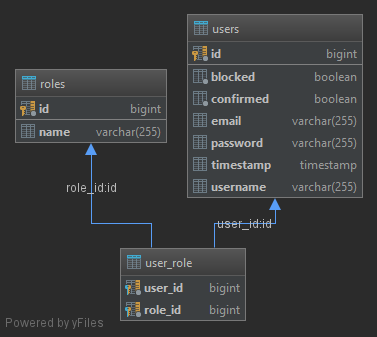
Яцкевич Фёдор, 522403

**Отчет по работе сайта на СУБД Postgresql**

Был произведен переход сайта с СУБД MySQL на PostgreSQl. В ходе перехода серьезных изменений совершено не было, т.к.:

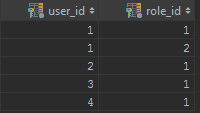
1. Обе СУБД имеют схожий синтаксис запросов
2. В данном приложении используется ORM Framework Hibernate (Spring Data JPA), позволяющий создавать универсальные запросы на языке HQL.
3. **Сложные группировки (GROUPING SETS, ROLLUP, CUBE)**

Рассмотрим таблицу user\_role



Для начала напишем запрос по получению из данной таблицы строки id роли и количество пользователей с данной ролью

Таблица содержит следующие данные:



Запрос:

**SELECT** role\_id, *COUNT*(\*)  
**FROM** user\_role  
**GROUP BY** role\_id;

Response:



Как видно суть оператора GROUP BY в том, что наборы строк с одинаковыми значениями преобразуются в одну строку, имеет смысл это только при использовании агрегатных функций примененных к этим группам.

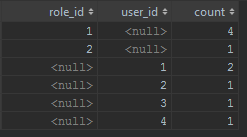
GROUPING SETS

Теперь изменим запрос так чтобы он выводил сначала список ролей с количеством пользователей имеющих данную роль, а затем список пользователей с количеством ролей у данного пользователя:

Запрос:

**SELECT** role\_id, user\_id, *COUNT*(\*)  
**FROM** user\_role  
**GROUP BY GROUPING SETS** (role\_id, user\_id);

Ответ:



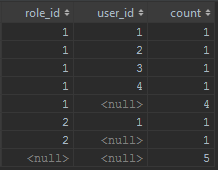
Как видно, для этого выл использован оператор GROUPING SETS в совокупности с GROUP BY.

Его использование похоже на последовательный вызов только оператора GROUP BY сначала с одним полем, а затем с другим.

ROLLUP

Выполним оператор ROLLUP:

**SELECT** role\_id, user\_id, *COUNT*(\*)  
**FROM** user\_role  
**GROUP BY ROLLUP** (role\_id, user\_id);



Как видно данный запрос равносилен запросу:

**SELECT** role\_id, user\_id, *COUNT*(\*)  
**FROM** user\_role  
**GROUP BY GROUPING SETS** ((role\_id, user\_id), (role\_id), ());

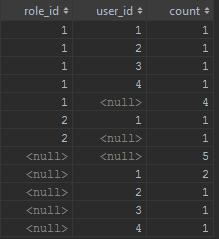
Данный оператор применим для обработки иерархических структур данных. Например подсчитать зарплату для подразделения, отдела, и т.д.

CUBE

Теперь выполним запрос с оператором CUBE:

**SELECT** role\_id, user\_id, *COUNT*(\*)  
**FROM** user\_role  
**GROUP BY CUBE** (role\_id, user\_id);

Получим ответ:



Данный запрос равносилен запросу:

**SELECT** role\_id, user\_id, *COUNT*(\*)  
**FROM** user\_role  
**GROUP BY GROUPING SETS** ((role\_id, user\_id), (role\_id), (user\_id), ());

Он создает группировки по заданному списку и всем его подмножествам.

1. **Оконные функции**

Оконная функция выполняет вычисления для набора строк, некоторым образом связанных с текущей строкой. Можно сравнить её с агрегатной функцией, но, в отличие от обычной агрегатной функции, при использовании оконной функции несколько строк не группируются в одну, а продолжают существовать отдельно. Внутри же, оконная функция, как и агрегатная, может обращаться не только к текущей строке результата запроса.