# Лабораторна робота № 13

# З'єднання таблиць (робота з записами таблиць, як з множинами)

## UNION

Оператор UNION подібно inner join або outer join дозволяє з'єднати дві таблиці. Але на відміну від inner / outer join об'єднання єднають не стовпці різних таблиць, а два однотипних набору в один. Формальний синтаксис об'єднання:

|  |
| --- |
| SELECT\_вираз1  UNION [ALL] SELECT\_вираз2  [UNION [ALL] SELECT\_виразN] |

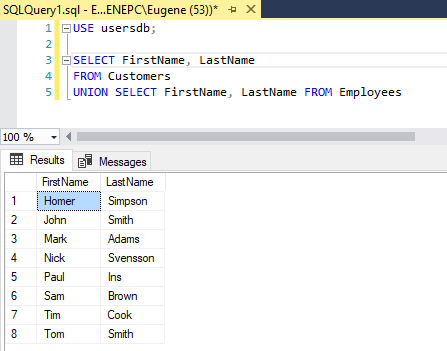
Наприклад, нехай в базі даних будуть дві окремі таблиці для клієнтів банку (таблиця Customers) і для співробітників банку (таблиця Employees):

|  |
| --- |
| USE usersdb;  CREATE TABLE Customers  (  Id INT IDENTITY PRIMARY KEY,  FirstName NVARCHAR (20) NOT NULL,  LastName NVARCHAR (20) NOT NULL,  AccountSum MONEY  );  CREATE TABLE Employees  (  Id INT IDENTITY PRIMARY KEY,  FirstName NVARCHAR (20) NOT NULL,  LastName NVARCHAR (20) NOT NULL,  );  INSERT INTO Customers VALUES  ( 'Tom', 'Smith', 2000),  ( 'Sam', 'Brown', 3000),  ( 'Mark', 'Adams', 2500),  ( 'Paul', 'Ins', 4200),  ( 'John', 'Smith', 2800),  ( 'Tim', 'Cook', 2800)  INSERT INTO Employees VALUES  ( 'Homer', 'Simpson'),  ( 'Tom', 'Smith'),  ( 'Mark', 'Adams'),  ( 'Nick', 'Svensson') |

Тут ми можемо помітити, що обидві таблиці, незважаючи на наявність різних даних, можуть характеризуватися двома загальними атрибутами - ім'ям (FirstName) і прізвищем (LastName). Виберемо відразу всіх клієнтів банку і його співробітників з обох таблиць:

|  |  |
| --- | --- |
| 1  2  3 | SELECT FirstName, LastName  FROM Customers  UNION SELECT FirstName, LastName FROM Employees |

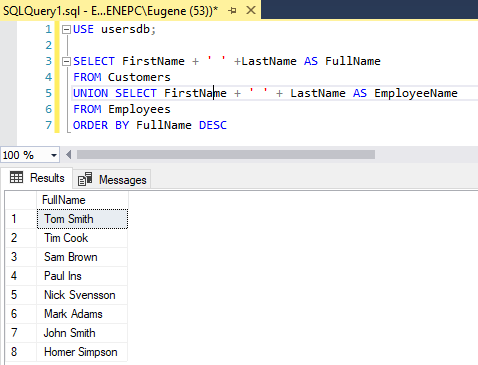
В даному випадку з першої таблиці вибираються два значення - ім'я та прізвище клієнта. З другої таблиці Employees також вибираються два значення - ім'я та прізвище співробітників. Тобто при об'єднанні кількість обираних стовпців і їх тип збігаються для обох вибірок.



При цьому назви стовпців об'єднаної вибірки будуть збігатися з назви стовпців першої вибірки. І якщо ми захочемо при цьому ще зробити сортування, то у виразах ORDER BY необхідно орієнтуватися саме на назви стовпців першої вибірки:

|  |
| --- |
| SELECT FirstName + '' + LastName AS FullName  FROM Customers  UNION SELECT FirstName + '' + LastName AS EmployeeName  FROM Employees  ORDER BY FullName DESC |

В даному випадку кожна вибірка має по одному стовпцю, який представляє об'єднання імені та прізвища клієнта або співробітника. Але у випадку з клієнтами стовпець буде називатися FullName, а у випадку з співробітниками - EmployeeName. Проте для сортування застосовується назва стовпчика з першої вибірки і він же буде в результуючої вибіркою:



Якщо ж в одній вибірці більше стовпців, ніж в інший, то вони не зможуть бути об'єднані. Наприклад, в наступному випадку об'єднання завершиться з помилкою:

|  |
| --- |
| SELECT FirstName, LastName, AccountSum  FROM Customers  UNION SELECT FirstName, LastName  FROM Employees |

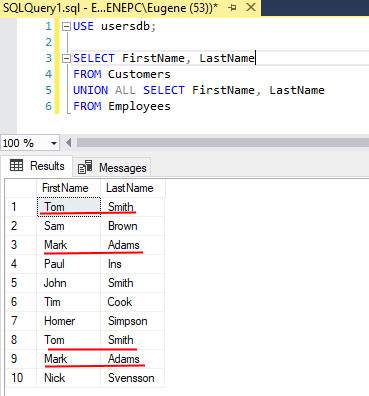
Також відповідні стовпці повинні відповідати за типом. Так, наступний приклад завершиться з помилкою через невідповідності по типу даних:

|  |
| --- |
| SELECT FirstName, LastName  FROM Customers  UNION SELECT Id, LastName  FROM Employees |

В даному випадку перший стовпець першої вибірки має тип NVARCHAR, тобто зберігає рядок. Перший стовпець другої вибірки - Id має тип INT, тобто зберігає число.

Якщо обидва об'єднуються набору містять в рядках ідентичні значення, то при об'єднанні повторювані рядки видаляються. Наприклад, у випадку з таблицями Customers і Employees співробітники банку можуть бути одночасно його клієнтами і міститися в обох таблицях. При об'єднанні в прикладах вище всіх дублюються рядки віддалялися. Якщо ж необхідно при об'єднанні зберегти все, в тому числі повторювані рядки, то для цього необхідно використовувати оператор ALL:

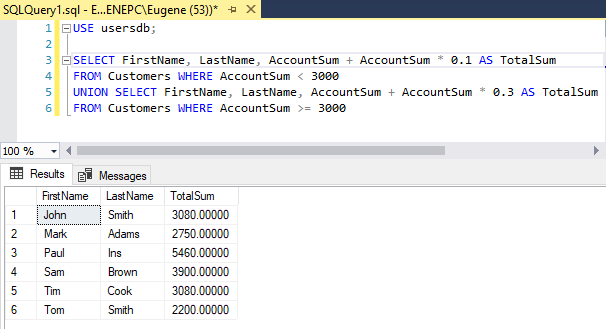
|  |
| --- |
| SELECT FirstName, LastName  FROM Customers  UNION ALL SELECT FirstName, LastName  FROM Employees |



Об'єднувати вибірки можна і з однієї і тієї ж таблиці. Наприклад, в залежності від суми на рахунку клієнта нам треба нараховувати йому певні відсотки:

|  |
| --- |
| SELECT FirstName, LastName, AccountSum + AccountSum\*0.1 AS TotalSum  FROM Customers WHERE AccountSum <3000  UNION SELECT FirstName, LastName, AccountSum+AccountSum\*0.3 AS TotalSum  FROM Customers WHERE AccountSum> = 3000 |

В даному випадку якщо сума менше 3000, то нараховуються відсотки в розмірі 10% від суми на рахунку. Якщо на рахунку більше 3000, то відсотки збільшуються до 30%.



## EXCEPT

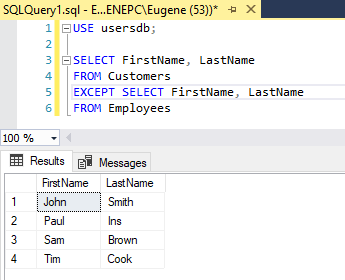
Оператор EXCEPT дозволяє знайти різницю двох вибірок, тобто ті рядки які є в першій вибірці, але яких немає в другій. Для його використання застосовується наступний формальний синтаксис:

|  |
| --- |
| SELECT\_вираз1  EXCEPT SELECT\_вираз2 |

Візьмемо таблиці Employees і Customers з минулого підрозділу.

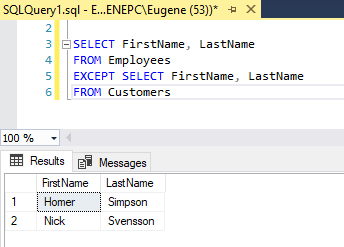
Таблиця Employees містить дані про всіх співробітників банку, а таблиця Customers - про всіх клієнтів. Але співробітники банку можуть також бути його клієнтами. Нам треба знайти всіх клієнтів банку, які не є його співробітниками:

|  |
| --- |
| SELECT FirstName, LastName  FROM Customers  EXCEPT SELECT FirstName, LastName  FROM Employees |



Подібним чином можна отримати всіх співробітників банку, які не є його клієнтами:

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT FirstName, LastName  FROM Employees  EXCEPT SELECT FirstName, LastName  FROM Customers |



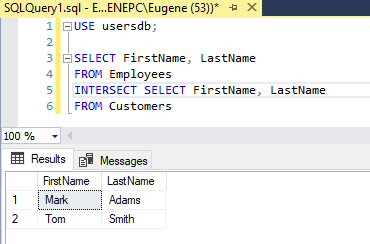
## INTERSECT

Оператор INTERSECT дозволяє знайти загальні рядки для двох вибірок, тобто даний оператор виконує операцію перетину множин. Для його використання застосовується наступний формальний синтаксис:

|  |  |
| --- | --- |
| 1  2 | SELECT\_вираженіе1  INTERSECT SELECT\_вираженіе2 |

У таблиці Customers зберігаються всі клієнти банку, а в таблиці Employees - все його співробітники. Але співробітники можуть бути одночасно і клієнтами банку, тому їх дані можуть зберігатися відразу в двох таблицях. Знайдемо всіх співробітників банку, які одночасно є його клієнтами. Тобто нам треба знайти спільні елементи двох вибірок:

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT FirstName, LastName  FROM Employees  INTERSECT SELECT FirstName, LastName  FROM Customers |



# Завдання

1. З використанням UINION, EXCEPT або INTERSECT треба та без використання ROLLUP додати підсумовуючий рядок для таблиці з максимальним, мінімальним та середнім значенням ціни товару для кожного виробника.
2. З використанням UINION, EXCEPT або INTERSECT треба показати замовників і кількість їх замовлень у минулому і позаминулому місяцях, хто зробив більше 2 замовлень у минулому місяці та більше 5 замовлень у позаминулому місяці (замовлення в окремих записах по місяцям).
3. З використанням UINION, EXCEPT або INTERSECT треба показати тих замовників і кількість їх замовлень у минулому місяці, хто зробив більше 2 замовлень у минулому місяці, але не робив замовлень у позаминулому (замовлення в окремих записах по місяцям)
4. З використанням UINION, EXCEPT або INTERSECT вивести товари і кількість їх замовлень у березнях кожного року, які замовлялись більше 5 разів у березнях кожного року та більше 10 разів за рік (замовлення в окремих записах по місяцям)
5. Створити в своїй базі таблицю (Representative\_office) з компаніями, які містять представництва в Україні (id, Name\_Company) та знайти серед них ті компанії, продукцію яких замовляли у жовтні цього року (УВАГА!!! Дана таблиця не має зв’язків з іншими таблицями).
6. Знайти продані в жовтні товари, фірми-виробники яких не мають представництва в Україні.