

MySQL

Заняття #5

- Неявне з'єднання таблиць
- Явне з'єднання таблиць



Неявне з'єднання таблиць

Нерідко виникає необхідність в одному запиті отримати дані відразу з декількох таблиць. Для зведення даних з різних таблиць ми можемо використовувати різні способи.

З'єднаємо дві таблиці Orders і Customers:

```
SELECT * FROM Orders, Customers;
```

При такій вибірці кожен рядок з таблиці Orders буде з'єднуватися з кожним рядком з таблиці Customers. Тобто, вийде перехресне з'єднання. Наприклад, в Orders три рядки, а в Customers той же три рядки, значить ми отримаємо $3 * 3 = 9$ рядків.

Але навряд чи це той результат, який хотілося б бачити. Тим більше кожне замовлення з Orders пов'язаний з конкретним покупцем з Customers, а не з усіма можливими покупцями.

Щоб вирішити задачу, необхідно використовувати вираз WHERE і фільтрувати рядки за умови, що поле CustomerId з Orders відповідає полю Id з Customers, це і є неявне з'єднання таблиць:

```
SELECT * FROM Orders, Customers WHERE Orders.CustomerId = Customers.Id;
```

Неявне з'єднання таблиць

Тепер об'єднаємо дані по трьом таблиць Orders, Customers і Products. Тобто отримаємо всі замовлення і додамо інформацію по клієнту і пов'язаного товару:

```
SELECT Customers.FirstName, Products.ProductName, Orders.CreatedAt  
FROM Orders, Customers, Products  
WHERE Orders.CustomerId = Customers.Id AND Orders.ProductId = Products.Id;
```

Так як тут потрібно з'єднати три таблиці, то застосовуються як мінімум дві умови. Ключовий таблицею залишається Orders, з якої витягуються всі замовлення, а потім до неї приєднується дані по клієнту за умовою Orders.CustomerId = Customers.Id і дані по товару за умовою Orders.ProductId = Products.Id.

В даному випадку назви таблиць сильно збільшують код, але ми його можемо скоротити за рахунок використання псевдонімів таблиць:

```
SELECT C.FirstName, P.ProductName, O.CreatedAt  
FROM Orders AS O, Customers AS C, Products AS P  
WHERE O.CustomerId = C.Id AND O.ProductId = P.Id;
```

Явне з'єднання таблиць

Як правило, більш поширений підхід з'єднання даних з різних таблиць представляє застосування оператора **JOIN**. Загальний формальний синтаксис застосування оператора **INNER JOIN**:

```
SELECT стовпці  
FROM таблиця1  
  [INNER] JOIN таблиця2  
    ON умова1  
  [[INNER] JOIN таблиця3  
    ON умова2]
```

Після оператора **JOIN** йде назва другої таблиці, з якої треба додати дані до вибірки. Перед **JOIN** може використовуватися необов'язкове ключове слово **INNER**. Його наявність або відсутність ні на що не впливає. Потім після ключового слова **ON** вказується умова з'єднання. Ця умова встановлює, як дві таблиці будуть порівнюватися. У більшості випадків для з'єднання застосовується первинний ключ головної таблиці і зовнішній ключ залежною таблицею.

Явне з'єднання таблиць (INNER JOIN)

Використовуючи **JOIN**, виберемо всі замовлення і додамо до них інформацію про товари:

```
SELECT Orders.CreatedAt, Orders.ProductCount, Products.ProductName  
FROM Orders JOIN Products ON Products.Id = Orders.ProductId;
```

Оскільки таблиці можуть містити стовпці з однаковими назвами, то при вказівці стовпців для вибірки вказується їх повне ім'я разом з ім'ям таблиці, наприклад, "**Orders.ProductCount**".

Використовуючи псевдоніми для таблиць, можна скоротити код:

```
SELECT O.CreatedAt, O.ProductCount, P.ProductName  
FROM Orders AS O JOIN Products AS P ON P.Id = O.ProductId
```

Також можна приєднувати дані відразу з декількох таблиць. Наприклад, додамо до замовлення інформацію про покупця з таблиці Customers:

```
SELECT Orders.CreatedAt, Customers.FirstName, Products.ProductName  
FROM Orders  
JOIN Products ON Products.Id = Orders.ProductId  
JOIN Customers ON Customers.Id = Orders.CustomerId;
```

Явне з'єднання таблиць (OUTER JOIN)

У попередній темі розглядали **INNER JOIN** або внутрішнє з'єднання таблиць. Але також в MySQL ми можемо використовувати і так зване зовнішнє з'єднання або **OUTER JOIN**. На відміну від **INNER JOIN** зовнішнє з'єднання повертає всі рядки однієї або двох таблиць, які беруть участь в з'єднанні.

Outer Join має наступний формальний синтаксис:

```
SELECT стовпці  
FROM таблиця1  
    {LEFT | RIGHT} [OUTER] JOIN таблиця2 ON умова1  
    [{LEFT | RIGHT} [OUTER] JOIN таблиця3 ON умова2] ...
```

Перед оператором **JOIN** вказується одне з ключових слів **LEFT** або **RIGHT**, які визначають тип з'єднання:

- **LEFT**: вибірка буде містити всі рядки з першої або лівої таблиці
- **RIGHT**: вибірка буде містити всі рядки з другої або правої таблиці

Також перед оператором **JOIN** може вказуватися ключове слово **OUTER**, але його застосування необов'язково. Далі після **JOIN** вказується приєднується таблиця, а потім йде умова з'єднання.

Явне з'єднання таблиць (OUTER JOIN)

У попередній темі розглядали **INNER JOIN** або внутрішнє з'єднання таблиць. Але також в MySQL ми можемо використовувати і так зване зовнішнє з'єднання або **OUTER JOIN**. На відміну від **INNER JOIN** зовнішнє з'єднання повертає всі рядки однієї або двох таблиць, які беруть участь в з'єднанні.

Наприклад, з'єднаємо таблиці **Orders** і **Customers**:

```
SELECT FirstName, CreatedAt, ProductCount, Price, ProductId  
FROM Orders LEFT JOIN Customers  
ON Orders.CustomerId = Customers.Id
```

Таблиця **Orders** є першою або лівою таблицею, а таблиця **Customers** - правою таблицею. Тому, так як тут використовується вибірка по лівій таблиці, то спочатку будуть вибиратися всі рядки з **Orders**, а потім до них за умовою **Orders.CustomerId = Customers.Id** будуть додаватися пов'язані рядки з **Customers**.

За вищенаведеним результатом може здатися, що лівостороннє з'єднання аналогічно **INNER JOIN**, але це не так. **INNER JOIN** об'єднує рядки з двох таблиць при відповідності умові. Якщо одна з таблиць містить рядки, які не відповідають цій умові, то дані рядки не включаються у вихідну вибірку. **LEFT JOIN** вибирає всі рядки першої таблиці і потім приєднує до них рядки правої таблиці.

Явне з'єднання таблиць (OUTER JOIN)

У попередній темі розглядали **INNER JOIN** або внутрішнє з'єднання таблиць. Але також в MySQL ми можемо використовувати і так зване зовнішнє з'єднання або **OUTER JOIN**. На відміну від **INNER JOIN** зовнішнє з'єднання повертає всі рядки однієї або двох таблиць, які беруть участь в з'єднанні.

Наприклад, візьмемо таблицю Customers і додамо до покупців інформацію про їх замовленнях:

```
SELECT FirstName, CreatedAt, ProductCount, Price  
FROM Customers LEFT JOIN Orders ON Orders.CustomerId = Customers.Id;
```

У випадку з **LEFT JOIN** MySQL вибирає спочатку всіх покупців з таблиці **Customers**, потім зіставляє їх із замовленнями з таблиці **Orders** через умову **Orders.CustomerId = Customers.Id**. Однак не у всіх покупців є замовлення. У цьому випадку покупцеві для відповідних стовпців встановлюються значення **NULL**. Змінимо в прикладі вище тип з'єднання для **OUTER JOIN** з лівостороннього на правосторонній:

```
SELECT FirstName, CreatedAt, ProductCount, Price  
FROM Customers RIGHT JOIN Orders ON Orders.CustomerId = Customers.Id;
```

Тепер будуть вибиратися всі рядки з Orders (з правої таблиці), а до них вже буде приєднуватися пов'язані з умовою рядки з таблиці Customers.

Додаткові джерела інформації:

Посилання

https://www.w3schools.com/sql/sql_join.asp

https://www.w3schools.com/sql/sql_join_inner.asp

https://www.w3schools.com/sql/sql_join_left.asp

https://www.w3schools.com/sql/sql_join_right.asp

https://www.w3schools.com/sql/sql_examples.asp