Волинський національний університет імені Лесі Українки

Факультет інформаційних технологій і математики

Кафедра комп’ютерних наук та кібербезпеки

Л. В. Булатецька, В. В. Булатецький

РЕЛЯЦІНА АЛГЕБРА. РЕЛЯЦІЙНЕ ЧИСЛЕННЯ

Методичні вказівки для підготовки до контрольної роботи з нормативних навчальних дисциплін “Бази даних та розподілені інформаційно-аналітичні системи” та “Організація баз даних та знань”

Луцьк 2020

УДК 004.655
Б 90

Рекомендовано до друку науково-методичною радою
Волинського національного університету імені Лесі Українки
(протокол № 4 від 16 грудня 2020 р.)

Рецензенти:
Собчук О. М. – кандидат пед. наук, доцент кафедри загальної математики та методики навчання інформатики Волинського національного університету імені Лесі Українки;
Ліщина Н. М. – кандидат технічних наук, доцент, завідувач кафедри інженерії програмного забезпечення Луцького національного технічного університету

Б 90 Реляційна алгебра. Реляційне числення [Електронний ресурс] : методичні вказівки для підготовки до контрольної роботи з нормативних навчальних дисциплін "Бази даних та розподілені інформаційно-аналітичні системи", "Організація баз даних та знань"/ В. В. Булатецький, Л. В. Булатецька; ВНУ ім. Лесі Українки. – Електронні текстові данні (1 файл: 992 КБ). – Луцьк : ВНУ ім. Лесі Українки, 2020. – 36 с.

Викладено теоретичні відомості, які відносяться до теми "Реляційна алгебра. Реляційне числення", що входить до лекційного курсу нормативних навчальних дисциплін "Бази даних та розподілені інформаційно-аналітичні системи" та "Організація баз даних та знань". Наведено приклади розв'язків типових задач, запропоновано вправи і задачі для самостійного розв'язку та для самостійної підготовки до контрольної роботи по темі "Реляційна алгебра. Реляційне числення". Рекомендовано студентам 2, 3 курсів, спеціальності 122 Комп'ютерні науки, 014 Середня освіта, 113 Прикладна математика та 125 Кібербезпека, освітньо-професійних програм Комп'ютерні науки та інформаційні технології, Інформатика, Прикладна математика та Інформаційна безпека відповідно.

УДК 004.655

ВСТУП

“Бази даних та розподілені інформаційно-аналітичні системи” є нормативною навчальною дисципліною підготовки бакалавра напрямів 122 Комп’ютерні науки та інформаційні технології, 014 Середня освіта, 113 Прикладна математика. “Організація баз даних та знань” є нормативною навчальною дисципліною підготовки бакалавра напрямів 125 Кібербезпека. Метою викладання теорії баз даних у вищих навчальних закладах є сформувати у слухачів знання, вміння та навички, необхідні для ефективного використання засобів сучасних інформаційних систем у своїй майбутній професійній діяльності, формування у слухачів знань, вмінь та навичок з проектування, розробки баз даних, використання сучасних мов запитів до баз даних, методів оптимізації, які застосовуються в процесі експлуатації бази даних.

При вивченні баз даних в закладах вищої освіти, велика увага приділяється теорії реляційних баз даних, оскільки реляційна модель даних стала першою працездатною моделлю даних і має ефективний інструментарій – операції реляційної алгебри. Відношення в теорії реляційних баз даних є множинами, елементами яких є кортежі, тому до них застосовні всі теоретико-множинні операції. Ці операції виражаються певним чином через звичайні операції над множинами, такі як об’єднання, перетин, різниця та декартовий добуток. Реляційна алгебра включає традиційні операції над множинами, модифіковані з урахуванням того, що їх операндами є відношення та спеціальні реляційні операції. Реляційна алгебра описує порядок операцій у запиті, який вказує, як отримати результат запиту. На відміну від реляційної алгебри, реляційне числення є декларативною мовою вищого рівня. Реляційне числення визначає, який результат повинен бути отриманий. Реляційне числення також не визначає послідовність операцій, в яких буде оцінюватися запит. Вивчення реляційного числення виправдана тим, що багато результатів у формулюванні запитів до бази даних прийшла з логіки. Реляційна алгебра та реляційне числення є формальними мовами запитів для реляційної моделі.

Для кращого розуміння студентами теоретичного матеріалу, велику увагу при вивченні баз даних потрібно приділити розв’язуванню задач на застосування реляційної алгебри та реляційного числення, а також розгляду прикладів оптимізації запитів, використовуючи властивості операцій реляційної алгебри. В даній методичній розробці велику увагу також зосереджено на зв’язку між реляційною алгеброю і реляційним численням. Для контролю якості знань по темі «Реляційна алгебра та реляційне числення» в навчальній програмі курсів запланована контрольна робота, метою якої є закріплення знань з даної теми, що передбачає оволодіння студентами методами наукового аналізу, самостійного вивчення теоретичного матеріалу та набуття навичок написання запитів до бази даних. Завдання по контрольній роботі повинні допомогти студенту зрозуміти, чому мови запитів побудовані певним чином і усвідомлено писати запити до бази даних. В методичній розробці подано приклади розв’язування задач та перелік завдань для самостійного опрацювання при підготовці до контрольної роботи.

1. РЕЛЯЦІЙНА АЛГЕБРА

До складу *реляційної моделі даних*, крім структури даних, мають входити операції маніпулювання даними. З усіх таких операцій складається *мова запитів*. Найбільш відомими мовами запитів у реляційній моделі є реляційна алгебра та реляційне числення.

У класичному розумінні алгебра визначається як пара, що складається з основної множини і множини операцій (сигнатури), при цьому аргументи й результат кожної операції належать основній множині.

Реляційна алгебра є алгеброю в строгому класичному розумінні її визначення. Елементами основної множини є реляційні відношення. У зв'язку з цим операції алгебри можуть вкладатися одна в одну, тобто аргументом певної операції може бути результат виконання іншої операції. Це дає можливість записувати запити довільного рівня складності у вигляді виразів, що містять вкладені одна в одну операції.

Сигнатура реляційної алгебри Кодда складається з восьми операцій. Перш ніж детально розглянути ці операції, введемо поняття сумісності реляційних відношень. Це поняття є необхідним, оскільки деякі операції (а саме: теоретико-множинні операції об'єднання, перетину та різниці) визначені лише для сумісних реляційних відношень.

Реляційні відношення $R_1(A_1,\ldots A_n)$ і $R_2(B_1,\ldots B_k)$ називаються *сумісними*, якщо:

1) у них однакова кількість атрибутів, тобто $k=n$;

2) можна встановити взаємно однозначну відповідність між доменами атрибутів першої та другої реляцій, тобто існує таке бієктивне відображення

$S:\{1,\ldots,k\}\to\{1,\ldots,k\}$

Що $N(A_i)=N(B_{S(i)}),i=1,\ldots k$, тобто домени зіставлених атрибутів однакові.

Для зручності вважатимемо, що зіставлені атрибути сумісних відношень повинні мати однакові імена.

Дамо також означення кількох властивостей бінарних операцій:

- операція ϕ є комутативною, якщо $A\ \phi\ B=B\ \phi\ A$;
- операція ϕ є асоціативною, якщо $(A\ \phi\ B)\ \phi\ C=A\ \phi\ (B\ \phi\ C)$;
- операція ϕ є дистрибутивною з операцією θ, якщо $A\ \phi\ (B\ \theta\ C)=(A\ \phi\ B)\ \theta\ (A\ \phi\ C)$.

Даючи означення бінарним операціям реляційної алгебри, ми будемо вказувати, які з цих властивостей вони мають.

Оскільки різні відношення можуть містити атрибути з однаковими іменами, то під час виконання бінарних операцій у кінцевому відношенні можуть повторюватися імена атрибутів. Для забезпечення унікальності імен атрибутів вони уточняються іменами відповідних відношень згідно з таким синтаксисом: < ім'я відношення >.< ім'я атрибута >.

Під час розгляду операцій реляційної алгебри атрибути позначатимемо великими літерами з початку латинського алфавіту: *A, B, …,* а множини атрибутів – великими літерами з середини латинського алфавіту: *L, M, …*.

1.1. Операції над множинами.

Об'єднання. Нехай L – певна множина атрибутів. *Об'єднанням* сумісних реляційних відношень R_1 і R_2 зі схемами $R_1(L)$ і $R_2(L)$ (позначається як $R_1 \cup R_2$) називається таке реляційне відношення R зі схемою $R(L)$, що містить кортежі обох поєднуваних відношень, але без повторень (рис. 1):

$$R(L)=R_1(L) \cup R_2(L)=\{r \mid r \in R_1 \vee r \in R_2\}$$

Операція комутативна, асоціативна й дистрибутивна щодо перетину.

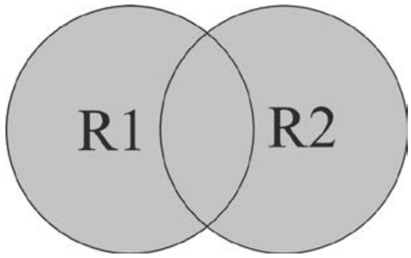


Рис. 1. Операція об'єднання ($R_1 \cup R_2$)

Приклад:

R_1

A	B
a_1	b_1
a_1	b_2
a_2	b_3

R_2

A	B
a_1	b_1
a_2	b_1

$R = R_1 \cup R_2$

A	B
a_1	b_1
a_1	b_2
a_2	b_1
a_2	b_3

Перетин. Припустимо, що L – певна множина атрибутів. *Перетином* сумісних реляційних відношень R_1 і R_2 зі схемами $R_1(L)$ і $R_2(L)$ (позначається як $R_1 \cap R_2$) називається таке реляційне відношення R зі схемою $R(L)$, яке містить кортежі, що входять до складу обох операндів (рис. 2):

$$R(L)= R_1(L) \cap R_2(L)=\{r \mid r \in R_1 \,\&\, r \in R_2\}$$

Операція комутативна, асоціативна й дистрибутивна щодо об'єднання.

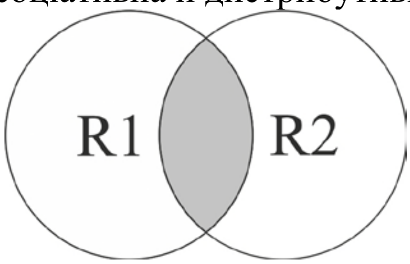


Рис. 2. Операція перетину ($R_1 \cap R_2$)

Приклад:

R_1

A	B
a_1	b_1
a_1	b_2
a_2	b_3

R_2

A	B
a_1	b_1
a_2	b_1

$R_1 \cap R_2$

A	B
a_1	b_1

Різниця. Нехай L – певна множина атрибутів. *Різницею* сумісних реляційних відношень R_1 і R_2 зі схемами $R_1(L)$ і $R_2(L)$ (позначається як R_1-R_2) називається реляційне відношення R зі схемою $R(L)$, що містить ті кортежі з першого операнда R_1, яких немає у другому операнді R_2 (рис. 3):

$R(L)= R_1(L)- R_2(L)=\{r \mid r\in R_1 \ \& \ r\notin R_2\}$

Операція не комутативна, не асоціативна й не дистрибутивна з іншими операціями.

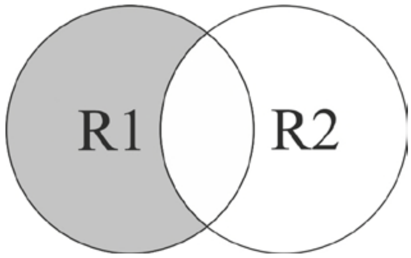


Рис. 3. Операція різниці (R_1-R_2)

Приклад:

R_1

A	B
a_1	b_1
a_1	b_2
a_2	b_3

R_2

A	B
a_1	b_1
a_2	b_1

R_1-R_2

A	B
a_1	b_2
a_2	b_3

Зауважимо, що $R\cap S=R-(R-S)$.

Зазначимо деякі особливості теоретико-множинних операцій:

- у реляційній алгебрі, на відміну від алгебри множин, не використовується операція доповнення, оскільки певні домени можуть бути нескінченними або містити дуже багато значень і в результаті операції доповнення можна отримати або нескінченне відношення, або відношення з дуже великою кількістю кортежів;
- вимога сумісності операндів зумовлена тим, що без цього обмеження результатом теоретико-множинних операцій могли б бути різноструктурні кортежі, а не реляційні відношення.

1.2. Спеціальні реляційні оператори

Розглянемо операції, які визначені лише в реляційній алгебрі.

Проекція. *Проекцією* реляційного відношення R зі схемою $R(A_1,\ldots, A_k)$ за атрибутами $A_{i1},\ldots, A_{in}$, де $\{A_{i1},\ldots, A_{in}\}\subset\{A_1,\ldots, A_k\}$, що позначається $R[A_{i1},\ldots, A_{in}]$, називається таке відношення S зі схемою $S(A_{i1},\ldots, A_{in})$, кортежі якого отримані з кортежів відношення R шляхом видалення значень, що не належать атрибутам, за якими виконується проекція. При цьому в кінцевому відношенні повторні екземпляри кортежів видаляються. Операція проекції є фільтром по атрибутах, тобто деякий вертикальний фільтр.

Якщо r є кортежем відношення R, то записом $r[L]$, де L– підмножина атрибутів відношення R, позначимо множину тих елементів кортежу r, що відповідають значенням атрибутів з L. Тоді наведене вище визначення проекції може бути записане у такий спосіб:

$$S= R[A_{i1},\dots, A_{in}]=\{ r[A_{i1},\dots, A_{in}] \mid r\in R\}.$$

Операція проекції також записується як $\pi_{Ai1,\dots, Ain}(R)$.

З теоретичної точки зору операція проекції не є „чистою”, оскільки список атрибутів не належить основній множині (тобто не є реляційним відношенням), а тому не може бути операндом. За „чистого” теоретичного підходу операцію проекції слід вважати бінарною, а на практиці вона розглядається як унарна з параметрами.

Такі ж самі проблеми мають місце в усіх операціях, де операндами є списки атрибутів.

Приклад:

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_1
a_2	b_3	c_1
a_2	b_4	c_2

$R[A,C]$

A	C
a_1	c_1
a_2	c_1
a_2	c_2

Декартовий добуток. Декартовим добутком реляційних відношень R і S зі схемами $R(A_1, A_2,\dots A_n)$ та $S(B_1, B_2,\dots B_m)$ відповідно, що позначається $R\times S$, називається відношення Q зі схемою $Q(A_1, A_2,\dots A_n, B_1, B_2,\dots B_m)$, яке містить усі можливі з’єднання кортежів відношення R з кортежами відношення S:

$$Q = R\times S = \{(r, s) \mid r\in R \ \&\ s\in S\}.$$

Операція комутативна й асоціативна.

Приклад:

R_1

A	B
a_1	b_1
a_1	b_2
a_2	b_3

R_2

C	D
c_1	d_1
c_2	d_1

$R=R_1 \times R_2$

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_1
a_1	b_2	c_1	d_1
a_1	b_2	c_2	d_1
a_2	b_3	c_1	d_1
a_2	b_3	c_2	d_1

Обмеження (селекція). Спочатку дамо означення θ-порівнянності атрибутів. Нехай θ є одним з операторів порівняння: $=, \neq, \geq, >, \leq, <$ (набір операторів можна розширити). Атрибути A і B одного й того самого чи різних відношень називаються *θ-порівнянними,* якщо для будь-яких значень $a \in A$ і $b \in B$ результат операції $a\theta b$ є визначеним (істинним або хибним). Інакше кажучи, ця операція визначена на відповідних атрибутах. Набори атрибутів $L=(A_1,\ldots, A_k)$ та $M=(B_1,\ldots, B_n)$ називаються θ-порівнянними, якщо $k=n$ і A_i θ-порівнянне з B_i $(i=1,2,\ldots k)$. Тоді вираз $L\ \theta M$ розуміють так:

$$L\ \theta M\ =(A_1\ \theta B_1)\&\ldots\&\ (A_k\ \theta B_k).$$

Тепер дамо означення операції обмеження.

Нехай L і M – набори θ-порівнянних атрибутів схеми відношення R. Тоді *обмеженням* реляційного відношення R за умовою $L\ \theta M$, що позначається $R[L\ \theta M]$, називається реляційне відношення, кортежі якого відповідають умові $L\ \theta\ M$:

$$S= R[L\ \theta M]=\{r \mid r \in R\ \&\ r[L]\ \theta r[M]\}$$

Множина M може складатися як з атрибутів, так і з констант.

Операція обмеження також записується як $\sigma_{L\ \theta M}(R)$.

Приклад:

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_1
a_2	b_3	c_1
a_2	b_4	c_2

$R[A=a_2]$

A	B	C
a_2	b_3	c_1
a_2	b_4	c_2

Крім того, умова порівняння для θ-вибірки може містити довільне число простих порівнянь, з'єднаних логічними зв’язками (AND(⋀, &), OR(⋁)). При необхідності можуть використовуватися круглі дужки.

З’єднання. Припустимо, що відношення R має схему $R(L, M)$, а відношення S– схему $S(N, P)$. Нехай множини атрибутів M і N – θ-порівнянні. Тоді *з’єднанням*, або *θ-з’єднанням*, відношень R і S за умовою $M\ \theta N$, що позначається як $R[M\ \theta\ N]S$, називається відношення Q зі схемою $Q(L,M,N,P)$, кортежі якого можна отримати з’єднанням тих кортежів відношень R і S, на яких виконується умова $M\ \theta N$:

$$Q= R[M\ \theta N]S=\{(r, s) \mid r \in R\ \&\ s \in S\ \&\ r[M]\ \theta s[N]\}.$$

Під час з’єднання атрибути, за якими виконується така операція, повторюються в кінцевому реляційному відношенні. Операція комутативна й асоціативна.

Іноді операція з’єднання позначається як $R \bowtie_F S$, де F– умова з’єднання.

З’єднання за умовою рівності називається *еквіз’єднанням.* З’єднання за умовою рівності, коли один з порівнюваних атрибутів (чи група порівнюваних атрибутів) видаляється з кінцевого відношення, називається *природним з’єднанням*; на його позначення використовується символ „*”. Наприклад, якщо задані відношення $R(A,B,C,D)$ і $S(C,D,E)$, то в результаті виконання операції $Q=R*S$ отримаємо реляційне відношення $Q(A,B,C,D,E)$.

Серед операцій θ-з'єднання виділяють операцію *напівз'єднання,* за якої з результату видаляються всі атрибути одного з відношень, що з'єднуються. Вона записується як $R[M\theta N\rangle S$ і формально визначається так:

$$Q= R[M\theta N\rangle S =\{r \mid r\in R \ \&\ s\in S \ \&\ r[M]\ \theta s[N]\}.$$

Операція напівз'єднання не розширює можливостей реляційної алгебри, оскільки вона виражається через з'єднання і проекцію в такий спосіб:

$$R[M\theta N\rangle S=(\,R[M\ \theta N]S)[L,M]$$

Приклад виконання операції природного з'єднання:

R

A	*B*	*C*	*D*
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_1
a_1	b_2	c_1	d_1
a_2	b_3	c_1	d_1
a_2	b_4	c_2	d_3

S

C	*D*	*E*
c_1	d_1	e_2
c_2	d_1	e_3
c_2	d_1	e_1

$R[C,D=C,D]S$

A	*B*	*C*	*D*	*C*	*D*	*E*
a_1	b_1	c_1	d_1	c_1	d_1	e_2
a_1	b_2	c_1	d_1	c_1	d_1	e_2
a_2	b_3	c_1	d_1	c_1	d_1	e_2
a_1	b_1	c_2	d_1	c_2	d_1	e_3
a_1	b_2	c_2	d_1	c_2	d_1	e_3
a_1	b_1	c_2	d_1	c_2	d_1	e_1
a_1	b_2	c_2	d_1	c_2	d_1	e_1

$R*S$

A	*B*	*C*	*D*	*E*
a_1	b_1	c_1	d_1	e_2
a_1	b_2	c_1	d_1	e_2
a_2	b_3	c_1	d_1	e_2
a_1	b_1	c_2	d_1	e_3
a_1	b_2	c_2	d_1	e_3
a_1	b_1	c_2	d_1	e_1
a_1	b_2	c_2	d_1	e_1

Крім операцій природного з'єднання та тета з'єднання є ще операція зовнішнього з'єднання. Часто буває таке, що при з'єднанні двох відношень кортеж з одного відношення не знаходить відповідного кортежу в іншому відношенні. Інакше кажучи, в атрибутах, за якими відбувається операція з'єднання, виявляються незбіжні значення. Може знадобитися, щоб кортеж з одного або двох відношень був поданий у результаті з'єднання, навіть якщо у відповідних атрибутах немає співпадаючих значень. Ця мета може бути досягнута за допомогою операції зовнішнього з'єднання.

Приклад виконання операції лівого зовнішнього з'єднання:

R

A	*B*	*C*
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_2	c_3
a_2	b_3	c_1
a_2	b_4	c_3

S

C	*D*	*E*
c_1	d_1	e_2
c_2	d_1	e_3
c_2	d_1	e_1

$R\supset\!\lhd S$

A	*B*	*C*	*D*	*E*
a_1	b_1	c_1	d_1	e_2
a_1	b_1	c_2	d_1	e_3
a_2	b_3	c_1	d_1	e_2
a_1	b_1	c_2	d_1	e_1
a_1	b_2	c_3	*null*	*null*
a_2	b_3	c_1	d_1	e_2
a_2	b_4	c_3	*null*	*null*

Відсутні значення у результуючому відношенні позначаються за допомогою визначника *NULL*. Перевагою операції зовнішнього з'єднання є зберігання вихідної інформації: кортежі, які втрачаються при використанні інших типів з’єднань. Наприклад лівим зовнішнім з'єднанням ($R \supset\!\lhd S$) називається операція, коли кортежі відношення R, які не мають співпадаючих значень у спільних стовпцях з відношенням S, також включаються в результуюче відношення.

Ділення. Нехай задано відношення зі схемою $R(M,N)$. *Образом реляційного відношення R за кортежем* $t_1 \in R[M]$ називається така множина кортежів $t_2 \in R[N]$, для яких зчеплення (t_1, t_2) належить відношенню R. Образ R за кортежем t_1 позначається $I_R(t_1)$ і формально визначається у такий спосіб:

$$I_R(t_1)=\{\ t_2 \mid t_2 \in R[N]\ \&\ (t_1, t_2) \in R\}.$$

Приклад:

R

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_3	c_2
a_2	b_1	c_4

$I_R(a_1)$

B	C
b_1	c_1
b_1	c_2
b_3	c_2

$I_R(a_2)$

B	C
b_1	c_4

$I_R(a_1,b_1)$

C
c_1
c_2

$I_R(c_2)$

A	B
a_1	b_1
a_1	b_3

Нехай задано відношення R і S зі схемами $R(M,N)$ та $S(K,L)$, для яких проекції $R[N]$ та $S[K]$ є сумісними. *Діленням* відношення R на відношення S за наборами атрибутів N і K (позначається $R[N \div K]S$) називається операція, результатом якої є відношення Q зі схемою $Q(M)$, що складається з таких кортежів $t \in R[M]$, образи $I_R(t)$ яких містять усі кортежі проекції $S[K]$, тобто:

$$Q= R[N \div K]S=\{t \mid t \in R[M]\&\ I_R(t) \supseteq S[K]\}$$

Можна сказати, що операція ділення задається через інші операції алгебри в такий спосіб:

$$R[N \div K]S= R[M]-((\ R[M]\times S[K])-R)[M]$$

Операція не комутативна й не асоціативна.

Приклад:

R

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_3	c_2
a_2	b_1	c_4

S

C	D
c_1	d_1
c_1	d_2
c_2	d_1
c_2	d_3

$S[C]$

C
c_1
c_2

$R[\mathrm{C} \div C]S$

A	B
a_1	b_1

1.3. Задачі на застосування реляційної алгебри

Нехай задано реляційну схему бази даних вищого навчального закладу (рис. 4):

а)

STUDENT (Студент)	**LECTURER (Викладач)**	**SUBJECT (Предмет навчання)**	**UNIVERSITY (Університети)**	**EXAM_MARKS (Екзаменаційні оцінки)**	**SUBJ_LECT (Навчальні дисципліни викладачів)**
STUDENT_ID (PK) SURNAME NAME STIPEND YEAR_LEARNING CITY BIRTHDAY UNIV_ID_ID (FK)	LECTURER_ID (PK) SURNAME NAME CITY UNIV_ID_ID (FK)	SUBJ_ID (PK) SUBJ_NAME HOUR SEMESTER	UNIV_ID (PK), UNIV_NAME RATING CITY	EXAM_ID (PK) STUDENT_ID_ID (FK) SUBJ_ID_ID (FK) MARK EXAM_DATE	LECTURER _ID_ID (PK) (FK) SUBJ_ID_ID (PK) (FK)

б)

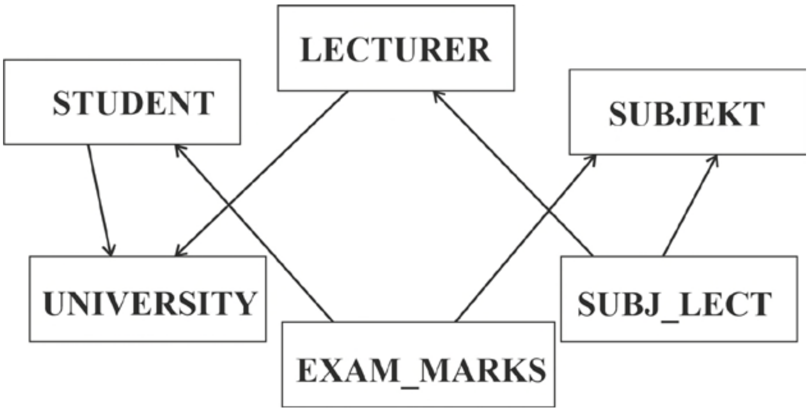


Рис. 4. Реляційна схема бази даних вищого навчального закладу:
а) відношення; б) зв'язки між відношеннями.

Приклади згрупуємо за операціями, застосування яких у них демонструються.

Проекція

1. Отримати прізвища усіх викладачів.

В даному завданні точно вказано, що потрібно отримати відношення, що включає атрибут SURNAME. *Цей атрибут входить в відношення* LECTURER, *отже, треба буде застосувати операцію проекції до відношенню, яке містить необхідні кортежі.*

LECTURER [SURNAME]

$\pi_{\mathrm{SURNAME}}(\mathrm{LECTURER})$

2. Отримати прізвища усіх викладачів і місто, де вони проживають.

В даному завданні потрібно отримати відношення, що включає два атрибути SURNAME *і* CITY *відношення* LECTURER.

LECTURER [SURNAME, CITY]

$\pi_{\mathrm{SURNAME,\ CITY}}(\mathrm{LECTURER})$

3. Отримати прізвища усіх студентів їх день народження та рік навчання.
STUDENT [SURNAME, BIRTHDAY, YEAR_LEARNING]
$\pi_{\text{SURNAME, BIRTHDAY, KURS}}$(STUDENT)

4. Отримати назви всіх університетів та їх рейтинг.
UNIVERSITY [UNIV_NAME, RATING]
$\pi_{\text{UNIV\_NAME, RATING}}$ (UNIVERSITY)

Обмеження

5. Отримати всі відомості про студентів першого року навчання.

У даній задачі потрібно отримати всі дані, тобто отримати весь кортеж відношення STUDENT. Але в результуюче відношення повинні потрапити тільки ті кортежі, що задовольняють умові (YEAR_LEARNING =1). В даному випадку не потрібно виконувати операцію проекції.

STUDENT [YEAR_LEARNING =1]
$\sigma_{\text{YEAR\_LEARNING =1}}$(STUDENT)

6. Отримати всі відомості про предмети, які вивчаються в 5 семестрі і на вивчення яких відводиться більше 100 годин.
SUBJECT [(SEMESTER=5) AND (HOUR>100)]
$\sigma_{\text{(SEMESTER=5) AND (HOUR>100)}}$(SUBJECT)

7. Отримати всі відомості про університети, які знаходяться в місті 'Lviv' або 'Lutsk'.
UNIVERSITY [(CITY='Lviv') OR (CITY='Lutsk')]
$\sigma_{\text{(CITY='Lviv') OR (CITY='Lutsk')}}$(UNIVERSITY)

Використання констант у наведених прикладах операції θ-обмеження є відступом від теоретичної „чистоти” реляційної моделі, але розробники мов на основі реляційної алгебри використовували подібні відступи задля зручності користувача та практичної ефективності. Для того, щоб написати теоретично „чистий” варіант даних запитів, потрібно використати допоміжну таблицю з єдиним кортежем, який містить значення константи і виконати з’єднання використаної в запиті таблиці і допоміжної таблиці.

Обмеження з проекцією

8. Отримати прізвища всіх студентів першого року навчання.

В даній задачі потрібно отримати всі відомості про студентів першого року навчання (STUDENT [YEAR_LEARNING =1]), і в отриманому відношенні виконати операцію проекції на атрибут SURNAME.

(STUDENT [YEAR_LEARNING =1])[SURNAME]
$\pi_{\text{SURNAME}}(\sigma_{\text{YEAR\_LEARNING =1}}$(STUDENT))

9. Отримати прізвища та імена усіх викладачів які проживають в місті 'Lviv'.

(LECTURER[CITY='Lviv'])[SURNAME, NAME]

$\pi_{\text{SURNAME, NAME}}$ ($\sigma_{\text{CITY='Lviv'}}$ (LECTURER))

10. Отримати прізвища усіх студентів першого року навчання та їх день народження.

(STUDENT [YEAR_LEARNING =1])[SURNAME, BIRTHDAY]

$\pi_{\text{SURNAME, BIRTHDAY}}$($\sigma_{\text{YEAR\_LEARNING =1}}$ (STUDENT))

11. Отримати назви всіх університетів міста 'Lviv', якщо його рейтинг знаходиться в межах (10–40).

UNIVERSITY[(CITY='Lviv') AND (RATING>10) AND (RATING<40)] [UNIV_NAME]

$\pi_{\text{UNIV\_NAME}}$ ($\sigma_{\text{(CITY='Lviv') AND (RATING>10) AND (RATING<40)}}$ (UNIVERSITY)

З'єднання

12. Отримати всі відомості про студентів, та всі відомості про їх оцінки (всі дані з таблиці STUDENT та всі дані з таблиці EXAM_MARKS)

Особливістю цього запиту порівняно з попередніми є те, що шукані значення розташовані в різних таблицях. Тому потрібно з'єднати два відношення STUDENT *і* EXAM_MARKS *при умові, що значення коду студента таблиці* STUDENT *буде рівним значенню коду студента таблиці* EXAM_MARKS [STUDENT_ID=STUDENT_ID_ID].

STUDENT[STUDENT_ID=STUDENT_ID_ID] EXAM_MARKS

STUDENT $\bowtie_{\text{STUDENT\_ID=STUDENT\_ID\_ID}}$ EXAM_MARKS

З'єднання, обмеження, проекція

13. Отримати прізвища студентів та їх оцінки.

В даній задачі дані про прізвища студентів знаходяться в таблиці STUDENT, а дані про оцінки в таблиці EXAM_MARKS, тому потрібно виконати операцію з'єднання, як у задачі 12, і в отриманому відношенні виконати проекцію на атрибути [SURNAME, MARK].

(STUDENT[STUDENT_ID=STUDENT_ID_ID] EXAM_MARKS) [SURNAME, MARK]

$\pi_{\text{SURNAME, MARK}}$(STUDENT $\bowtie_{\text{STUDENT\_ID=STUDENT\_ID\_ID}}$ EXAM_MARKS)

14. Отримати прізвища студентів та назви предметів по яких вони здавали екзамени та дати здачі цих екзаменів.

В даній задачі дані отримуються з трьох таблиць: STUDENT, EXAM_MARKS, SUBJECT. Ці таблиці нам потрібно з'єднати. Спочатку з'єднаємо таблиці STUDENT та EXAM_MARKS (STUDENT[STUDENT_ID=STUDENT_ID_ID] EXAM_MARKS), а потім до отриманої таблиці доєднаємо таблицю SUBJECT ((STUDENT[STUDENT_ID=STUDENT_ID_ID] EXAM_MARKS) [SUBJ_ID_ID = SUBJ_ID] SUBJECT). Остання дія, виконати проекцію з отриманого відношення на задані в умові атрибути.

((STUDENT[STUDENT_ID=STUDENT_ID_ID] EXAM_MARKS) [SUBJ_ID_ID = SUBJ_ID] SUBJECT) [SURNAME, EXAM_DATE, SUBJ_NAME]

$\pi_{SURNAME, EXAM\_DATE, SUBJ\_NAME}$ ((STUDENT $\bowtie_{STUDENT\_ID=STUDENT\_ID\_ID}$ EXAM_MARKS) $\bowtie_{SUBJ_ID_ID = SUBJ_ID}$SUBJECT

15.Отримати прізвища викладачів, та предмети, які вони викладають

Таблиці LECTURER та SUBJECT поєднані між собою через таблицю SUBJ_LECT, тому участь в запиті будуть приймати три таблиці. Спочатку з'єднаємо таблиці LECTURER та SUBJ_LECT, **(***LECTURER[LECTURER _ID=LECTURER _ID_ID] SUBJ_LECT) а потім до отриманої таблиці доєднаємо таблицю SUBJECT – ((LECTURER[LECTURER _ID=LECTURER _ID_ID] SUBJ_LECT) [SUBJ_ID= SUBJ_ID_ID] SUBJECT). Після цього потрібно отримати проекцію на поля SURNAME та SUBJ_NAME.*

((LECTURER[LECTURER_ID=LECTURER_ID_ID]SUBJ_LECT) [SUBJ_ID= SUBJ_ID_ID] SUBJECT) [SURNAME, SUBJ_NAME]

$\pi_{SURNAME, SUBJ\_NAME}$((LECTURER$\bowtie_{LECTURER_ID=LECTURER_ID_ID}$SUBJ_LECT) $\bowtie_{SUBJ\_ID= SUBJ\_ID\_ID}$ SUBJECT)

16.Отримати прізвища студентів та їх оцінки з предмету 'Programming'.

В даному випадку необхідність використання операції з'єднання зумовлена тим, що поля результату і аргумент пошуку і перебувають в різних таблицях. *Дані вибираються з двох таблиць STUDENT та EXAM_MARKS, а умова пошуку ставиться до поля третьої таблиці SUBJECT. Для обчислення цього запиту спочатку здійснюється з'єднання трьох відношень (STUDENT, EXAM_MARKS, SUBJECT) за рівністю первинних і зовнішніх ключів, потім вибираються ті кортежі, які стосуються предмету 'Programming' і нарешті здійснюється проекція за необхідними атрибутами.*

(((STUDENT[STUDENT_ID=STUDENT_ID_ID] EXAM_MARKS) [SUBJ_ID_ID= SUBJ_ID] SUBJECT) [SUBJ_NAME='Programming']) [SURNAME, MARK]

$\pi_{SURNAME, MARK}$ ($\sigma_{SUBJ\_NAME='Programming'}$ ((STUDENT $\bowtie_{STUDENT\_ID=STUDENT\_ID\_ID}$ EXAM_MARKS) $\bowtie_{SUBJ\_ID\_ID = SUBJ\_ID}$ SUBJECT))

17.Отримати прізвища студентів, які вивчають предмет 'Programming' у викладача з прізвищем 'Makarchuk'. Також вивести оцінку з даного предмета.

У даному запиті участь приймають п'ять таблиць, які потрібно з'єднати. Крім того в таблицях, що приймають участь у запиті є однойменні стовпці, то, будуючи запит, слід уточнити імена атрибутів іменами відношень.

(((((STUDENT[STUDENT.STUDENT_ID= EXAM_MARKS.STUDENT_ID_ID] EXAM_MARKS) [EXAM_MARKS.SUBJ_ID_ID= SUBJECT.SUBJ_ID] SUBJECT) [SUBJECT.SUBJ_ID = SUBJ_LECT.SUBJ_ID_ID] SUBJ_LECT) [SUBJ_LECT.LECTURER_ID_ID = LECTURER.LECTURER_ID] LECTURER) [SUBJECT.SUBJ_NAME='Programming' AND LECTURER.SURNAME='Makarchuk']) [STUDENT.SURNAME, EXAM_MARKS.MARK]

$\pi_{\text{STUDENT.SURNAME, EXAM\_MARKS.MARK}}(\sigma_{\text{SUBJECT.SUBJ\_NAME='Programming' AND LECTURER.SURNAME='Makarchuk'}}((((\text{STUDENT} \bowtie_{\text{STUDENT.STUDENT\_ID= EXAM\_MARKS.STUDENT\_ID\_ID}} \text{EXAM\_MARKS}) \bowtie_{\text{EXAM\_MARKS.SUBJ\_ID\_ID = SUBJECT.SUBJ\_ID}} \text{SUBJECT}) \bowtie_{\text{SUBJECT.SUBJ\_ID = SUBJ\_LECT.SUBJ\_ID\_ID}} \text{SUBJ\_LECT}) \bowtie_{\text{SUBJ\_LECT.LECTURER\_ID\_ID = LECTURER.LECTURER\_ID}} \text{LECTURER}))$

Послідовність виконання операцій запиту можна зобразити графічно, як дерево, що його листки позначають відношення, а інші вершини – операції (рис. 5).

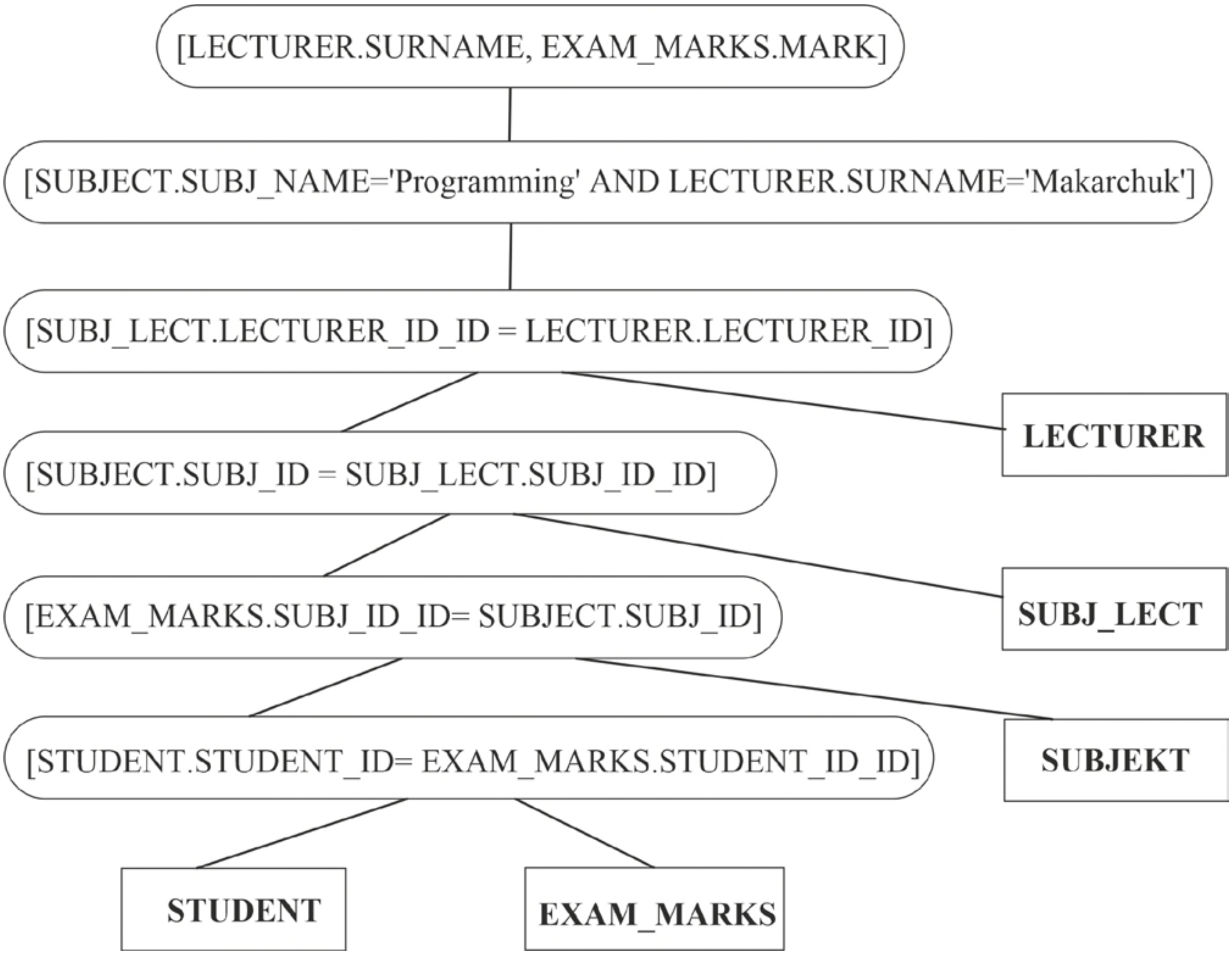


Рис. 5. Схема виконання запиту завдання 17.

В тих випадках, коли виникають проблеми з записом запиту формальною мовою, для зручності можна побудувати схему його виконання.

Теоретико-множинні операції

18. Отримати прізвища та імена усіх студентів і викладачів, які проживають в місті 'Lviv'.

(STUDENT [CITY='Lviv'])[SURNAME, NAME] ∪
(LECTURER[CITY='Lviv'])[SURNAME, NAME]

$\pi_{\text{SURNAME, NAME}}(\sigma_{\text{CITY='Lviv'}}(\text{STUDENT})) \cup \pi_{\text{SURNAME, NAME}}(\sigma_{\text{CITY='Lviv'}}(\text{LECTURER}))$

У першій частині запиту (до символу „$\cup$”) ми знаходимо всіх студенів, які проживають в місті 'Lviv', а в другій частині – отримуємо список викладачів, які проживають в місті 'Lviv'. Операція об’єднання зводить в одну таблицю (без повторень) вміст двох сумісних між собою таблиць-операндів.

19.Отримати міста, де проживають і викладачі і студенти?

В даному завданні ми маємо операцію перетину.

STUDENT [CITY] $\cap$ LECTURER[CITY]

π_{CITY} (STUDENT) $\cap$ π_{CITY}(LECTURER)

20.Хто з викладачів не викладає жодної дисципліни?

Цей запит доцільно будувати за методом „від супротивного”, тобто від усієї множини викладачів (точніше їх прізвищ) віднімається множина тих, які викладають хоча б одну дисципліну.

(LECTURER[SURNAME, NAME]-
((LECTURER[LECTURER_ID=LECTURER_ID_ID]SUBJ_LECT)[SUBJ_ID_ID= SUBJ_ID] SUBJECT) [SURNAME, NAME]

$\pi_{SURNAME,\ NAME}$ (LECTURER)- $\pi_{SURNAME,\ NAME}$ (((LECTURER $\bowtie_{LECTURER\_ID=LECTURER\_ID\_ID}$ SUBJ_LECT) $\bowtie_{SUBJ\_ID\ \_ID\ =\ SUBJ\_ID}$ SUBJECT))

Операція ділення

21.Отримати ідентифікаційні номери студентів, які здали всі предмети.

((STUDENT[STUDENT_ID=STUDENT_ID_ID] EXAM_MARKS) [SUBJ_ID_ID= SUBJ_ID] SUBJECT) [STUDENT_ID, SUBJ_ID]
÷ SUBJECT

$\pi_{STUDENT\_ID,\ SUBJ\_ID}$ ((STUDENT$\bowtie_{STUDENT_ID=STUDENT_ID_ID}$ EXAM_MARKS) $\bowtie_{EXAM\_MARKS.SUBJ\_ID\_ID\ =\ SUBJECT.SUBJ\_ID}$ SUBJECT) ÷ SUBJECT

2. ОПТИМІЗАЦІЯ ОБЧИСЛЕННЯ ВИРАЗІВ РЕЛЯЦІЙНОЇ АЛГЕБРИ

2.1. Властивості операцій реляційної алгебри. Еквівалентні перетворення

Опишемо основні властивості операцій реляційної алгебри, на яких базуються правила еквівалентних перетворень її виразів. Два вирази реляційної алгебри називаються *еквівалентними*, якщо за будь-яких значень реляційних відношень, що входять до їхнього складу, результати обчислення виразів збігаються. Правила еквівалентних перетворень дають змогу вирішувати проблему оптимізації виконання запитів реляційної алгебри.

Наведемо основні властивості операцій реляційної алгебри.

1. Комутативність, асоціативність та дистрибутивність теоретико-множинних операцій об'єднання, перетину і різниці.
2. Ідемпотентність проекцій.

   Нехай L і M – множини атрибутів деякого реляційного відношення R. Якщо $L \subseteq M$, то

$\pi_L(\pi_M R) = \pi_L R$.

3. Дистрибутивність проекції з теоретико-множинними операціями, декартовим добутком, з'єднанням, селекцією.

   Нехай K– певна підмножина атрибутів реляційного відношення R, P– підмножина атрибутів реляційного відношення S і $N=K \cup P$. Тоді:

   а) $\pi_N(R \cup S) = \pi_N R \cup \pi_N S$;

   б) $\pi_N(R \cap S) = \pi_N R \cap \pi_N S$;

   в) $\pi_N(R - S) = \pi_N R - \pi_N S$;

   г) $\pi_N(R \times S) = \pi_K R \times \pi_P S$;

   д) $\pi_N(R \bowtie_F S) = \pi_K R \bowtie_F \pi_P S$ (якщо в умові F використовуються лише атрибути з множини N);

   е) $\pi_K \sigma_F R = \sigma_F \pi_K R$ (якщо в умові F використовуються лише атрибути з множини K).

4. Ідемпотентність (комутативність) селекцій:

$\sigma_F(\sigma_G(R)) = \sigma_G(\sigma_F(R)) = \sigma_{F\&G}(R)$.

5. Комутативність селекції з декартовим добутком:

   а) $\sigma_F(R \times S) = \sigma_F(R) \times S$ (якщо в умові F використовуються лише атрибути з множини R);

   б) $\sigma_{F1\&F2}(R \times S) = \sigma_{F2}(\sigma_{F1}(R \times S)) = \sigma_{F1}(R) \times \sigma_{F2}(S)$ (якщо всі атрибути з F_1 містяться у відношенні R і всі атрибути з F_2 міститься у відношенні S);

   в) $\sigma_{F1\&F2\&F3}(R \times S) = \sigma_{F3}(\sigma_{F1}(R) \times \sigma_{F2}(S))$ (якщо всі атрибути з F_1 містяться у відношенні R і всі атрибути з F_2 міститься у відношенні S).

6. Комбінування селекції з декартовим добутком та θ-з'єднанням:

а) $\sigma_{K\theta P}(R\times S)= R\bowtie_{K\theta P}S$;
б) $\sigma_G(R\bowtie_F S)= R\bowtie_{G\theta F}S$; (атрибути з G належать відношенню R, атрибути з F – відношенню S).

7. Дистрибутивність селекції з теоретико-множинними операціями.

Нехай атрибути з умови F входять до складу як реляційного відношення R, так і реляційного відношення S. Тоді:
а) $\sigma_F(R\cup S)= \sigma_F(R)\cup\sigma_F(S)$;
б) $\sigma_F(R\cap S)= \sigma_F(R)\cap\sigma_F(S)$;
в) $\sigma_F(R-S)= \sigma_F(R)-\sigma_F(S)$;

8. Комутативність селекції і з'єднання:
а) якщо всі атрибути з логічного виразу G містяться в реляційному відношенні R, то
$\sigma_G(R\bowtie_F S)= \sigma_G(R) \bowtie_F S$;
б) якщо $G=G_1\&G_2$, всі атрибути з G_1 містяться у реляційному відношенні R і всі атрибути з G_2 – у реляційному відношенні S, то
$\sigma_G(R\bowtie_F S)= \sigma_{G1}(R) \bowtie_F \sigma_{G2}(S)$;

9. Комутативність та асоціативність добутку:
а) $R\times S=S\times R$– комутативність;
б) $R\times(S\times T)=(R\times S) \times T$ – асоціативність.

10. Комутативність та асоціативність з'єднання:
а) $R\bowtie_F S=S\bowtie_F R$ – комутативність;
б) $R\bowtie_F(S\bowtie_F T)=(R\bowtie_F S)\bowtie_F T$ – асоціативність.

11. Дистрибутивність з'єднання з теоретико-множинними операціями:
а) $R\bowtie_F(S\cup T)=(R\bowtie_F S)\cup(R\bowtie_F T)$;
б) $R\bowtie_F(S\cap T)=(R\bowtie_F S)\cap(R\bowtie_F T)$;
в) $R\bowtie_F(S-T)=(R\bowtie_F S)-(R\bowtie_F T)$.

2.2. Використання реляційної алгебри для оптимізації запитів

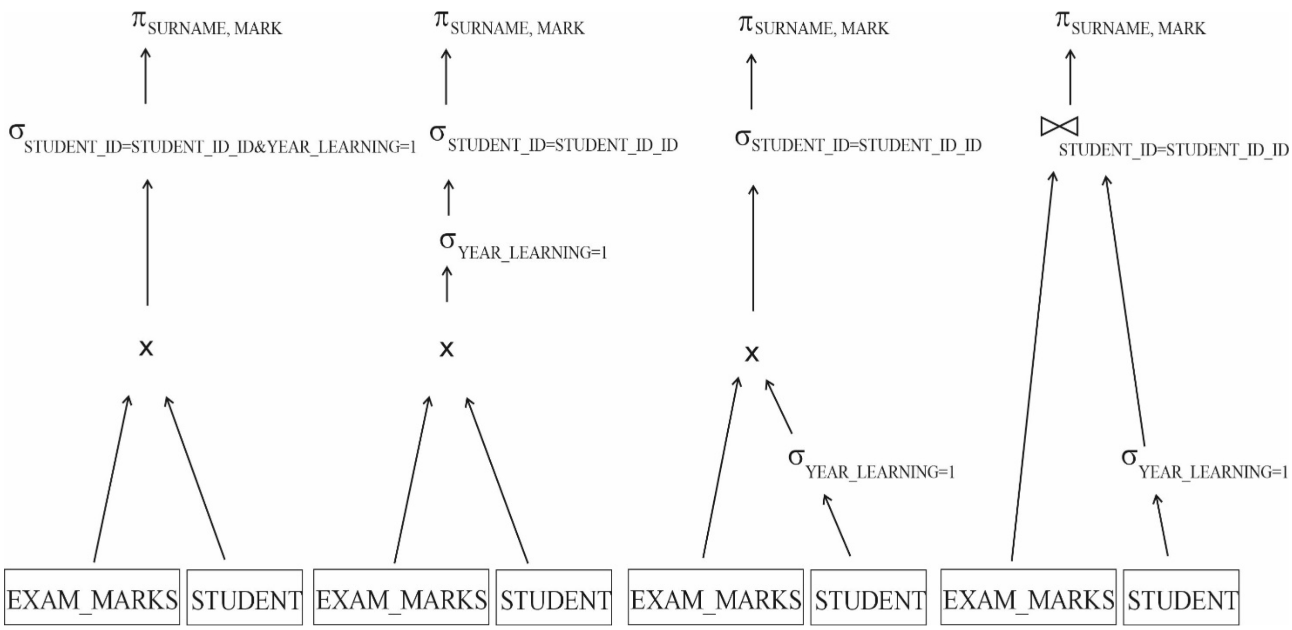
Описані в п. 2.1. властивості операцій реляційної алгебри дають змогу вирішувати завдання логічної оптимізації алгебраїчних виразів. Під терміном *логічна оптимізація* ми маємо на увазі оптимізацію, що дає можливість прискорити обчислення реляційних виразів незалежно від способів реалізації реляційних відношень. На відміну від алгебри числових виразів, складність виконання формул реляційної алгебри залежить не лише від кількості операцій, але й від розміру операндів.

Розглянемо приклад оптимізаційних перетворень реляційного виразу. Нехай потрібно отримати прізвища студентів першого року навчання, та їх оцінки. (реляційна схема рис. 5), Один з варіантів розв'язку може бути наступним: $\pi_{\text{SURNAME, MARK}}((\sigma_{\text{STUDENT\_ID=STUDENT\_ID\_ID\&YEAR\_LEARNING =1}}$ $(\text{STUDENT}\times\text{EXAM\_MARKS})$

Можливі шляхи послідовних еквівалентних перетворень, що оптимізують обчислення цього виразу, наведені на рис. 6. Оптимізаційні перетворення здійснюються згідно з властивостями, описаними в п. 2.1.

Наведемо тепер основні правила оптимізації виразів реляційної алгебри.

1. Кожна селекція $\sigma_{F1\&\ldots\&Fn}(E)$ згідно з властивістю 4 подається у вигляді послідовності селекцій $\sigma_{F1}(\ldots(\sigma_{Fn}(E)\ldots))$.
2. Кожна селекція переміщується деревом виразу вниз наскільки це можливо (властивості 3е, 5, 7, 8). У такий спосіб зменшується кардинальність відношень.
3. Розташовані поруч селекції і декартові добутки замінюються з'єднанням, якщо це допускається властивістю 6.
4. Кожна проекція переміщується деревом виразу вниз наскільки це можливо (властивості 2, 3). У такий спосіб зменшується ступінь відношень.
5. Кожен каскад селекцій і проекцій перетворюється на одиничну селекцію, одиничну проекцію чи селекцію з наступною проекцією (властивості 2, 3е, 4). Перетворення може суперечити твердженню «роби проекцію якомога раніше», однак ефективніше виконати всі можливі операції селекції і проекції за один перегляд відношення, ніж здійснювати кілька переглядів.



а *б* *в* *г*

Рис. 6. Приклад еквівалентних перетворень, що оптимізують вирази реляційної алгебри: *а* – вихідний вираз обчислюється так, як він записаний; *б* – селекція поділяється на каскад селекцій відповідно до властивості 4; *в* – одна з отриманих селекцій опускається нижче декартового добутку відповідно до властивості 5а; *г* – декартовий добуток і селекція замінюються на з'єднання згідно з властивістю 6.

3. РЕЛЯЦІЙНЕ ЧИСЛЕННЯ КОДДА

3.1. Завдання реляційного числення

Завдання реляційного числення полягає у тому, щоб можна було сформулювати спеціальне числення предикатів, яке інтерпретується реляційними відношеннями.

З множини формул числення предикатів варто вибрати таку підмножину, і яка давала б змогу, одержувати формули, що інтерпретуються скінченними відношеннями.

Більшість працюючих мов запитів засновані на реляційному численні. Реляційні обчислення – непроцедурні системи. Обчислення висловлюють тільки те, яким повинен бути результат обчислення, але не те, яким чином проводити обчислення. Це завдання покладається на процесор мови запитів даної СУБД.

Розрізняють реляційне числення кортежів і реляційне числення доменів. Оскільки реляційне числення доменів схоже з реляційним обчисленням кортежів за винятком того, що змінні приймають значення в доменах, а не є кортежами, а числення кортежів використовується частіше, розглянемо тільки числення кортежів.

Реляційне числення кортежів є, по суті, формалізацією системи позначень, призначеної для утворення множин. В реляційному численні використовуються булеві операції (I, АБО, НЕ) над умовами, які можуть бути істинними або помилковими. У ньому також використовуються квантори існування і загальності, які означають, відповідно, що елемент певного типу існує або що умова істинна для кожного елемента певного типу.

3.2. Цільовий список і визначальний вираз

Розглянемо реляційне відношення бази даних вищого навчального закладу:

r – **UNIVERSITY** (UNIV_ID, UNIV_NAME, RATING, CITY)

Запит: Які університети знаходяться в місті 'Lutsk'?

У реляційній алгебрі для виконання цього запиту необхідно використовувати вираз, який повинен містити наступні операції:

- операцію Вибір над відношенням r, результатом її застосування до відношення r є інше відношення, яке представляє собою підмножину кортежів відношення r зі значенням, рівним 'Lutsk' в атрибуті CITY;
- проекція результату попередньої операції на атрибут UNIV_NAME.

У реляційному численні формулювання цього запиту повинен мати наступний вигляд:

{t.UNIV_NAME | t in r and t.CITY = 'Lutsk'},

де t – це змінна, що позначає довільний рядок.

Відношення, з якого береться t, визначається виразом "in r" яке означає, що t – рядок відношення. t.UNIV_NAME – значення атрибута UNIV_NAME в рядку r; символ (|) – розділяє цільової список і визначальний вираз. В даному випадку:

t.UNIV_NAME – цільовий список;

t in r and t.CITY = 'Lutsk' – визначальний вираз;

t.CITY = 'Lutsk' означає, що значення атрибута CITY в рядку t дорівнює 'Lutsk'.

Фігурні дужки "{}", в які укладено вираз, визначають результат запиту, як множину значень даних. Що саме входить в цю множину, описує вираз в дужках.

Розв’язком кожного запиту в реляційному численні є відношення, яке задається цільовим списком і визначальним виразом. Цільовий список визначає атрибути відношення розв’язку. Визначальний вираз – це умова, на підставі якої відбираються значення з бази даних, які увійдуть до відношення розв’язку.

У попередньому прикладі UNIV_NAME береться з рядка і поміщається в рядок розв’язку, якщо рядок t задовольняє умові t in r and t.CITY = 'Lutsk' . Система переглядає рядки відношення r один за одним. Першому рядку тимчасово присвоюється ім’я t і перевіряється істинність визначального виразу. Якщо значення виразу істинне, то рядок поміщається у відношення розв’язку. Потім система переходить до наступного рядка, дає їй ім’я t і знову перевіряє істинність визначального виразу. Процес повторюється для кожного рядка відношення r.

Цільовий список може складатися з декількох атрибутів, розділених комами.

{t.UNIV_NAME, t. RATING, t. CITY | t in r and t.CITY = 'Lutsk'},

У визначальному виразі, умова, за якою відбираються записи в результуюче відношення, будується за допомогою стандартних операцій логіки AND, OR, NOT, шести операцій порівняння (=,! =,>,> =, <, <=) і оператора IN.

Таким чином, розглянутий приклад демонструє запис операцій вибірки і створення проекцій. Крім того, з розглянутих вже конструкцій можна отримати і всі аналоги теоретико-множинних операцій. Розглянемо теоретико-множинні операції на прикладах, при цьому ми повинні пам’ятати, що реляційні відношення повинні бути сумісними.

Нехай дані два відношення:

r – **LECTURER1** (LECT_ID, NAME, CITY, PHONE)

s – **LECTURER2** (LECT_ID, NAME, CITY, PHONE)

Нехай ми маємо алгебраїчний вираз $\pi_{CITY}(r \cap s)$. Даний вираз містить операцію перерізу і операцію проекції.

У реляційному ж численні, якщо *t* кортеж *r*, а *y* кортеж *s*, цей запит може включати в себе наступні компоненти:

t.CITY – цільовий список;

t in *r* and *y* in *s* and *t*.LECT_ID = *y*. LECT_ID and *t*. NAME = *y*. NAME and *t*.CITY = *y*.CITY and *t*.PHONE = *y*.PHONE – визначальний вираз.

{*t*.CITY| *t* in *r* and *y* in *s* and *t*.LECT_ID = *y*. LECT_ID and *t*. NAME = *y*. NAME and *t*.CITY = *y*.CITY and *t*.PHONE = *y*.PHONE}

Це означає, що всі поля кортежа *t* рівні полям кортежа *y*. У результуючий набір взято ті кортежі, які є і у *r*-відношенні і у *s*- відношенні.

В даному випадку не обв’зково порівнювати відповідні поля. Тобто даний вираз можна записати:

{*t*.CITY| *t* in *r* and *t* in *s*}

Аналогічно можна отримати конструкції реляційного числення, які відповідають операціям об’єднання, різниці і декартового добутку.

Напрприклад розглянемо вираз, що містить операцію об’єднання: $\pi_{CITY}(r \cup s)$.

t.CITY – цільовий список;

t in *r* or *t* in *s* – визначальний вираз.

{*t*.CITY| *t* in *r* or *t* in *s* }

Розглянемо вираз, що містить операцію різниці: $\pi_{CITY}(r\text{-}s)$.

Результатом буде множина кортежів:

{*t*.CITY| *t* in *r* and *t* not in *s*}

Операцію декартового добутку розглянемо для двох відношень:

r – **LECTURER** (LECT_ID, NAME, CITY, PHONE, UNIV_ID)

s – **UNIVERSITY (**UNIV_ID, UNIV_NAME, CITY, RATING**)**

r – **LECTURER**

LECT_ID	NAME	CITY	UNIV_ID
1	Victor	Uzhhorod	1
2	Semen	Uzhhorod	1
3	Stepan	Kyiv	2

s – **UNIVERSITY**

UNIV_ID	UNIV_NAME	CITY	RATING
1	Uzhhorod National University	Uzhhorod	67
2	Taras Shevchenko National University of Kyiv	Kyiv	1

($r \times s$):

{*t*.LECT_ID, *t*.NAME, *t*.CITY, *t*. UNIV_ID, *y*. UNIV_ID, *y*. UNIV_NAME, *y*.CITY, *y*. RATING, | *t* in *r* and *y* in *s* }

Результатом буде наступна множина:

r.LECT_ID	*r*.NAME	*r*.CITY	*r*.UNIV_ID	*s*.UNIV_ID	*s*.UNIV_NAME	*s*.CITY	*s*.RATING
1	Victor	Uzhhorod	1	1	Uzhhorod National University	Uzhhorod	67
2	Semen	Uzhhorod	1	1	Uzhhorod National University	Uzhhorod	67
3	Stepan	Kyiv	2	1	Uzhhorod National University	Uzhhorod	67
1	Victor	Uzhhorod	1	2	Uzhhorod National University	Kyiv	1
2	Semen	Uzhhorod	1	2	Taras Shevchenko National University of Kyiv	Kyiv	1
3	Stepan	Kyiv	2	2	Taras Shevchenko National University of Kyiv	Kyiv	1

Аналізуючи результат виконання запиту, ми бачимо, що в деяких рядках містяться несумісні дані: в одному рядку дані про викладача і дані про університет іншого викладача. Для того, щоб отримати коректні дані, в результуючий набір нам потрібно взяти тільки відповідні дані, тобто ті дані де поле *r*.UNIV_ID = *s*.UNIV_ID.

{*t*.LECT_ID, *t*.NAME, *t*.CITY, *t*. UNIV_ID, *y*. UNIV_ID, *y*. UNIV_NAME, *y*.CITY, *y*. RATING, | *t* in *r* and *y* in *s* and *r*.UNIV_ID = *s*.UNIV_ID. }

Трохи інакше виглядають аналоги тільки двох операцій реляційної алгебри – операцій з'єднання і ділення. для побудови аналогів цих операцій потрібні квантори: квантор існування для з'єднання і квантор загальності для поділу.

3.3. Квантор існування

Квантор існування означає, що існує хоча б один екземпляр певного типу об'єктів. У реляційному численні квантор існування використовується для завдання умови того, що певний тип рядків у відношенні існує.

Розглянемо відношення:

LECTURER (<u>LECT_ID</u>, NAME, CITY, PHONE, UNIV_ID)

UNIVERSITY (<u>UNIV_ID (PK)</u>,UNIV_NAME, RATING, CITY**)**

Запит: Вивести прізвища викладачів, якщо викладач прикріплений до якогось університету.

Цільовим списком тут буде: *t*.NAME.

де *t* – рядок з відношення LECTURER, а *y* – рядок відношення UNIVERSITY.

Формування визначального виразу здійснюється, виходячи з такого. Для того щоб прізвище викладача увійшло в відношення розв'язку, має задовольнятися умова, що є такий університет до якого він прикріплений. Іншими словами, якщо поле UNIV_ID зустрічається в рядку відношення LECTURER і існує таке значення UNIV_ID у відношенні UNIVERSITY, то поле NAME повинно бути включене в розв'язок. Таким чином, умова така: існує хоча б один рядок у відношенні UNIVERSITY, що містить необхідне значення UNIV_ID. Це формується таким чином:

exists y in UNIVERSITY (*y*.UNIV_ID = *t*.UNIV_ID).

Такий вираз читається: «Існує рядок *y* у відношенні UNIVERSITY такий, що *y*.UNIV_ID = *t*.UNIV_ID». Наведений вираз визначає рядок *t*. Якщо він істинний, тобто для рядка *t* існує такий рядок *y*, то *t*. Поле NAME, поміщається в результуюче відношення. Якщо вираз помилковий – тобто такого рядка у не існує – тоді *t*.NAME, не поміщається в результуюче відношення. Повний розв'язок в реляційному численні виглядає наступним чином:

{*t*.NAME | *t* in LECTURER and exists *y* in UNIVERSITY (*y*.UNIV_ID = *t*.UNIV_ID)}.

У реляційній алгебрі для виконання цього запиту потрібно виконати операцію з'єднання. Таким чином, квантор існування використовується в реляційному обчисленні, для того щоб виконувати функцію з'єднання.

3.4. Квантор загальності

Квантор загальності означає, що деяка умова застосовується до всіх рядків або до кожного рядка деякого типу. Він використовується в тих же цілях, що і операція ділення реляційної алгебри.

Нехай дано відношення:

STUDENT_V (Відомість_Студента) (SURNAME, NAME, SUBJ, EXAM_DATE)

EXAM (Розклад_екзаменів) (SUBJ, EXAM_DATE)

Потрібно визначити прізвища тих студентів, кожен з яких здавав всі екзамени перераховані в таблиці **EXAM**. Необхідно звернути увагу на те, що умова вибору прізвища студента містить визначення кожен. У результуюче відношення включаються тільки ті прізвища студентів, які здавали всі екзамени.

Повний розв’язок в реляційному численні з використанням квантора загальності FORALL наступний:

{t.SURNAME | t in STUDENT_V and forall *s* in EXAM (t.SUBJ = *s*.SUBJ and t.EXAM_DATE = *s*.EXAM_DATE)},

де *t* – кортеж відношення STUDENT;

s – кортеж відношення EXAM.

Прізвище студента з стрічки t таблиці STUDENT_V попадає в результуючий набір, якщо визначальний вираз істинний для стрічки t. А визначальний вираз істинний, якщо для кожної стрічки *s* відношення EXAM повинна існувати стрічка t у відношенні STUDENT_V з одним і тим самим прізвищем.

Наприклад маємо заповнені відношення STUDENT_V та EXAM

STUDENT

SURNAME	NAME	SUBJ	EXAM_DATE
Olijnyk	Andrij	Programming	01.05.2020
Bobryk	Ivan	Programming	01.05.2020
Fedun	Pavlo	Programming	01.05.2020
Panasiuk	Vitalij	Programming	01.05.2020
Kotan	Mykola	Discrete Math	13.05.2020
Olijnyk	Andrij	Discrete Math	13.05.2020
Fedun	Pavlo	Discrete Math	13.05.2020
Fedun	Pavlo	Databases	23.05.2020
Bobryk	Ivan	Databases	23.05.2020
Olijnyk	Andrij	Databases	23.05.2020

EXAM

SUBJ	EXAM_DATE
Programming	01.05.2020
Discrete Math	13.05.2020
Databases	23.05.2020

Повна відповідність всім стрічкам відношення EXAM є тільки у студентів Olijnyk Andrij та Fedun Pavlo. Тільки ці студенти здавали всі іспити, тому дані про них і входять в результуючий набір.

3.5. Приклади розв'язування задач на застосування реляційного числення

Розглянемо реляційну схему бази даних вищого навчального закладу подану на рис. 5. Для прикладу будемо розглядати ті самі задачі, що і в п 1.3. Аналогічно приклади згрупуємо за операціями, застосування яких у них демонструється. Пояснення до прикладів залишаються ті самі, що в п.1.3.

Вибірка.

1. Отримати прізвища усіх викладачів.

{t. SURNAME | t in LECTURER }

2. Отримати прізвища усіх викладачів і місто, де вони проживають.

{t. SURNAME, t. CITY | t in LECTURER }

3. Отримати прізвища усіх студентів їх день народження та рік навчання.

{t. SURNAME, t.BIRTHDAY, t.YEAR_LEARNING | t in STUDENT}

4. Отримати назви всіх університетів та їх рейтинг.

{t.UNIV_NAME, t.RATING | t in UNIVERSITY}

Обмеження

5. Отримати всі відомості про студентів першого року навчання.

{t.STUDENT_ID, t.SURNAME, t.NAME, t.STIPEND, t.YEAR_LEARNING, t.CITY, t.BIRTHDAY, t.UNIV_ID_ID| t in STUDENT and t.YEAR_LEARNING =1}

6. Отримати всі відомості про предмети, які вивчаються в 5 семестрі і на вивчення яких відводиться більше 100 годин.

{t.SUBJ_ID, t.SUBJ_NAME, t.HOUR, t.SEMESTER| t in SUBJECT and t.SEMESTER=5 and t.HOUR>100}

7. Вивести всі відомості про університети, які знаходяться в місті 'Lviv' або 'Lutsk'.

{t.UNIV_ID, t.UNIV_NAME, t.RATING, t.CITY| t in UNIVERSITY and ((t.CITY='Lviv') or (t.CITY='Lutsk'))}

Обмеження з проекцією

8. Отримати прізвища всіх студентів першого року навчання.

{t.SURNAME | t in STUDENT and t.YEAR_LEARNING =1}

9. Отримати прізвища та імена усіх викладачів які проживають в місті 'Lviv'.

{t.SURNAME, t.NAME | t in LECTURER and t.CITY='Lviv' },

10. Отримати прізвища усіх студентів першого року навчання та їх день народження.
{t.SURNAME, t.BIRTHDAY | t in STUDENT and t.YEAR_LEARNING =1},

11. Отримати назви всіх університетів міста 'Lviv', якщо його рейтинг знаходиться в межах (10–40).

{t.UNIV_NAME | t in UNIVERSITY and t.CITY='Lviv' and t.RATING>10 and t.RATING<40},

З'єднання

12. Отримати всі відомості про студентів , та всі відомості про їх оцінки (всі дані з таблиці STUDENT та всі дані з таблиці EXAM_MARKS)
В даному запиті, для того *щоб виконувати функцію з'єднання двох таблиць STUDENT і EXAM_MARKS потрібно використати квантор існування.*
{ t.STUDENT_ID, t.SURNAME, t.NAME, t.STIPEND, t.YEAR_LEARNING, t.CITY, t.BIRTHDAY, t.UNIV_ID_ID, *y*.EXAM_ID, *y*.SUBJ_ID_ID, *y*.MARK, *y*.EXAM_DATE | *t* in STUDENT and exists *y* in EXAM_MARKS (*t*.STUDENT_ID = *y*.STUDENT_ID_ID)}.

З'єднання, обмеження, проекція

13. Отримати прізвища студентів та їх оцінки.
{t.SURNAME, *y*.MARK | *t* in STUDENT and exists *y* in EXAM_MARKS (*y*.STUDENT_ID = *t*.STUDENT_ID_ID)}.

14. Отримати прізвища студентів та назви предметів по яких вони здавали екзамени та дати здачі цих екзаменів.
{t.SURNAME, *y*.EXAM_DATE, z. SUBJ_NAME | *t* in STUDENT and exists *y* in EXAM_MARKS ((*y*.STUDENT_ID = *t*.STUDENT_ID_ID) and exists *z* in SUBJECT (*y*.SUBJ_ID_ID =z.SUBJ_ID))}.

15. Отримати прізвища викладачів, та предмети, які вони викладають.
{t.SURNAME, z.SUBJ_NAME | *t* in LECTURER and exists *y* in SUBJ_LECT ((*t*.LECTURER_ID = *y*.LECTURER_ID_ID) and exists *z* in SUBJECT (*y*.SUBJ_ID_ID = z.SUBJ_ID))}.

16. Отримати прізвища студентів та їх оцінки з предмету 'Programming'.
{t.SURNAME, *y*.MARK | *t* in STUDENT and exists *y* in EXAM_MARKS ((t.STUDENT_ID = *y*.STUDENT_ID_ID) and exists *z* in SUBJECT (z.SUBJ_ID = *y*.SUBJ_ID_ID)) and *z*.SUBJ_NAME='Programming'}.

17. Отримати прізвища студентів, які вивчають предмет 'Programming' у викладача з прізвищем 'Makarchuk'. Також вивести оцінку з даного предмета.

{t.SURNAME, *y*.MARK | *t* in STUDENT and exists *y* in EXAM_MARKS ((t.STUDENT_ID = *y*.STUDENT_ID_ID) and exists *z* in SUBJECT (z.SUBJ_ID = *y*.SUBJ_ID_ID) and *p* in SUBJ_LECT ((*z*.SUBJ_ID = *p*.SUBJ_ID_ID) and exists *k* in LECTURER (*p*.LECTURER_ID_ID = *k*.LECTURER_ID))) and *z*.SUBJ_NAME='Programming'}.

Теоретико-множинні операції

18. Отримати прізвища та імена усіх студентів і викладачів, які проживають в місті 'Lviv'.

В даному завданні ми маємо операцію об'єднання.

{t. SURNAME, t. CITY | (t in STUDENT or t in LECTURER) and t.CITY='Lviv'}

19. Отримати міста, де проживають і викладачі і студенти?

В даному завданні ми маємо операцію перетину.

{t. CITY | (t in STUDENT and exists y in LECTURER (t.CITY = y.CITY)}.

20. Отримати прізвища викладачів, які не викладають жодної дисципліни?

В даному завданні ми маємо операцію різниці.

{t.SURNAME | *t* in LECTURER and not(*s* in LECTURER and exists *y* in SUBJ_LECT ((*s*.LECTURER_ID = *y*.LECTURER_ID_ID) and exists *z* in SUBJECT (*y*.SUBJ_ID_ID = z.SUBJ_ID)))}.

Операція ділення

21. Отримати ідентифікаційні номери студентів, які здали всі предмети.

Побудову запиту виконаємо поетапно.

Оскільки потрібні ідентифікаційні номери студентів, то необхідно шукати записи у відношенні STUDENT. Тому цільовий список повинен містити атрибут STUDENT_ID з цього відношення, а визначальний вираз оператор приналежності IN до STUDENT, наприклад, так:

{t.STUDENT_ID | t in STUDENT... }

Оскільки нас цікавлять студенти, котрі складали хоча б один раз КОЖЕН предмет, то до відношення SUBJECT слід застосувати квантор загальності:

{t.STUDENT_ID | t in STUDENT and forall s in SUBJECT... }

І нарешті, оскільки нас цікавлять СТУДЕНТИ, які здавали кожен предмет, то відповідні кортежі повинні існувати в відношенні EXAM_MARKS

{t.STUDENT_ID | *t* in STUDENT and forall s in SUBJECT (exists r IN EXAM_MARKS (r.SUBJ_ID_ID =s.SUBJ_ID and t.STUDENT_ID = r.STUDENT_ID))}

4. ЕКВІВАЛЕНТНІСТЬ РЕЛЯЦІЙНОЇ АЛГЕБРИ І РЕЛЯЦІЙНОГО ЧИСЛЕННЯ

Реляційна алгебра та реляційне числення є формальними мовами запитів для реляційної моделі. Обидві форми є базою для мови SQL, яка використовується в більшості реляційних СУБД. Реляційна алгебра є процедурною мовою, а реляційне числення є декларативною мовою. Розв'язуючи задачі реляційного числення, ми одночасно показували його аналогічність реляційній алгебрі. В табл 1. проілюстровано еквівалентність підходів.

Таблиця 1.

Зведені операції реляційної алгебри та реляційного числення

Операція	**Опис**	**Синтаксис реляційної алгебри**	**Синтаксис реляційного числення**
Проекція	На вході – одне відношення *R*1. Вибір з відношення окремих атрибутів	R1[R1.*поле 1,* *R*1.*поле2, ...*]	{*r.поле1, r.поле2, …*| *r* in *R*1}
Вибірка	На вході – одне відношення *R*1. Вибір з відношення окремих кортежів, які задовольняють умову відбору	*R*1[умова вибору]	{*r.поле1, r.поле2, ...* | *r* in *R*1 and [умова вибору]}
Об'єднання	На вході – два сумісних відношення *R*1 та *R*2. Включення в результуюче відношення всіх кортежів, які входять у хоча б одне відношення.	$R1 \cup R2$	{r.*поле1,* r.*поле2, ...* | r in *R*1 or r in *R*2}
Перетин	На вході – два сумісних відношення *R*1 та *R*2. Включення в результуюче відношення всіх кортежів, які входять в обидва відношення одночасно.	$R1 \cap R2$	{r.*поле1,* r.*поле2, ...* | r in *R*1 and r in *R*2}
Різниця	На вході – два сумісних відношення *R*1 та *R*2. Включення в результуюче відношення всіх кортежів, які входять в *R*1 і не входять в *R*2	*R*1- *R*2	{r.*поле*1, r.*поле*2, ... | r in *R*1 and r not in *R*2}
Декартовий добуток	На вході – два відношення *R*1 та *R*2. Результуюче відношення складається з всіх кортежів *R*1, до кожного з яких приєднані всі кортежі з *R*2	$R1 \times R2$	{r.*поле*1, r.*поле*2, ... , *s.поле*1, *s.поле*2,… | r in *R*1 and s in *R*2}

<table>
<tr><td>З’єднання</td><td>На вході два відношення R1 та R2, які мають атрибути зв’язку R1.n1 та R2.n2. Результуюче відношення містить відповідні дані з обох відношень.</td><td>1. Природне з’єднання R1*R2

2. З’єднання за умовою R1⋈_F R2

3.Зовнішнє з’єднання R⊃⊲S</td><td>1. {r.поле1, r.поле2, ... , s.поле1, s.поле2, … | r in R1and exists s in R2 (s.n1=s.n2) }
2. { r.поле1, r.поле2, ... , s.поле1, s.поле2, … | r in R1 and exists s in R2 (s.n1 <оператор порівняння> s.n2) }
3. { r.поле1, r.поле2, ... , s.поле1, s.поле2, | (r in R1 and exists s in R2 (s.n1 <оператор порівняння> s.n2)) or (r in R1 and not exists s in R2 (s.n1 <оператор порівняння> s.n2))}.</td></tr>
<tr><td>Ділення</td><td>На вході два відношення R1 та R2, причому атрибути R2 мають складати підмножину атрибутів R1. Результат ділення R1/R2 містить ті кортежі, що входять в R1 з кожним кортежем з R2</td><td>R1 / R2</td><td>{r. поле1 | r in R1 and forall s in R2 (exists t in R1 (t.n1=s.n2) and (r.n1=t.n1)) }</td></tr>
</table>

Основні відмінності між реляційною алгеброю та реляційним численням:

- Основна відмінність між реляційною алгеброю та реляційним обчисленням полягає в тому, що реляційна алгебра є процедурною мовою, тоді як реляційне числення є декларативною.
- Реляційна алгебра визначає, як отримати результат, тоді як реляційне числення визначає, яку інформацію повинен містити результат.
- Реляційна алгебра вказує послідовність, в якій операції повинні виконуватися в запиті. З іншого боку, реляційне числення не визначає послідовність операцій, що виконуються в запиті.
- Реляційна алгебра не залежить від домену, тоді як реляційне числення може бути залежним від домену, оскільки ми маємо доменне реляційне числення.
- Мова запитів реляційної алгебри тісно пов’язана з мовою програмування, тоді як реляційний числення тісно пов’язане з природною мовою.

5. ЗАДАЧІ ДЛЯ САМОСТІЙНОЇ ПІДГОТОВКИ ДО КОНТРОЛЬНОЇ РОБОТИ

Задача1. Нехай задано реляційну схему бази даних закладу вищої освіти:

FACULTY (Факультет)	**DEPARTMENT** (Кафедра)	**LECTURER** (Викладач)	**GROUP** (Група)	**LECTURE** (Лекція)	**SUBJECT** (Предмет навчання)	**CLASSROOM** (Аудиторія)
F_ID(PK) NAME (назва) DEAN (декан) BUILDING(корпус FUND (фонд)	D_ID (PK) F_ID (FK) NAME (назва) HEAD (FK) (завідувач) BUILDING (корпус) FUND (фонд)	L_ID (PK) D_ID(FK) SURNAME (прізвище) NAME (ім'я) POST (посада) PHONE (телефон)	G_ID(PK) D_ID(FK) YEAR_LEARNING (рік навчання) NAME (назва) COUNT (кількість) CURATOR(FK) (куратор)	S_ID (PK) (FK) L_ID (PK) (FK) C_ID(PK) (FK) G_ID(PK) (FK) TYPE (тип) DAY (день) WEEK (тиждень)	S_ID (PK) NAME (назва)	C_ID(PK) BUILDING (корпус) NAMBER (номер) COUNT (місткість)

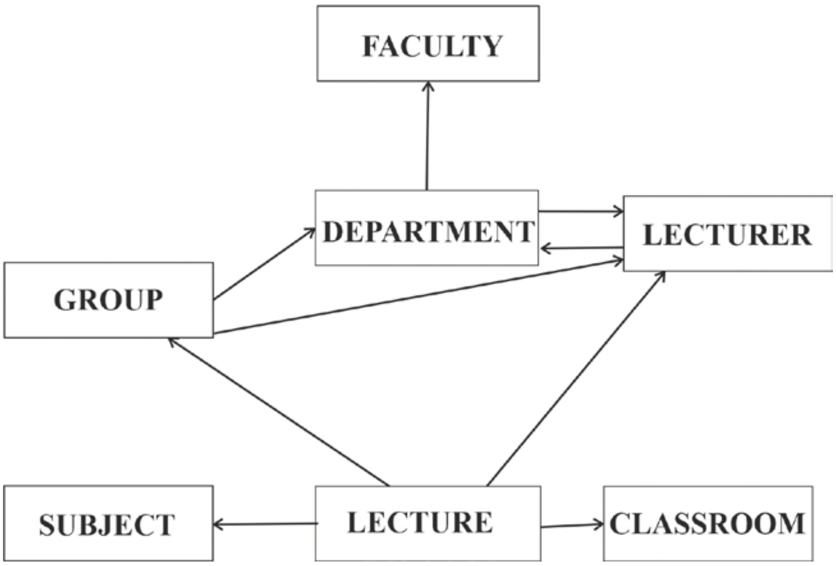


Рис 2. Зв'язки між відношеннями в реляційній схемі вищого навчального закладу

Написати запити використовуючи синтаксис реляційної алгебри та реляційного числення. Побудувати схему виконання запиту.

1. Отримати список усіх викладачів, які мають посаду 'доцент'.
2. Отримати список усіх викладачів, з прізвищем 'Макарчук'.
3. Отримати номери усіх аудиторій 2 корпусу університету.
4. Отримати номери усіх аудиторій які поміщають більше 40 студентів.
5. Отримати номери усіх аудиторій які поміщають більше 40 студентів в корпусі №1.
6. Отримати всі відомості про лекції, які проводяться на 2 тиждні.
7. Отримати всі відомості про лекції, які проводяться у 'Вівторок'.
8. Отримати відомості про лекцію (тип, день, тиждень), якщо її тип 'Оглядова лекція'.
9. Отримати список усіх кафедр 1 корпусу університету.
10. Отримати список усіх факультетів 5 корпусу університету.

11. Отримати список усіх номерів малокомплектних груп (кількість студентів менше15).
12. Отримати список усіх номерів груп першого року навчання.
13. Отримати усі дані про викладачів, які мають прізвище 'Сергієнко'.
14. Отримати прізвище викладача, який має номер телефону '099-220-33-56'.
15. Отримати корпус університету в якому міститься аудиторія 2С.
16. Отримати корпус університету в якому знаходиться факультет 'математики'.
17. Отримати корпус університету в якому знаходиться кафедра 'прикладної математики'.
18. Отримати посаду викладача 'Сергієнко'.
19. Отримати телефон викладача 'Сергієнко'.
20. Отримати всіх деканів корпусу№5 університету.
21. Отримати всі відомості про факультети корпусу №6 університету.
22. Отримати кількість студентів 23 групи.
23. Отримати відомості про кількість студентів та назви груп в групах 1 року навчання.
24. Отримати назви факультетів, які розміщені в корпусі №1 університету.
25. Отримати номери телефонів викладачів, які мають прізвище 'Сергієнко'.
26. Отримати список усіх номерів груп та рік навчання групи , якщо кількість студентів більше 20.
27. Отримати список усіх деканів та назви факультету 7 корпусу університету.
28. Отримати корпус університету в якому знаходиться факультет 'Фізика'.
29. Отримати посаду та номер телефону викладача Сергієнко.
30. Отримати всі відомості про факультети фонд яких перевищує 300000.
31. Отримати всі відомості про кафедри фонд яких перевищує 200000.
32. В якому корпусі університету знаходяться факультети фонд яких перевищує 300000.
33. В якому корпусі університету знаходяться кафедри фонд яких перевищує 200000.
34. Отримати список усіх кафедр факультету інформатики.
35. Отримати список усіх викладачів кафедри АСУ разом із номерами їхніх телефонів.
36. Отримати список усіх викладачів факультету інформатики разом з номерами їхніх телефонів.
37. Отримати список номерів усіх груп першого року навчання кафедри АСУ.
38. Отримати список номерів усіх груп першого року навчання кафедри АСУ разом із прізвищами кураторів цих груп.
39. Вияснити в яких аудиторіях проводяться лекції на 4 тижні.
40. Отримати список усіх кафедр та відомості про те, хто є завідувачем кожної.
41. Отримати прізвище завідувача кафедри математичного аналізу та його номер телефону.
42. Отримати номери всіх груп, їх кураторів та номери телефонів кураторів.

43. Вияснити на якому факультеті, та на якій кафедрі працює викладач з прізвищем Степанюк.
44. Отримати всіх кураторів 2 курсу та номери груп.
45. Отримати на якому тижні буде викладатися предмет 'Програмування'.
46. В яких групах читає лекції викладач з прізвищем Григорчук.
47. В яких аудиторіях проводить лекції викладач з прізвищем Григорчук.
48. Отримати список викладачів, які читають лекції на 4 тижні в 1 корпусі університету.
49. Отримати список викладачів і їх номери телефонів, які читають на 6 тижні лекції з програмування.
50. Отримати список викладачів, факультети яких знаходяться в 5 корпусі університету.
51. Отримати список викладачів які читають лекції на 5 тижні в 23 групі.
52. Отримати список викладачів які читають лекції 503 аудиторії.
53. Отримати список викладачів та номер аудиторії, в якій вони читають лекції на 6 тижні, місткість яких перевищує 40 місць.
54. Отримати список викладачів і їх номери телефонів, які читають лекції типу 'Оглядова лекція' для студентів 43 групи.
55. Отримати прізвище викладача, який має номер телефону '099-220-33-56', якщо він читає лекції в 45 групі.
56. Отримати список усіх кафедр 1 корпусу університету та прізвища викладачів, які належать до даної кафедри.
57. Отримати номери усіх аудиторій які поміщають більше 40 студентів та прізвища викладачів, які читають пари в цих аудиторіях.
58. Отримати прізвища всіх завідувачів кафедр та назву кафедри корпусу №5 університету.
59. Отримати прізвище куратора 24 групи.
60. Отримати назви кафедр, які забезпечують викладання предмету 'Програмування'.
61. Отримати список викладачів, які забезпечують викладання предмету 'Програмування'.
62. Отримати список викладачів, які викладають програмування в 503 аудиторії.
63. Отримати список кафедр, викладачі яких викладають в аудиторії 321 корпусу №1 університету.
64. Отримати список аудиторій в яких викладається предмет програмування, якщо аудиторія місткістю більше 20 місць.
65. Отримати прізвища викладачів та тижні, на яких читається предмет програмування.
66. Отримати список усіх корпусів університету, де розташовуються кафедри факультету інформатики або факультету систем автоматизованого проектування, а також деканати цих факультетів.
67. Отримати список лекцій, на яких кількість студентів у групі перевищує кількість місць в аудиторії. У списку вказати номер аудиторії, номер групи, дисципліну, що читається, тиждень та день тижня.

68. Хто з викладачів кафедри АСУ не читає лекцій?
69. Отримати номери тих викладачів, які викладають лише в усіх групах першого курсу.
70. Отримати список тих предметів, які викладаються у всіх аудиторіях.
71. Отримати список тих предметів, які викладаються на всіх факультетах.
72. Отримати список тих предметів, які викладаються для студентів всіх кафедр.
73. Отримати список груп, яким викладають всі викладачі кафедри 'Кафедра математичного аналізу'.
74. Отримати список предметів, які викладаються у всіх групах 1 і 2 року навчання.
75. Отримати імена викладачів, які читають лекції на всіх факультетах.
76. Отримати імена викладачів, які читають лекції у всіх аудиторіях 1 корпусу університету.

Задача2.

Використовуючи властивості операцій реляційної алгебри, провести оптимізаційні перетворення та побудувати схему виконання вихідного запиту і схеми кроків оптимізації наступних виразів:

1. $\pi_{A,C,D}(\pi_{A,C,D,F}(R(A, B,C)\times S(D, E, F)))$;
2. $\pi_{F,A,D}$ $(\pi_{A,C,D,F}$ $(R(A, B,C)\times S(D, E, F)))$;
3. $\pi_{A,D,E,F}$ $(\pi_{A,C,D,E,F}$ $(R(A, B,C)\times S(D, E, F)))$;
4. $\pi_{A,D,E}$ $(\pi_{A,C,D,E,F}$ $(R(A, B,C)\times S(D, E, F)))$;
5. $\pi_{A,D,E,F}(\pi_{A,C,D,E,F}$ $(R(A, B,C)\times S(D, E, F)))$;
6. $\sigma_{D=9\&E=1}(\pi_{A,D,E}$ $(R(A, B,C)$ $\times S(D, E, F)))$;
7. $\sigma_{A=9\&C=1}(\pi_{A,C,D}$ $(R(A, B,C)$ $\times S(D, E, F)))$;
8. $\sigma_{D=9\&A=1}(\pi_{A,D}(R(A, B,C)\times S(D, E, F)))$;
9. $\sigma_{D=9\&A=1}(\pi_{F,D,A}$ $(R(A, B,C)\times S(D, E, F)))$;
10. $\sigma_{D=9\&A=1\&E='f'}$ $(\pi_{A,D,E,F}$ $(R(A, B,C)\times S(D, E, F)))$;
11. $\pi_{F,F}(\sigma_{A=1\&E='f'}$ $(R(A, B,C)\times S(D, E, F)))$;
12. $\pi_{A,D}(\sigma_{D=9\&E=1}(R(A, B,C)$ $\times S(D, E, F)))$;
13. $\pi_{C,D}(\sigma_{A=9\&C=1}(R(A, B,C)$ $\times S(D, E, F)))$;
14. $\pi_{A}(\sigma_{D=9\&A=1}(R(A, B,C)\times S(D, E, F)))$;
15. $\pi_{F}(\sigma_{D=9\&A=1}(R(A, B,C)\times S(D, E, F)))$;
16. $\pi_{F}(\sigma_{D=9\&A=1\&E='f'}$ $(R(A, B,C)\times S(D, E, F)))$;
17. $\pi_{F}(\sigma_{A=1\&E='f'}$ $(R(A, B,C)\times S(D, E, F)))$;
18. $\pi_{F}(\sigma_{F=1\&E='f'}$ $(R(A, B,C)\times S(D, E, F)))$;
19. $\pi_{A}(\sigma_{C=D\&D=9\&A=1}(R(A, B,C)\times S(D, E, F)))$;
20. $\pi_{F}(\sigma_{A=D\&D=9\&A=1}(R(A, B,C)\times S(D, E, F)))$;
21. $\pi_{F}(\sigma_{A=D\&D=9\&A=1\&E='f'}$ $(R(A, B,C)\times S(D, E, F)))$;
22. $\pi_{F}(\sigma_{A=D\&A=1\&E='f'}$ $(R(A, B,C)\times S(D, E, F)))$;
23. $\pi_{F}(\sigma_{A=D\&F=1\&E='f'}$ $(R(A, B,C)\times S(D, E, F)))$;
24. $\pi_{A,C,D}(\sigma_{C=D\&D=9\&A=1}(R(A, B,C)\times S(D, E, F)))$;

25. $\pi_{F,A,D}\ (\sigma_{A=D\&D=9\&A=1}(R(A,\ B,C)\times S(D,\ E,\ F)))$;
26. $\pi_{\ A,D,E,F}\ (\sigma_{A=D\&D=9\&A=1\&E='f'}\ (R(A,\ B,C)\times S(D,\ E,\ F)))$;
27. $\pi_{\ A,D,E}\ (\sigma_{A=D\&A=1\&E='f'}\ (R(A,\ B,C)\times S(D,\ E,\ F)))$;
28. $\pi_{\ A,D,E,F}(\sigma_{A=D\&F=1\&E='f'}\ (R(A,\ B,C)\times S(D,\ E,\ F)))$;
29. $\pi_{A,C,D}(\sigma_{C=D}\ (\sigma_{D=9}(\sigma_{A=1}(R(A,\ B,C)\times S(D,\ E,\ F)))))$;
30. $\pi_{F,A,D}\ (\sigma_{A=D}(\sigma_{D=9}(\sigma_{A=1}(R(A,\ B,C)\times S(D,\ E,\ F)))))$;
31. $\pi_{\ A,D,E,F}\ (\sigma_{A=D}(\sigma_{D=9}(\sigma_{A=1}(\sigma_{E='f'}\ (R(A,\ B,C)\times S(D,\ E,\ F))))))$;
32. $\pi_{\ A,D,E}\ (\sigma_{A=D}(\sigma_{A=1}(\sigma_{E='f'}\ (R(A,\ B,C)\times S(D,\ E,\ F)))))$;
33. $\pi_{\ A,D,E,F}(\sigma_{A=D}(\sigma_{F=1}(\sigma_{E='f'}\ (R(A,\ B,C)\times S(D,\ E,\ F)))))$;
34. $\pi_{A,C,D}(\sigma_{C=D}\ (\sigma_{D=9}(\sigma_{A=1\&\ D=5}\ (R(A,\ B,C)\times S(D,\ E,\ F)))))$;
35. $\pi_{F,A,D}\ (\sigma_{A=D}(\sigma_{D=9}(\sigma_{A=1\&\ E=5}\ (R(A,\ B,C)\times S(D,\ E,\ F)))))$;
36. $\pi_{\ A,D,E,F}\ (\sigma_{A=D}(\sigma_{D=9}(\sigma_{A=1\&\ F=5}\ (\sigma_{E='f'}\ (R(A,\ B,C)\times S(D,\ E,\ F))))))$;
37. $\pi_{\ A,D,E}\ (\sigma_{A=D}(\sigma_{A=1\&\ F=5}\ (\sigma_{E='f'}\ (R(A,\ B,C)\times S(D,\ E,\ F)))))$;
38. $\pi_{\ A,D,E,F}(\sigma_{A=D\&\ F=5}\ (\sigma_{E='f'}\ (R(A,\ B,C)\times S(D,\ E,\ F))))$;

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пасічник В.В., Резніченко В.А. Організація баз даних та знань. - К. : Видавнича група BHV, 2006. - 384 с.
2. Советов, Б. Я. Базы данных : учебник для прикладного бакалавриата / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — Москва : Издательство Юрайт, 2015. — 463 с. [Електронний ресурс] — Режим доступу : https://urait.ru/bcode/382832.
3. Бураков П.В., Петров В.Ю. Введение в системы баз данных: учебное пособие. — СПб: СПбГУ ИТМО, 2010. — 128 с. [Електронний ресурс] — Режим доступу : http://window.edu.ru/catalog/pdf2txt/433/70433/45697?p_page=10
4. Реляційні бази даних: табличні алгебри та SQL-подібні мови / В. Н. Редько, Ю. Й. Брона, Д. Б. Буй, С.А. Поляков. — К. : Видавничий дім “Академперіодика”, 2001. — 198 с.
5. Дунаев В. В. Базы даннях. Язык SQL / В. В. Дунаев. — СПб. : БХВ-Петербург, 2006. — 288 с.
6. Пушников А.Ю. Введение в системы управления базами данных. Часть 1. Реляционная модель данных: Учебное пособие/Изд-е Башкирского ун-та. - Уфа, 1999. - 108 с. - ISBN 5-7477-0350-1. численням [Електронний ресурс] — Режим доступу : http://citforum.ru/database/dblearn/index.shtml
7. Дейт, К. Дж. Введение в системы баз данных, 8-е издание. : Пер. с англ. - М. : Издательский дом «Вильямс», 2005. – 1328 с.
8. Реляційне числення [Електронний ресурс] — Режим доступу : https://studfile.net/preview/5993100/page:4/
9. Решение задач в реляционной алгебре [Електронний ресурс] — Режим доступу : https://studopedia.org/13-27483.html
10. Різниця між реляційною алгеброю та реляційним численням [Електронний ресурс] — Режим доступу : https://uk.gadget-info.com/difference-between-relational-algebra
11. Куценко И.Л. Оптимизация запросов в реляционных базах данных / И.Л. Куценко // Евразийский Союз Ученых. – Т.56, №11. –2018. С. 52–58 [Електронний ресурс] — Режим доступу : https://euroasia-science.ru/
12. Отображение реляционной алгебры в дескриптивную логику / И.С. Чистякова // Проблеми програмування. — 2018. — № 2-3. — С. 214-225. — Бібліогр.: 12 назв. — рос. [Електронний ресурс] — Режим доступу : http://dspace.nbuv.gov.ua/bitstream/handle/123456789/144602/25-Chystiakova.pdf?sequence=1

ЗМІСТ

УДК 004.655

Електронне мережне навчальне видання видання

Л. В. Булатецька, В. В. Булатецький

РЕЛЯЦІНА АЛГЕБРА.
РЕЛЯЦІЙНЕ ЧИСЛЕННЯ

Методичні вказівки для підготовки до контрольної роботи з нормативних навчальних дисциплін
“Бази даних та розподілені інформаційно-аналітичні системи”
“Організація баз даних та знань”