# ЗАДАНИЕ SQL

**Выполнил:** Щурик Валерий

Группа: 1\_JavaST\_2018\_Morning\_Blinov

Вариант 19: Система «Аукцион цветов»

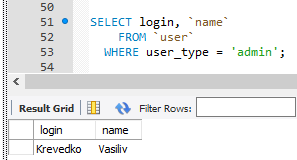
## 4). Запросы с WHERE-условиями

***4.1 Задача:*** *Показать всех администраторов в системе*

SELECT login, `name`

FROM `user`

WHERE user\_type = 'admin';

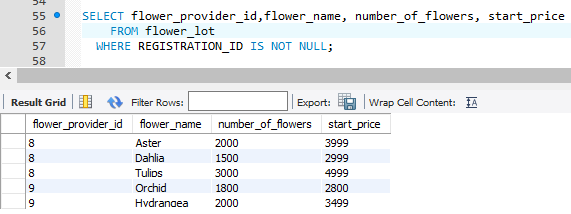


***4.2 Задача:*** *Показать все зарегистрированные лоты*

SELECT flower\_provider\_id,flower\_name, number\_of\_flowers, start\_price

FROM flower\_lot

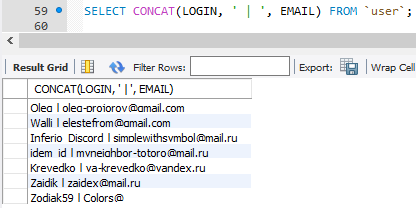
WHERE REGISTRATION\_ID IS NOT NULL;



## 5). Запросы с применением *встроенных* (кроме агрегатных) функций

***5.1 Задача:*** *Вывести всех пользователей в виде «логин | почтовый адрес»*

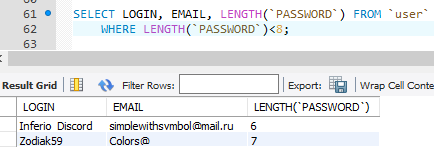
SELECT CONCAT(LOGIN, ' | ', EMAIL) FROM `user`;



***5.2 Задача:*** *Показать всех пользователей с ненадёжным паролем (менее 8 символов), чтобы порекомендовать им сменить пароль*

SELECT LOGIN, EMAIL, LENGTH(`PASSWORD`) FROM `user`

WHERE LENGTH(`PASSWORD`)<8;

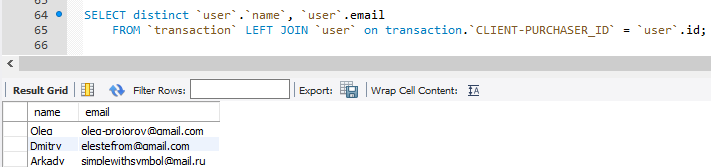


## 6). Запросы на соединения таблиц

***6.1 Задача:*** *Показать всех пользователей, кто хоть раз купил какой-либо лот.*

SELECT DISTINCT `user`.`name`, `user`.email, transaction.value

FROM `transaction` LEFT JOIN `user` on transaction.`CLIENT-PURCHASER\_ID` = `user`.id;

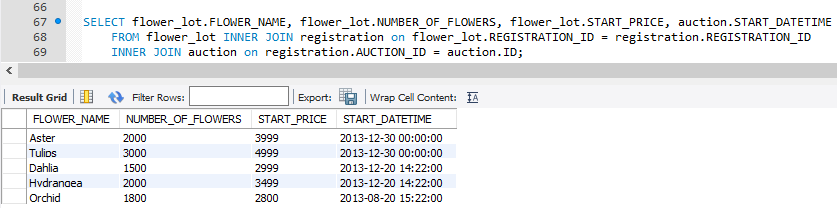


***6.2 Задача:*** *Показать все цветочные лоты, что участвовали или будут участвовать в аукционах, а также стартовую цену и дату аукциона, на котором предполагается продажа.*

SELECT flower\_lot.FLOWER\_NAME, flower\_lot.NUMBER\_OF\_FLOWERS, flower\_lot.START\_PRICE, auction.START\_DATETIME

FROM flower\_lot INNER JOIN registration on flower\_lot.REGISTRATION\_ID = registration.REGISTRATION\_ID

INNER JOIN auction on registration.AUCTION\_ID = auction.ID;



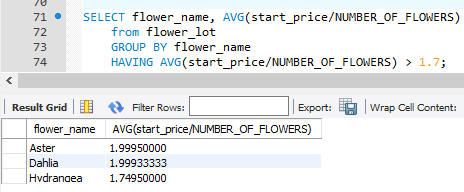
## 7). Запрос c GROUP BY и HAVING и агрегатными функциями

***7.1 Задача:*** *Показать все дорогие цветы (у которых среднее значение стартовой цены, деленное на количество цветов в лоте больше 1.7)*

SELECT flower\_name, AVG(start\_price/NUMBER\_OF\_FLOWERS)

from flower\_lot

GROUP BY flower\_name

HAVING AVG(start\_price/NUMBER\_OF\_FLOWERS) > 1.7; 

## 8). Запрос с UNION

***8.1 Задача:*** *Показать всех «активных» пользователей. Если это клиент-покупатель, то он должен хоть раз купить какой-либо лот; если это поставщик, то хоть раз предложить какой-либо лот на продажу*

SELECT `user`.id, `user`.`name`, `user`.email

FROM `user`

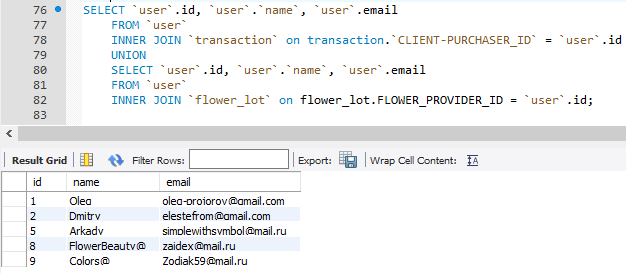
INNER JOIN `transaction` on transaction.`CLIENT-PURCHASER\_ID` = `user`.id

UNION

SELECT `user`.id, `user`.`name`, `user`.email

FROM `user`

INNER JOIN `flower\_lot` on flower\_lot.FLOWER\_PROVIDER\_ID = `user`.id

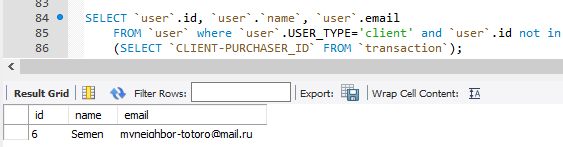


## 9). Запросы с подзапросами

***9.1 (невзаимосвязанный) Задача:*** *Показать всех клиентов, кто за всё время существования системы ничего не купил*

SELECT `user`.id, `user`.`name`, `user`.email

FROM `user` where `user`.USER\_TYPE='client' and `user`.id not in (SELECT `CLIENT-PURCHASER\_ID` FROM `transaction`);



***9.2 (взаимосвязанный) Задача:*** *Выбрать все крупные поставки цветов (больше, чем средняя для данного поставщика).*

SELECT\*

from flower\_lot fl1 where NUMBER\_OF\_FLOWERS >

(SELECT AVG(NUMBER\_OF\_FLOWERS)

from flower\_lot fl2 where

fl1.FLOWER\_PROVIDER\_ID=fl2.FLOWER\_PROVIDER\_ID);

