атрибуту KodCycl

Причиною неможливості визначення режиму посилальної може бути:

А) У батьківській таблиці поле, за яким встановлюється зв’язок, може припускати значення-дублікати. Для запобігання цьому треба, щоб воно було проіндексоване з параметром унікальності або визначено як первинний ключ. В останньому випадку воно автоматично буде й унікальним індексом. Цю вимогу можна сформулювати так: поле батьківській таблиці, по якому встановлюється зв’язок, має бути проіндексованим і унікальним. При невиконані цієї вимоги СКБД *Access* після натиснення кнопки Create виводе відповідне повідомлення (рис. 6.5).



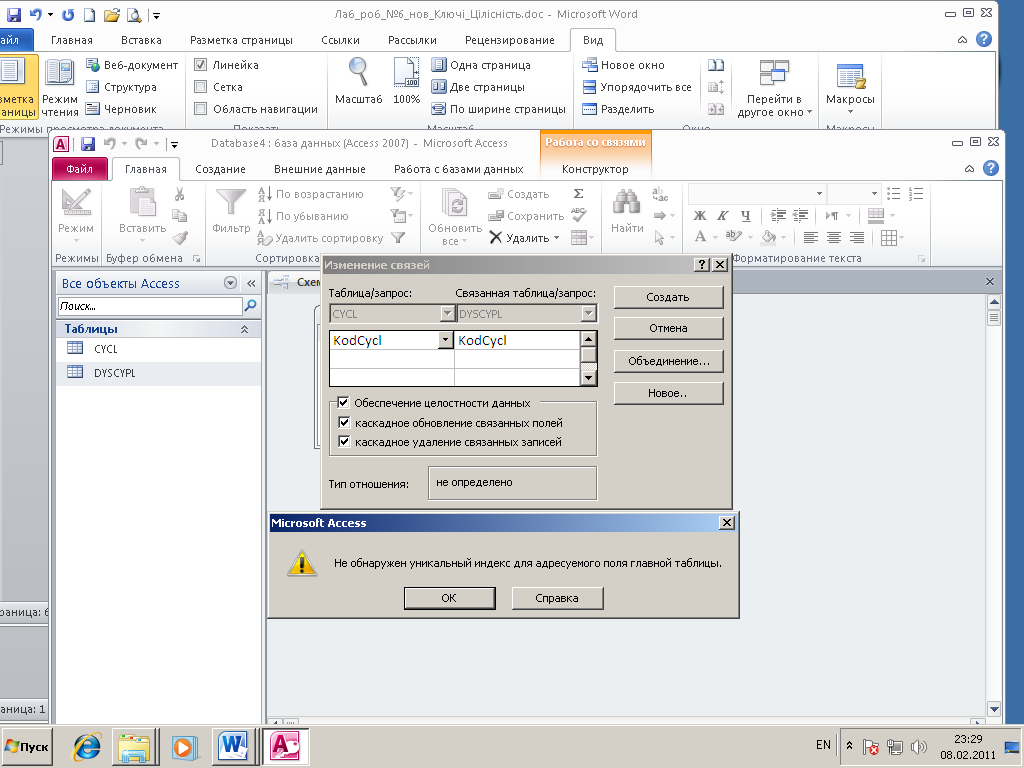
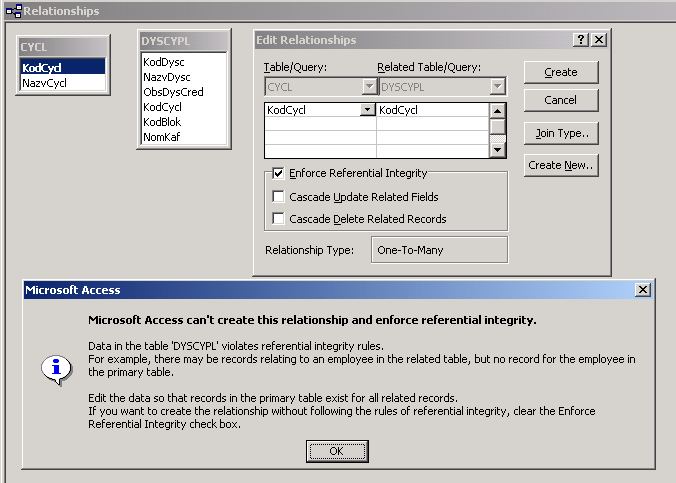


Рис. 6.5. Повідомлення про відсутність унікального індексу у батьківській таблиці

Б) Наявність у підпорядкованій (синовній або ще дочірній) таблиці кортежів-сиріт, тобто таких, що мають такі значення поля, за яким встановлюється зв’язок, що відсутні у батьківській (головній) для даного зв’язку таблиці навіть після того, як поле KodCycl таблиці CYCL визначили первинним ключем (рис 6.6).

Для виявлення наведених випадків в СКБД *Access* є відповідні допоміжні засоби – спеціальні запити, хоча їх можна встановити звичайними запитами, але це буде складніше.



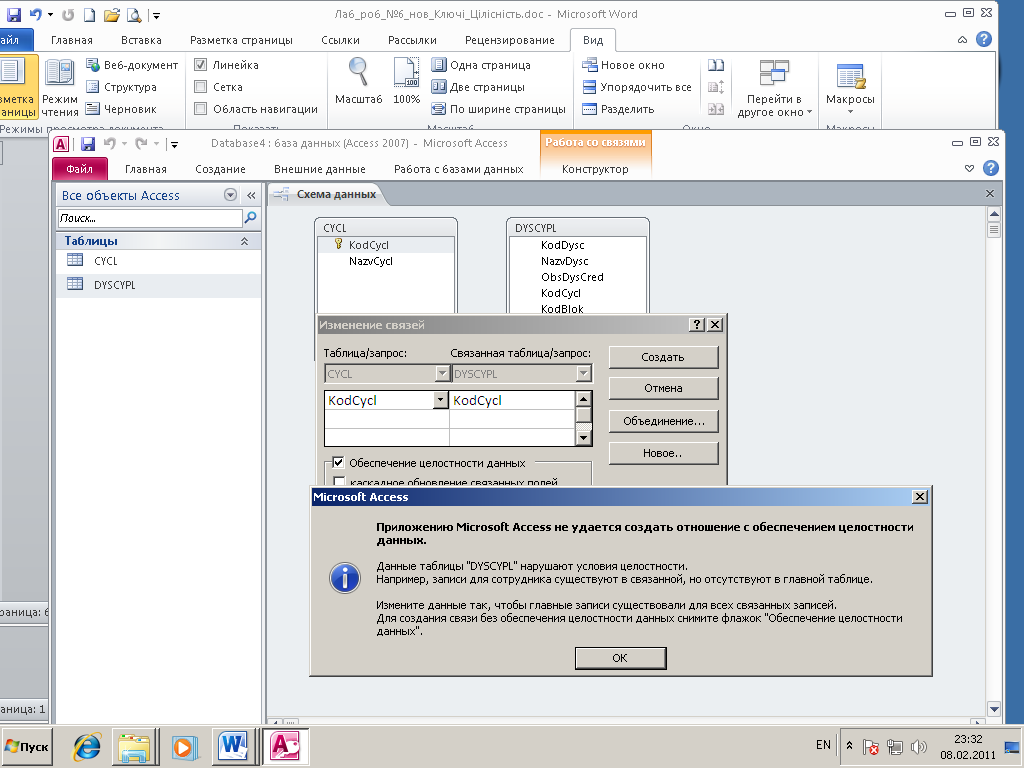


Рис.6.6 Повідомлення про відсутність у батьківській таблиці CYCL таких значень

поля KodCycl, що є у підпорядкованій таблиці DYSCYPL

У першому випадку (А) треба виконати *запит повторюваних записів* (Find Duplicates Query Wizard), який використовуються для перевірки унікальності значень полів у таблиці, якщо поле не встановлюється як Primary Key або коли при створенні індексу для нього неможливо встановити властивість „No duplicate”.

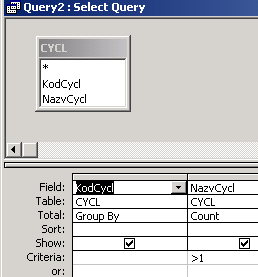
Результат такого запиту для таблиці CYCL, яка має два кортежі із значенням „7” для атрибута KodCycl, та вміст таблиці наведено на рис.6.7.

| **CYCL** | |
| --- | --- |
| **код циклу дисциплін** | **назва циклу дисциплін** |
| 1 | Цикл дисциплін гуманітарної та соціальної підготовки |
| 2 | Цикл дисциплін природничо-наукової підготовки |
| 3 | Цикл дисциплін професійної та практичної підготовки |
| 4 | Цикл дисциплін самостійного вибору ВНЗ |
| 5 | Блоки циклу дисциплін вільного вибору студентом |
| 6 | Блоки циклу дисциплін вільного вибору студентом за додаткові платні послуги |
| 7 | Дисципліни на запрошення провідних вчених |
| 7 | Дисципліни вибору наглядової ради університету |

| **Find duplicates for CYCL** | |
| --- | --- |
| **KodCycl** | **NazvCycl** |
| 7 | Дисципліни вибору наглядової ради університету |
| 7 | Дисципліни на запрошення провідних вчених |

Рис. 6.7 Результат виконання запита на дублювання рядків

При відсутності такого типу запитів дублі індексованого поля можна знайти звичайним запитом шляхом групування по цьому полю і підрахуванням кількості дублів, після цього використати фільтр “>1” (рис.6.8).

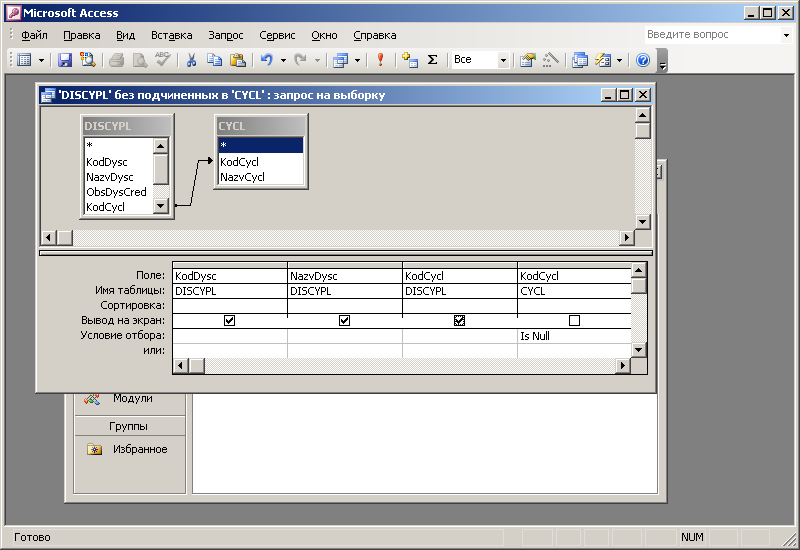


| **KodCycl** | **CountOfNazvCycl** |
| --- | --- |
| 7 | 2 |

Рис. 6.8. Запит на отримання дублікатів (зліва) та результат його виконання (справа)

Якщо нас цікавлять усі поля записів з дублями-полями, то треба зробити ще один запит над отриманим запитом, щоб отримати результат, аналогічний результату рис. 6.4. Використовуючи вкладені запити мови SQL, такий результат можна отримати в одному запиті.

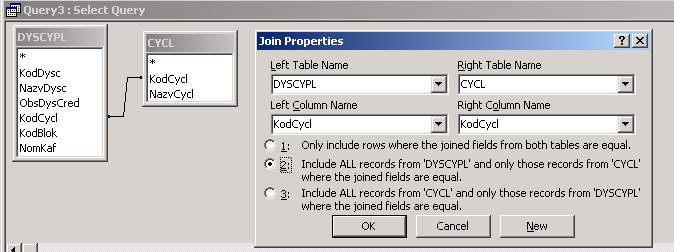
У другому випадку (Б) треба виконати з*апит* *на пошук записів-сиріт в підпорядкованій таблиці або записів, що не зв’язані з будь якими записами головної (батьківської) таблиці* (Find Unmatched Query Wizard). Приклад запиту наведено на рис. 6.9. В таблиці СYCL немає коду циклу 8, а в таблиці DISCYPL є декілька записів з таким кодом циклу (рис. 6.9).

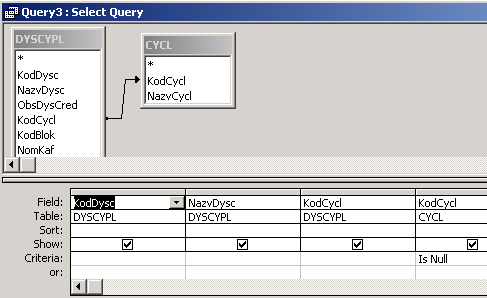


| **'DISCYPL' без подчиненных в 'CYCL'** | |
| --- | --- |
| Код дисципліни | Назва дисципліни | Код циклу дисциплін |
| 8.01 | Мови об'єктно-оріентованого програмування | 8 |
| 8.02 | Графічне та геометричне моделювання та інтерактивні системи | 8 |
| 8.03 | Графічне та геометричне моделювання та інтерактивні системи | 8 |

Рис. 6.9 Множинна різниця між таблицями

Такий запит, як відмічалося, можна створити також або мовою візуального програмування QBE (рис. 6.10), або командною мовою структурованих запитів SQL . При використанні QBE треба до бланку запита спочатку занести підпорядковану таблицю DYSCYPL, потім головну таблицю і встановити семафор властивостей зв’язків в положення “2” – зовнішнє ліве з’єднання. Після цього внести до бланку інформативні для даного запиту поля лівої і правої таблиць, при цьому поле правої таблиці, по якому встановлено зв’язок, вноситься обов’язково. Зрозуміло, що це поле прийматиме значення null у тих кортежах, де таке ж поле лівої таблиці приймає значення, які відсутні у правій (батьківській таблиці). Якщо на цьому полі поставити умову (criteria) Is Null, то й отримаємо записи-сиріти (рис. 6.11).

Рис. 6.10 Встановлення зовнішнього лівого з’єднання

Зовнішнє ліве з’єднання на бланку запиту має інший вигляд, ніж звичайне,

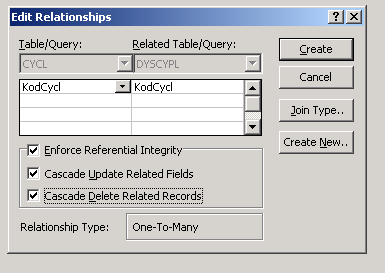
внутрішнє (рис. 6.11).

|  | | | |
| --- | --- | --- | --- |
| **KodDysc** | **NazvDysc** | **DYSCYPL.KodCycl** | **CYCL.KodCycl** |
| 8.01 | Мультиагентні системи | 8 |  |
| 8.02 | Мультимедійні системи | 8 |  |
| 8.03 | Системи обробки експериментальних даних | 8 |  |

Рис. 6.11. Структура і результат запиту на пошук неузгодженості полів між

таблицями DYSCYPL і CYCL по полю KodCycl

Після вилучення дублікатів поля KodCycl таблиці CYCL та визначення його первинним ключем, а також видалення із таблиці DYSCYPL кортежів (рядків), де KodCycl=’8’, режим посилальної цілісності встановлюється, а схема зв’язків наведених таблиць приймає вигляд, як на рис.6.12 справа. При встановленні режиму посилальної цілісності можна встановити прапорці ще на двох режимах: каскадного оновлення та каскадного видалення (рис.6.12 зліва.)



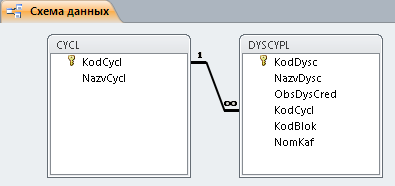


Рис. 6.12. Схема взаємозв’язків після встановлення режиму посилальної цілісності

* 1. **Лабораторна робота №6.**

**Завдання**

***Формування бази даних***

Передмова. Перед виконанням роботи мається на увазі, що в таблицях ключі ще не встановлені.

1. Відкрийте свою базу даних.
2. Створіть у своїй базі даних нові таблиці DYSCYPL \_6 шляхом імпорту відповідної Excel-таблиці.
3. В режимі конструктора встановіть потрібні типи даних та довжину. Маски, формати та інші властивості можна не задавати.
4. Визначте атрибут DYSCYPL\_6. KodDysc як первинний ключ.
5. Якщо не встановлюється властивість, знайдіть дублікати поля засобом *запит повторюваних записів* (Find Duplicates Query Wizard), видаліть дублікат(и) і повторить п.4.
6. Відкрийте схему даних – кнопка Relationship на панелі інструментів. До схеми підключіть таблицю DYSCYPL\_6 і спробуйте встановити режим посилальної цілісності між таблицями CYCL та DYSCYPL\_6 . Якщо не встановився режим, то п.7 інакше п.8
7. Створити запит на пошук записів-сиріт в підпорядкованій таблиці DYSCYPL\_6 (головна таблиця CYCL ) – запит Find Unmatched Query Wizard) по полю KodCycl . Записи-сироти видалити. Повторити п.6.
8. Додайте до режиму посилальної цілісності ще режими групового оновлення та групового видалення.
9. Перевірити встановлені ключі до таблиць, згідно з таблицями 2.4 – 2.17 (лабораторна робота №2).
10. Відкрити схему даних, додайте до схеми таблицю POSADA і створити (перевірити) зв’язки, згідно з рис.6.13.

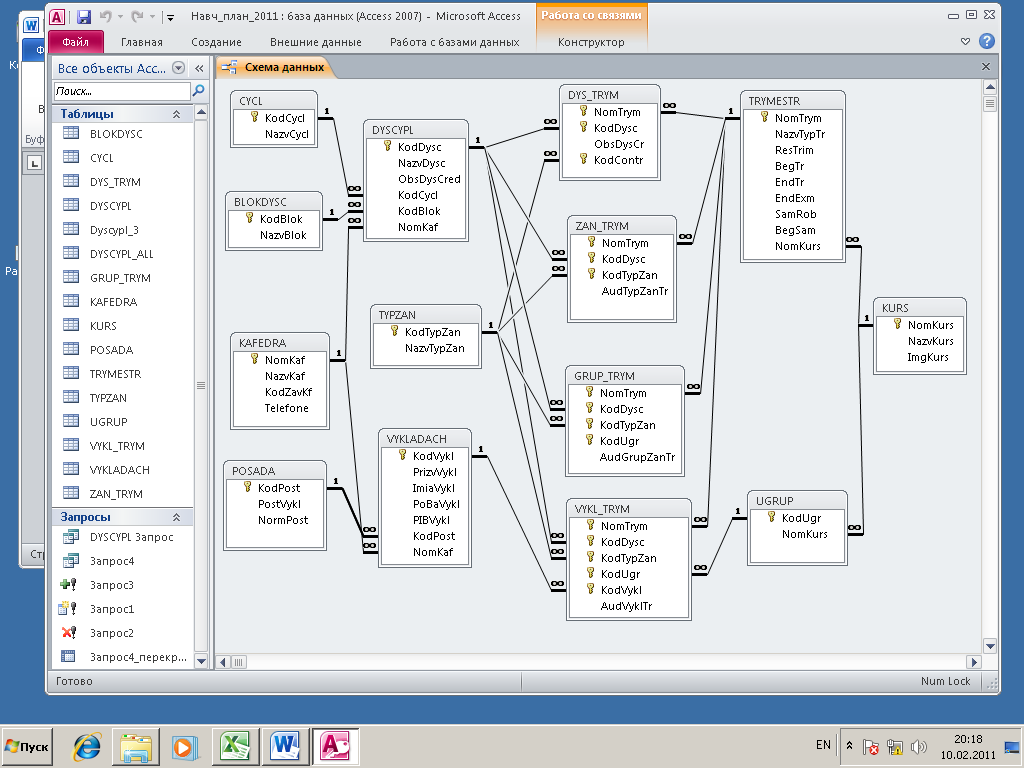


Рис.6.13. Схема бази даних з визначенням посилальної цілісності між таблицями

1. Кінець роботи.