**Выполнил**: Моисеенко Евгений Александрович

**Группа**: Java\_12

**Тема**: система «Отель»

**4)Запросы с WHERE-условиями**

*4.1 Задача: Показать количество и общую сумму заказов по каждому клиенту для указанной учетной записи.*

**SELECT** `client`.`first\_name`,`client`.`last\_name`, `client`.`nationality\_id`,

**COUNT**(`client`.`id`) **AS** **'quantity'**, **SUM**(`cost\_order`.`cost`) **AS** **'total'**

**FROM** `account` **LEFT JOIN**

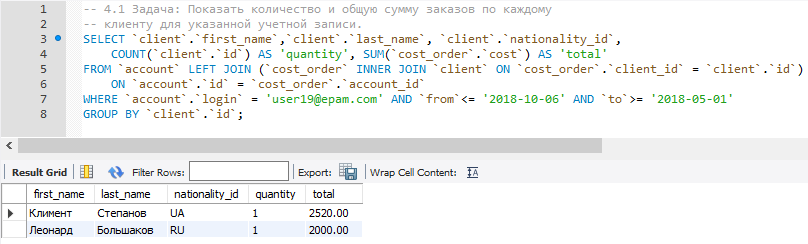
**(**`cost\_order` **INNER JOIN** `client` **ON** `cost\_order`.`client\_id` = `client`.`id`**)**

**ON** `account`.`id` = `cost\_order`.`account\_id`

**WHERE** `account`.`login` **=** 'user19@epam.com' **AND** `from`**<=** '2018-10-06' **AND**

`to`**>=** '2018-05-01'

**GROUP BY** `client`.`id`;



*4.2 Задача: Показать клиентов, являющихся гражданами Украины и Республики Беларусь.*

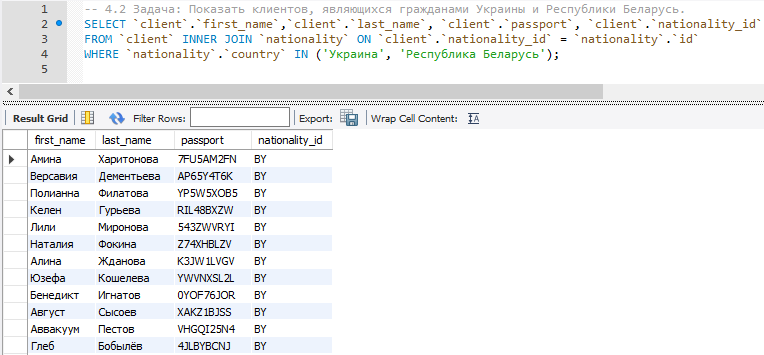
**SELECT** `client`.`first\_name`,`client`.`last\_name`, `client`.`passport`,

`client`.`nationality\_id`

**FROM** `client` **INNER JOIN** `nationality`

**ON** `client`.`nationality\_id` **=** `nationality`.`id`

**WHERE** `nationality`.`country` **IN** **(**'Украина', 'Республика Беларусь'**)**;

**

**5)Запросы с применением встроенных функций**

*5.1 Задача: Показать стоимость каждого заказа.*

-- 5.1 Задача: Показать стоимость каждого заказа.

**CREATE**

**ALGORITHM = UNDEFINED**

**DEFINER =** `root`@`localhost`

**SQL SECURITY DEFINER**

**VIEW** **`cost\_order`** **AS**

(**SELECT** `order`.`id` **AS** `id`, `order`.`room\_number` **AS** `room\_number`,

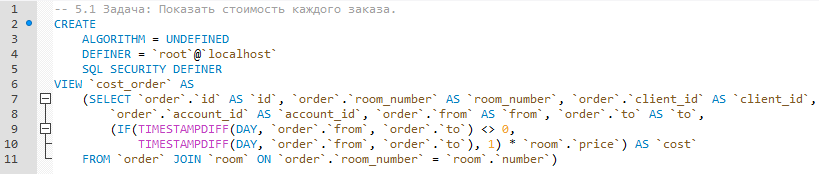
`order`.`client\_id` **AS** `client\_id`,`order`.`account\_id` **AS** `account\_id`,

`order`.`from` **AS** `from`, `order`.`to` **AS** `to`,

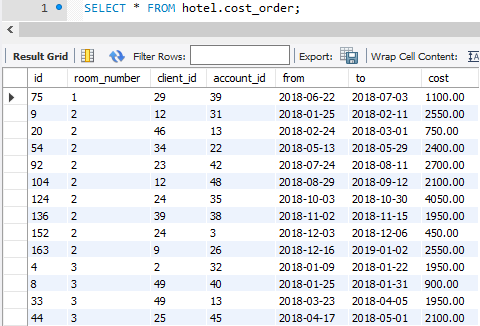
**(IF(TIMESTAMPDIFF(DAY**, `order`.`from`, `order`.`to`**)** **<> 0**,

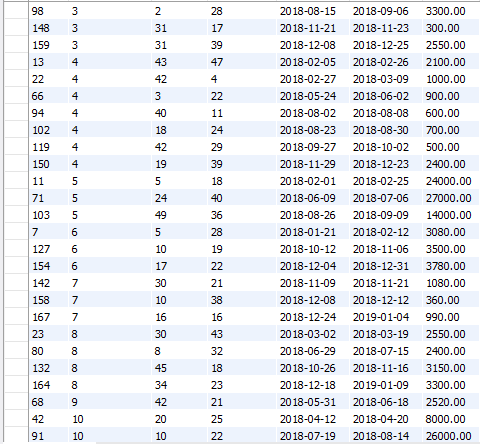
**TIMESTAMPDIFF(DAY**, `order`.`from`, `order`.`to`**)**, **1)** **\*** `room`.`price`**)AS** `cost`

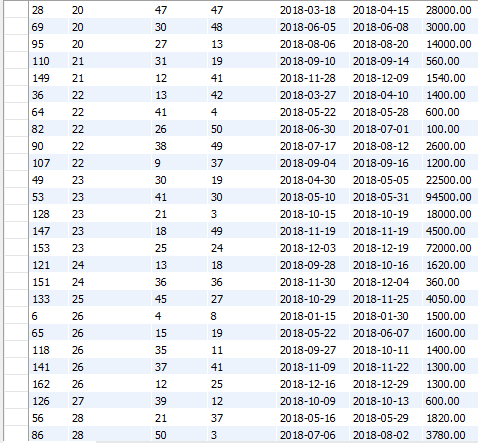
**FROM** `order` **JOIN** `room` **ON** `order`.`room\_number` **=** `room`.`number`)

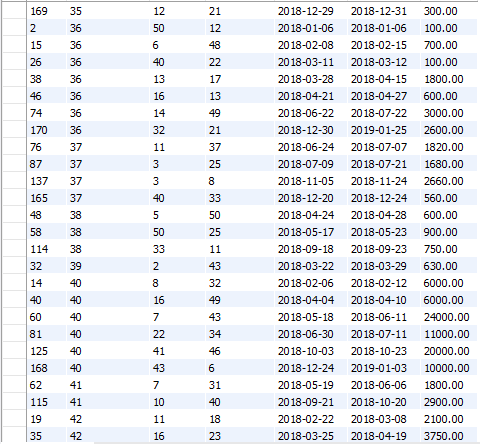
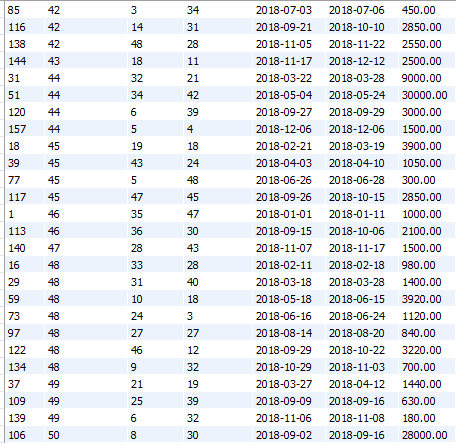
****

**SELECT** **\*** **FROM** hotel.cost\_order;

**1)**

**2)3)**

**4)5)**

**6)7)**

*5.2 Задача: Показать месяцы, в которые сумма заказов превышала среднюю сумму заказов по месяцам.*

-- 5.2 Задача: Показать месяцы, в которые сумма заказов превышала среднюю сумму заказов по месяцам.

**SELECT MONTHNAME**(`cost\_order`.`to`) **AS** `month`, **IFNULL**(**SUM**(`cost\_order`.`cost`),0)

**AS** `total`

**FROM** `cost\_order`

**GROUP BY** `month`

**HAVING** `total`**>** (**SELECT** **AVG**(`total`)

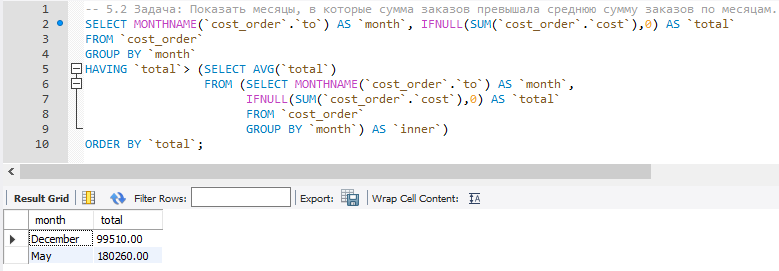
**FROM** (**SELECT** **MONTHNAME**(`cost\_order`.`to`) **AS** `month`,

**IFNULL**(**SUM**(`cost\_order`.`cost`),0) **AS** `total`

**FROM** `cost\_order`

**GROUP** **BY** `month`) **AS** `inner`)

**ORDER** **BY** `total`;



**6) Запросы на соединение таблиц(JOIN): INNER, LEFT (или RIGHT)**

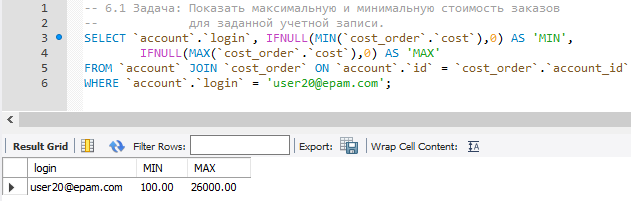
*6.1 Задача: Показать максимальную и минимальную стоимость заказов для заданной учетной записи.*

**SELECT** `account`.`login`, **IFNULL(MIN**(`cost\_order`.`cost`),**0)** AS **'MIN'**,

**IFNULL**(**MAX**(`cost\_order`.`cost`),0) AS **'MAX'**

**FROM** `account` JOIN `cost\_order` **ON** `account`.`id` **=** `cost\_order`.`account\_id`

**WHERE** `account`.`login` **=** 'user20@epam.com';



*6.2 Задача: Показать сколько раз заказывался каждый номер.*

**CREATE**

**ALGORITHM = UNDEFINED**

**DEFINER** = `root`@`localhost`

**SQL SECURITY DEFINER**

**VIEW** `times\_rooms` **AS**

(**SELECT**

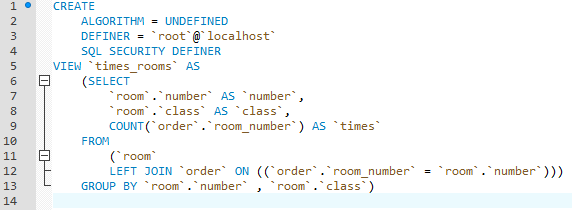
`room`.`number` **AS** `number`,

`room`.`class` **AS** `class`,

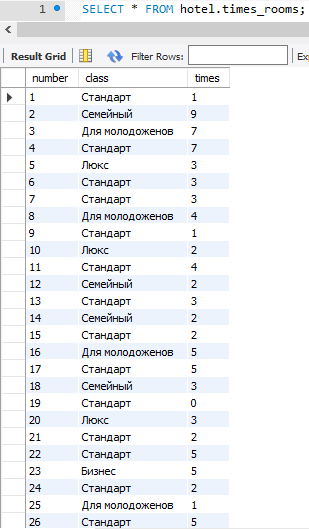
**COUNT**(`order`.`room\_number`) **AS** `times`

**FROM** `room`**LEFT** **JOIN** `order` **ON** `order`.`room\_number` = `room`.`number`

**GROUP** **BY** `room`.`number` , `room`.`class`)

1)

**SELECT** \* **FROM** hotel.times\_rooms;

2)

3)

**7) Запрос с GROUP BY и HAVING и агрегатными функциями**

*7.1 Задача: Показать учетные записи, с которых не производились заказы номеров.*

7.1 Задача: Показать учетные записи, с которых не производились заказы номеров.

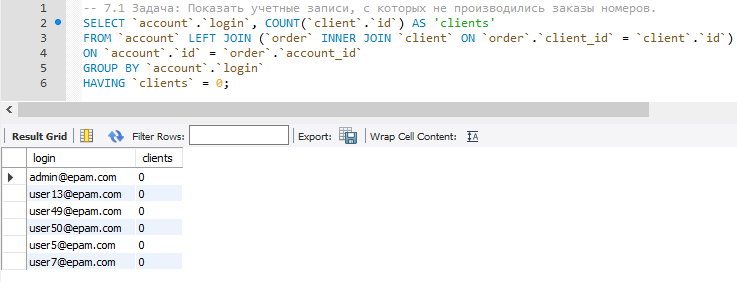
**SELECT** `account`.`login`, **COUNT**(`client`.`id`) **AS** 'clients'

**FROM** `account` **LEFT** **JOIN** **(**`order` **INNER** **JOIN** `client` **ON** `order`.`client\_id` =

`client`.`id`**)** **ON** `account`.`id` = `order`.`account\_id`

**GROUP** **BY** `account`.`login`

**HAVING** `clients` **= 0**;



**8) Запрос с UNION**

*8.1 Задача: Показать статус клиента. Статус зависит от общей суммы его заказов: не менее 40 000 – VIP; от 30 000(включительно) до 40 000 – IP; менее 30 000 – CLIENT.*

**SELECT** `first\_name`, `last\_name`, **SUM**(`cost`) **AS** 'total', **'VIP'** **AS** 'status'

**FROM** `client` **LEFT JOIN** `cost\_order` **ON** `client`.`id` = `cost\_order`.`client\_id`

**GROUP BY** `first\_name`, `last\_name`

**HAVING** total>=**40000**

**UNION**

**SELECT** `first\_name`, `last\_name`, **SUM**(`cost`) **AS** 'total', **'IP'**

**FROM** `client` **LEFT JOIN** `cost\_order` **ON** `client`.`id` = `cost\_order`.`client\_id`

**GROUP** **BY** `first\_name`, `last\_name`

**HAVING** `total`>=**30000** **AND** `total`<**40000**

**UNION**

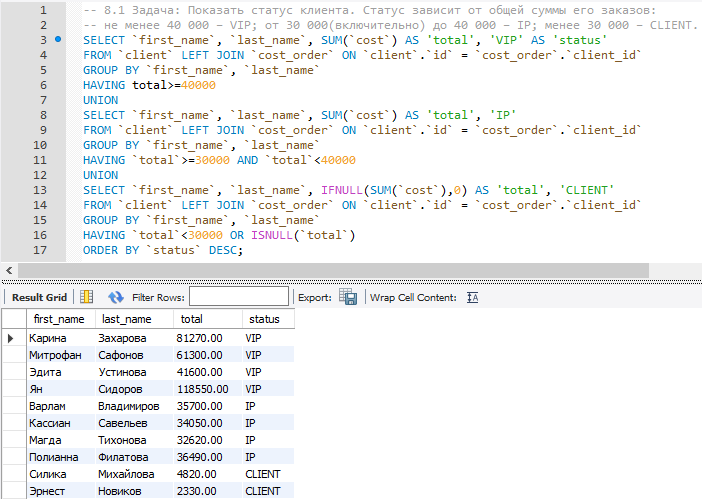
**SELECT** `first\_name`, `last\_name`, **IFNULL(SUM**(`cost`),**0)** **AS** 'total', **'CLIENT'**

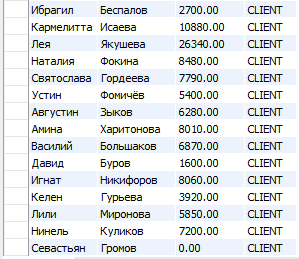
**FROM** `client` **LEFT JOIN** `cost\_order` **ON** `client`.`id` = `cost\_order`.`client\_id`

**GROUP** **BY** `first\_name`, `last\_name`

**HAVING** `total`<**30000** **OR** **ISNULL**(`total`)

**ORDER BY** `status` **DESC**;

1)

2)3)

**9) Запросы с подзапросами**

*9.1 Задача: Показать номера, заказываемые реже остальных в каждой категории.*

**SELECT** `outer`.`class`, `outer`.`number`

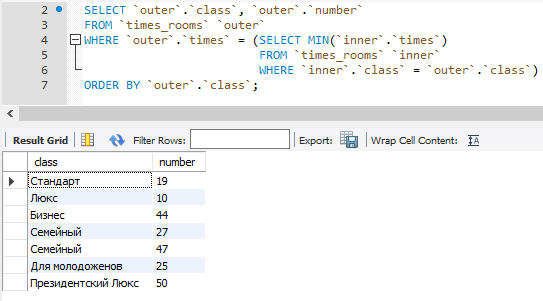
**FROM** `times\_rooms` `**outer**`

**WHERE** `outer`.`times` = (**SELECT** **MIN**(`inner`.`times`)

**FROM** `times\_rooms` `inner`

**WHERE** `**inner**`.`class` = `**outer**`.`class`)

**ORDER BY** `outer`.`class`;



*9.2 Задача: Показать свободные номера с подходящими классом, вместимостью и датами пребывания.*

**SET @input\_capasity** = 3;

**SET @input\_class** = "Стандарт";

**SET @input\_from** = '2018-01-19';

**SET @input\_to** = '2018-05-31';

**SELECT** `room`.`number`, `room`.`class`, `room`.`capacity`

**FROM** `room`

**WHERE** `capacity` = **@input\_capasity** **AND** `class` = **@input\_class**

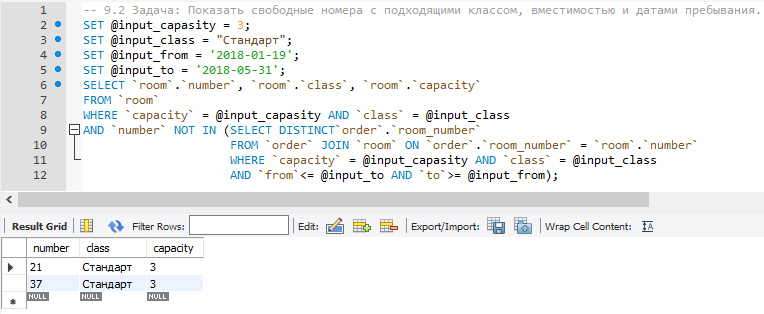
**AND** `number` **NOT IN** (**SELECT** **DISTINCT**`order`.`room\_number`

**FROM** `order` **JOIN** `room`

**ON** `order`.`room\_number` = `room`.`number`

**WHERE** `capacity` = **@input\_capasity** **AND** `class` = **@input\_class**

**AND** `from**`<= @input**\_to **AND** `to**`>= @input\_from**);

**