

Кафедра управління інформаційною безпекою  
Спеціальність «Кібербезпека», 5 курс навчання  
**ЛЕКЦІЯ. РЕЛЯЦІЙНА МОДЕЛЬ ДАНИХ**

**План:**

1. Базові поняття.
2. Цілісність бази даних.
3. Реляційна алгебра.
4. Обчислення кортежів.
5. Обчислення доменів.

**Література**

1. Гайна Г.А. Основи проектування баз даних: Навчальний посібник. К.: КНУБА, 2005. – 204 с.
2. Гайна Г.А. Організація баз даних і знань. Мови баз даних: Конспект лекцій.–К.:КНУБА, 2002. – 64 с.
3. Гайна Г.А., Попович Н.Л. Організація баз даних і знань. Організація реляційних баз даних: Конспект лекцій.–К.:КНУБА, 2000. – 76 с.

**1. Базові поняття.**

Реляційна модель даних заснована на математичному понятті відношення і представленні відношень у вигляді таблиць. Запропонована на початку 70-х років американським вченим Е.Коддом. В будь-якій реляційній СУБД припускається, що користувач сприймає БД як набір таблиць. Це стосується тільки логічної структури БД, тобто відноситься до концептуального і зовнішнього представлень.

На фізичному рівні БД реалізується за допомогою різних структур зберігання. В табл. 1 наведені елементи реляційної моделі. Для однозначної ідентифікації рядків, для зв'язування таблиць між собою, для прискорення операцій над даними застосовують ключі. В табл. 2 наведені можливі види реляційних ключів. Зовнішній і відповідний йому потенційний ключі повинні бути визначені на одному домені.

Таблиця 1

Елементи реляційної моделі

Елементи реляційної моделі	Форми представлення
Відношення	Таблиця
Кортеж	Рядок таблиці
Атрибут	Заголовок стовпця таблиці
Ключ	Сукупність атрибутів, які унікально визначають

	кожен рядок таблиці, або виконують функції зв'язування таблиць, або дозволяють прискорити операції над таблицями
Домен	Множина значень атрибута
Схема відношення	Рядок заголовків стовпців таблиці

Таблиця 2

### Реляційні ключі

Назва	Пояснення
<i>Потенційний ключ (Candidate Key)</i>	Мінімальна підмножина атрибутів відношення, які єдиним чином ідентифікують кортеж даного відношення
<i>Первинний ключ (Primary Key)</i>	Потенційний ключ, який обрано для унікальної ідентифікації кортежів відношення
<i>Вторинний ключ (Secondary Key)</i>	Ключ, кожному значенню якого може відповідати більш ніж один екземпляр індексованих даних
<i>Зовнішній ключ (Foreign Key)</i>	Сукупність атрибутів відношення, значення яких є одночасно і значеннями первинного або потенційного ключа іншого відношення

Приклад. Розглянемо відношення Студент (рис. 1).

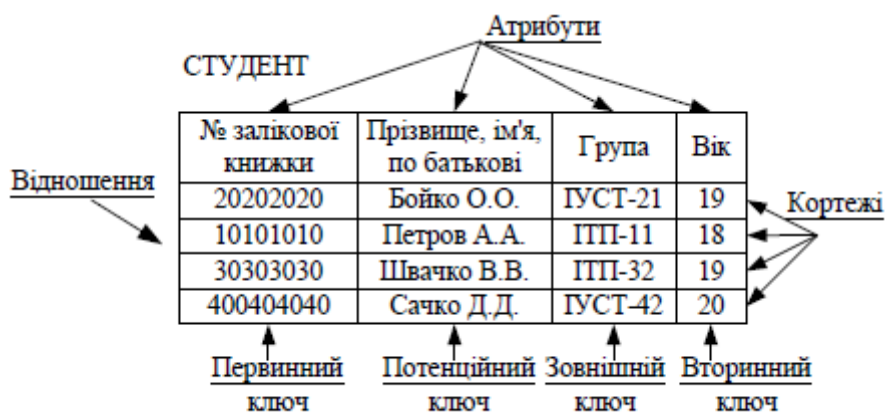


Рис. 1. Відношення *Студент*

Порядок кортежів у відношенні не визначений. В реляційних СУБД для зручності кортежі впорядковують за допомогою ключів (первинних або вторинних). В якості первинного ключа виступає атрибут № залікової книжки, який дозволяє унікально ідентифікувати кожен кортеж. Атрибут Вік обирається в якості вторинного ключа (не є обов'язковим) для виконання операцій сортування і групування студентів за віком. Атрибут Група обирається в якості

зовнішнього ключа для зв'язування з таблицею Група (на рис. 1 не представлена). Домени показують множину всіх можливих значень певного атрибута відношення. Наприклад, для атрибута Вік значення домену відносяться до типу цілих чисел.

Реляційна модель складається з таких частин:

- структурна (тут фіксується відношення як єдине ціле);
- маніпуляційна (тут використовуються два базових механізми маніпулювання реляційною БД – реляційна алгебра і реляційні обчислення);
- цілісність (тут використовується механізм, який запобігає руйнуванню даних).

Реляційній моделі даних властиві простота і природність використовуваних структур даних і операцій маніпуляції даними, повна незалежність від середовища зберігання даних, підтримка віртуальних, а не фізичних зв'язків між даними (на основі значень даних, а не покажчиків).

Реляційна БД включає в себе такі складові:

- інформаційні масиви (таблиці, індекси);
- системна інформація (структура БД, обмеження цілісності);
- прикладні програми (процедури, тригери).

Операційні можливості відношення мають дві еквівалентні форми – реляційна алгебра і реляційне обчислення. У свою чергу реляційне обчислення поділяється на реляційне обчислення зі змінними кортежами, яке називається обчислення кортежів, і на реляційне обчислення зі змінними доменами, яке називається обчислення доменів. Для виконання запитів до БД Е.Кодд запропонував відповідні принципи побудови трьох мов.

*Мови запитів реляційної алгебри* – це алгебраїчні мови, які дозволяють висловлювати запити засобами спеціалізованих операторів, що застосовуються до відношень.

*Мови реляційного обчислення* дозволяють висловлювати запити шляхом специфікації предиката, якому повинні відповідати потрібні кортежі (домени).

Реальні мови запитів (SQL, QBE і т.ін.) забезпечують не тільки функції відповідної теоретичної мови, але і реалізують деякі додаткові операції (арифметичні, друку і т.ін.).

## **2. Цілісність баз даних.**

Цілісність баз даних – властивість даних, що визначає повноту і правильність інформації, яка вміщується в БД. Підтримка цілісності даних включає такі складові:

- структурна цілісність;
- обмеження реальних значень даних;
- посилкова цілісність.

Структурна цілісність передбачає виконання таких умов:

- наявність тільки однорідних структур даних типу "реляційне відношення";
- відсутність дублікатів кортежів;
- обов'язкова наявність у кожному відношенні первинного ключа;
- обмеження доменів, яке передбачає визначення кожного атрибуту на своєму домені;
- можливість застосування невизначених значень NULL (позначає відсутність будь-якого значення атрибуту).

Невизначене NULL значення розглядається, як значення невідоме на даний момент часу. Це значення при появі додаткової інформації може бути замінено на деяке конкретне значення. Введення NULL викликало необхідність застосування замість двозначної логіки тризначної логіки. У цьому випадку передбачаються реляційні операції з невизначеними значеннями.

Обмеження реальних значень даних вимагають, щоби значення поля належали деякому діапазону значень, або задовольняли певному арифметичному співвідношенню між значеннями різних полів. Обмеження значень можуть включати також визначення певних форматів для полів, задоволення значень полів певним статистичним умовам, бізнес правилам предметної області і т.ін.

Посилкова цілісність означає, що зміни в таблицях повинні виконуватися синхронно, а зміст двох пов'язаних таблиць має відповідати таким правилам:

- кожному запису основної таблиці відповідає нуль або більше записів підлеглої таблиці;
- в підлеглій таблиці немає записів, які не мають батьківських записів в основній таблиці;
- кожний запис підлеглої таблиці має тільки один батьківський запис основної таблиці.

Умови цілісності даних визначають, які дані можуть бути записані в БД у результаті додавання або оновлення даних. При маніпулюванні даними в таблицях виконується контроль дій відповідно до табл. 3.

Таблиця 3

### Правила вилучення і оновлення

Операція	Правило	Пояснення
Вилучення (DELETE)	RESTRICT	Заборона вилученні рядка з батьківської таблиці, якщо в підлеглій таблиці цей рядок має нащадків
	CASCADE	При вилученні рядка з батьківської таблиці в підлеглій таблиці всі рядки-нащадки автоматично вилучаються
	SET NULL	При вилученні рядка з батьківської таблиці в підлеглій таблиці всім зовнішнім ключам рядків-нащадків автоматично присвоюється значення NULL
	SET DEFAULT	При вилученні рядка з батьківської таблиці в підлеглій таблиці всім зовнішнім ключам рядків-нащадків автоматично присвоюється певне значення встановлене за замовчуванням
Оновлення (UPDATE)	RESTRICT	Заборона зміни первинного ключа в рядку батьківської таблиці, якщо в підлеглій таблиці цей рядок має нащадків
	CASCADE	При зміні первинного ключа в рядку батьківської таблиці в підлеглій таблиці відповідні значення зовнішнього ключа

		також автоматично змінюються у всіх рядках-нащадках для того, щоби відповідати новому значенню первинного ключа
	SET NULL	При зміні первинного ключа в рядку батьківської таблиці в підлеглий таблиці відповідні значення зовнішнього ключа також автоматично змінюються у всіх рядках-нащадках і їм присвоюється значення NULL
	SET DEFAULT	При зміні первинного ключа в рядку батьківської таблиці в підлеглий таблиці відповідні значення зовнішнього ключа також автоматично змінюються у всіх рядках-нащадках і їм присвоюється певне значення встановлене за замовчуванням

Також можливо виконання правила NONE – не виконуються ніякі дії і правила NULL ALLOWED – дозволяються невизначені значення.

При введенні нових рядків (INSERT) необхідно дотримуватися такої послідовності введення: спочатку дані вводяться в батьківську таблицю, а потім – в підлеглу.

### 3. Реляційна алгебра.

*Алгеброю* називається множина об'єктів із заданою на ній сукупністю операцій, які замкнені відносно цієї множини.

Основною множиною в реляційній алгебрі є множина відношень. Варіант реляційної алгебри, запропонований Коддом, містить такі основні операції: об'єднання, різниця, перетин, декартовий добуток, проекція, селекція, з'єднання, ділення. На рис. 2 показані основні операції реляційної алгебри.

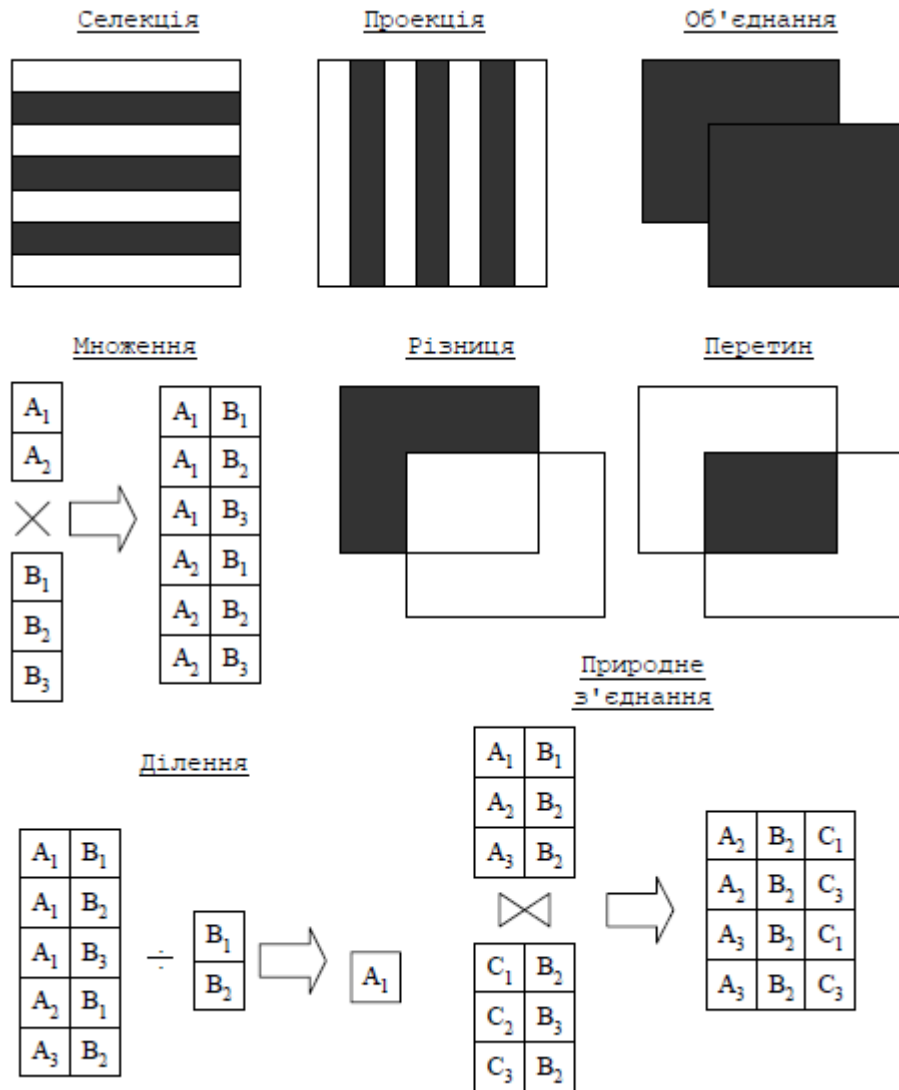


Рис. 2. Основні операції реляційної алгебри

В табл. 4 подані основні і додаткові операції реляційної алгебри.

Таблиця 4

Реляційні операції

Операція	Позначення	Зміст
Вибірка	$\sigma_{\text{предикат}}(R)$	Визначення відношення, яке вміщує тільки ті кортежі з відношення R, які задовольняють заданому предикату
Проекція	$\pi_{a_1 \dots a_n}(R)$	Визначення відношення, яке вміщує вертикальну підмножину відношення R, що утворюється шляхом отримання значень вказаних

		атрибутів і виключення з результату рядків-дублікатів
Об'єднання	$R \cup S$	Визначення відношення, яке вміщує всі кортежі, що належать $R$ або $S$ , при цьому виключаються з результату всі дублікати кортежів. Відношення $R$ і $S$ повинні бути сумісні за об'єднанням
Перетин	$R \cap S$	Визначення відношення, яке вміщує всі кортежі, що належать $R$ і $S$ . Відношення $R$ і $S$ повинні бути сумісні за об'єднанням
Різниця	$R - S$	Визначення відношення, яке вміщує всі кортежі, що належать $R$ і відсутні в $S$ . Відношення $R$ і $S$ повинні бути сумісні за об'єднанням
Декартовий добуток	$R * S$	Визначення відношення, яке є результатом конкатенації кожного кортежа з відношення $R$ з кожним кортежем з відношення $S$
Тета-з'єднання	$R \bowtie_F S$	Визначення відношення, яке вміщує кортежі з декартового добутку відношень $R$ і $S$ , що задовольняє предикату $F$
З'єднання еквівалентності	по $R \bowtie_F S$	Визначення відношення, яке вміщує кортежі з декартового добутку відношень $R$ і $S$ , що задовольняє предикату $F$ (предикат виконує порівняння тільки на рівність)
Природне з'єднання	$R \bowtie S$	Визначення відношення, яке отримано з'єднанням по еквівалентності двох



		відношень $R$ і $S$ , що виконано по всім спільним атрибутам $x$ , з результатів якого вилучається по одному екземпляру кожного спільного атрибута
Ліве зовнішнє з'єднання	$R \supset \supset S$	Визначення відношення, для якого кортежі відношення $R$ , які не мають співпадаючих значень в спільних стовпцях відношення $S$ , також включаються в результуюче відношення
Напівз'єднання	$R \triangleleft_F S$	Визначення відношення, яке вміщує ті кортежі відношення $R$ , які входять в з'єднання відношень $R$ і $S$
Ділення	$R \div S$	Визначення відношення, яке вміщує ті кортежі відношення $R$ , які визначені на атрибуті $C$ , що відповідає комбінації всіх кортежів відношення $S$ , де $C$ – множина атрибутів, які є у відношенні $R$ , але відсутні у відношенні $S$

*Приклад. Задані два відношення Студент і Дисципліна.*

#### Студент

Прізвище	Курс	Група	Спеціальність
Бойко	3	1	ІТП
Левченко	4	2	ІУСТ

#### Дисципліна

Назва	Курс	Спеціальність	Викладач	Семестр
Бази даних	3	ІТП	Петренко	5
Системний аналіз	4	ІУСТ	Гавриш	7

1. Визначити всіх студентів спеціальності ІУСТ.

$\Pi_{\text{прізвище}}(\sigma_{\text{спеціальність}=\text{"ІУСТ"}}(\text{Студент}))$

2. Визначити всіх студентів, для яких у 7 семестрі викладач Гавриш проводить заняття.

$\Pi_{\text{прізвище}}(\sigma_{\text{семестр}=7 \wedge \text{викладач}=\text{"Гавриш"}}(\text{Студент} \bowtie \text{Дисципліна}))$

$\Pi_{\text{прізвище}}(\text{Студент} \bowtie (\sigma_{\text{семестр}=7 \wedge \text{викладач}=\text{"Гавриш"}} \text{Дисципліна}))$

#### 4. Обчислення кортежів.

Будь-який вираз обчислення кортежів може бути представлений у такому вигляді:

$$\{ t / f(t) \}$$

де  $t$  – єдина вільна змінна – кортеж, яка позначає кортеж фіксованої довжини;  $f(t)$  – деякий предикат над змінною  $t$ .

Формули в реляційному обчисленні кортежів будують з атомів і сукупності операторів (арифметичних і логічних). Вираз в реляційному обчисленні кортежів будують над множиною відношень. Типи можливих атомів подані в табл. 5.

Для побудови формули (запиту) використовуються логічні зв'язки ( $\wedge, \vee, \neg$ ), а також квантори загальності  $\forall x$  і квантори існування  $\exists x$ . Квантори зв'язують певні змінні. Зв'язана змінна відповідає локальній змінній у мові програмування, а вільна змінна (змінна, яка не зв'язана кванторами) відповідає глобальній змінній.

Таблиця 5

#### Правила побудови атомів формул

Номер типу атому	Правила побудови атомів
1	$R(t)$ , де $R$ – ім'я відношення; цей атом означає, що $t$ є кортеж у

	відношенні $R$
2	$(s[i]\theta u[j])$ , де $s$ і $u$ – змінні кортежі; $\theta$ – арифметичний оператор відношення; $i$ і $j$ – номери або імена компонентів (стовпчиків) у відповідних кортежах; $s[i]$ – позначення $i$ -го компонента в кортежі-змінній $s$ ; $u[j]$ – позначення $j$ -го компонента в кортежі-змінній $u$
3	$(s[i]\theta a)$ або $(a\theta s[i])$ , де $a$ – константа

Формули будуються за певними правилами. Правила побудови формул наведені в табл. 6.

Таблиця 3.6

### Правила побудови формул

Номер правила	Правила побудови формул
1	Кожен атом є формула
2	Якщо $f_1$ і $f_2$ – формули, то вирази: $\neg f_1 \wedge f_2$ ; $\neg f_1 \vee f_2$ ; $\neg \neg f_1$ ; також є формулами
3	Якщо $f$ – формула, то вирази: $\forall t f(t)$ ; $\exists t f(t)$ ; також є формулами
4	Якщо $f$ – формула, то $(f)$ – також є формулою
5	Ніщо інше не є формулою

Приклад.

1. Вираз  $\{t/R_1 \wedge R_2\}$  означає, що в якості формули виступає запис  $R_1 \wedge R_2$  і що тут визначається множина кортежів, яка одночасно належить відношенням  $R_1$  і  $R_2$ . Цей вираз є еквівалентним до виразу реляційної алгебри  $R_1 \cap R_2$ .

2. Визначити всіх студентів спеціальності ІУСТ.

$\{t(\text{прізвище})/\exists t \text{Студент}(t) \wedge t(\text{спеціальність}) = \text{"ІУСТ"}\}$

3. Визначити всіх студентів, для яких у 7 семестрі викладач Гавриш проводить заняття.

$$\{ t(\text{прізвище}) / \exists t \text{ Студент}(t) \wedge \exists s \text{ Дисципліна}(s) \wedge \\ s(\text{семестр})=7 \wedge s(\text{викладач})=\text{"Гавриш"} \wedge \\ t(\text{курс, спеціальність}) = \\ s(\text{курс, спеціальність}) \}$$

### 5. Обчислення доменів.

Будь-який вираз обчислення доменів може бути представлений у такому вигляді:

$$\{ x_1, x_2, \dots, x_n / f(x_1, x_2, \dots, x_n) \}$$

де  $f$  – формула;  $x_1, x_2, \dots, x_n$  – вільні змінні

В обчисленні доменів не існує змінних кортежів. Замість них вводяться змінні на доменах. У всіх інших випадках реляційне обчислення зі змінними на доменах будується так само, як і реляційне обчислення зі змінними на кортежах.

Формули в реляційному обчисленні доменів будують з атомів і сукупності операторів (арифметичних і логічних). Вираз в реляційному обчисленні доменів будують над множиною відношень. Типи можливих атомів подані в табл. 7.

Для побудови формули (запиту) використовуються логічні зв'язки ( $\wedge$ ,  $\vee$ ,  $\neg$ ), а також квантори загальності  $\forall x$  і квантори існування  $\exists x$ .

Таблиця 7

#### Правила побудови атомів

Номер типу атому	Правила побудови атомів
1	$R(x_1, x_2, \dots, x_n)$ , де $R$ – $n$ -арне відношення; $x_i$ – константа або змінна на деякому домені. Атом $R(x_1, x_2, \dots, x_n)$ вказує на те, що значення тих $x_i$ , які є змінними, повинні бути вибрані так, щоби $(x_1, x_2, \dots, x_n)$ було кортежем відношення $R$
2	$(x\theta y)$ , де $x$ і $y$ – константи або змінні на деякому домені; $\theta$ –

	<p>арифметичний оператор відношення;</p> <p>Атом <math>(x\theta y)</math> вказує на те, що <math>x</math> і <math>y</math> являють собою значення, при яких істинно <math>(x\theta y)</math></p>
--	--

Приклад.

1. Визначити всіх студентів спеціальності ІУСТ.

$\{ \text{прізвище} / \exists x \exists y \text{ Студент}(\text{прізвище}, x, y, \text{"ІУСТ"}) \}$

2. Визначити всіх студентів, для яких у 7 семестрі викладач Гавриш проводить заняття.

$\{ \text{прізвище} / \exists x \exists y \exists z \text{ Студент}(\text{прізвище}, x, y, z) \wedge$

$\exists s \text{ Дисципліна}(s, x, z, \text{"Гавриш"}, 7) \}$

### Контрольні запитання

1. Які головні переваги реляційної моделі?
2. Дати визначення термінів: відношення, схема відношення, кортеж, ключ.
3. Які види ключів існують і навіщо вони потрібні?
4. Що таке домени і навіщо вони потрібні?
5. Що таке цілісність БД і як вона підтримується?
6. Що таке логічна і фізична цілісність БД?
7. Що таке посилкова цілісність і як вона підтримується?
8. Перелічити правила вилучення і оновлення даних у зв'язаних відношеннях.
9. Навести приклади обмежень значень і структурних обмежень.
10. Назвати основні операції реляційної алгебри.
11. Назвати додаткові операції реляційної алгебри, навести приклади.
12. Охарактеризувати варіанти реляційного обчислення.
13. Що таке обчислення кортежів?
14. Що таке обчислення доменів?
15. Перелічити правила побудови атомів в обчисленні кортежів і в обчисленні доменів.

16. Перелічити правила побудови формул в обчисленні кортежів і в обчисленні доменів.

17. Порівняти можливості реляційної алгебри і реляційного обчислення.

**Розробив: к.т.н., доц. Полотай О.І.**