**Задание 1**

Используйте таблицу **flight\_detail** из схемы **course\_schema**.

Постройте запрос, который рассчитывает скользящее среднее количество пассажиров

**(moving\_avg\_pax)** за последние **3 рейса** (включая текущий)

**PostgreSQL code:**

**SELECT**

flight\_number,

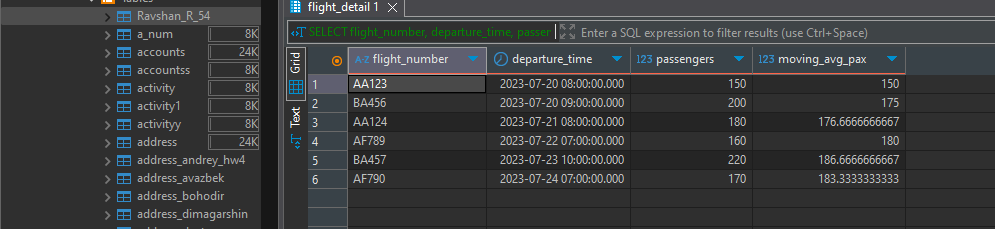
departure\_time,

passengers,

**avg**(passengers) **OVER** (**ORDER** **BY** departure\_time **ROWS** **BETWEEN** 2 **PRECEDING** **AND** **CURRENT** **ROW**) **AS** *moving\_avg\_pax*

**FROM** course\_schema.flight\_detail;

**Console output:**

****

**Задание 2**

Используйте таблицы **flight\_detail** и **aircompany\_detail.**

Создайте запрос, который ранжирует авиакомпании по общему количеству

перевезенных пассажиров.

Включите в запрос вывод ранга **(company\_rank)** и названия авиакомпании

**(aircompany\_name).**

**Попробуйте сделать это упражнение с**

1. Вложенным запросом или CTE

2. Одним скриптом, без вложенных конструкций

**PostgreSQL code:**

**WITH** *company\_passenger\_sum* **AS** (

**SELECT**

*fd*.aircompany\_id,

*ad*.aircompany\_name,

**sum**(*fd*.passengers) **AS** *total\_passengers*

**FROM** course\_schema.flight\_detail *fd*

**JOIN** course\_schema.aircompany\_detail *ad*

**ON** *fd*.aircompany\_id = *ad*.aircompany\_id

**GROUP** **BY** *fd*.aircompany\_id, *ad*.aircompany\_name

)

**SELECT**

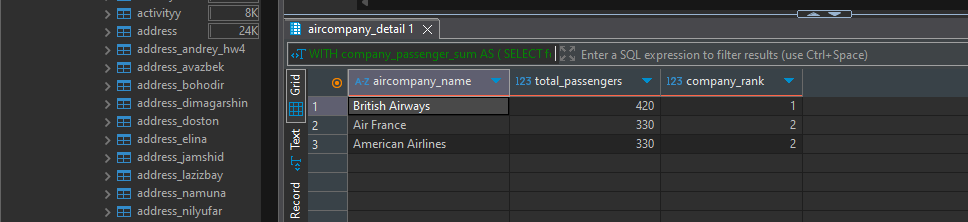
aircompany\_name,

*total\_passengers*,

**RANK**() **OVER** (**ORDER** **BY** *total\_passengers* **DESC**) **AS** *company\_rank*

**FROM** *company\_passenger\_sum*;

**Console output:**

****

**Задание 3**

Есть таблица сотрудников **course\_schema.employees .** Напишите запрос,

который для каждого сотрудника выводит:

1. сколько человек трудится в его отделе **( emp\_cnt );**

2. какая средняя зарплата по отделу **( sal\_avg );**

3. на сколько процентов отклоняется его зарплата от средней по отделу

**( diff ).**

\*для округления используйте функцию **round() .**

Сортировка результата: **department , salary , employee\_id**

**PostgreSQL code:**

**SELECT**

first\_name,

department,

salary,

**count**(\*) **OVER** (**PARTITION** **BY** department) **AS** emp\_cnt,

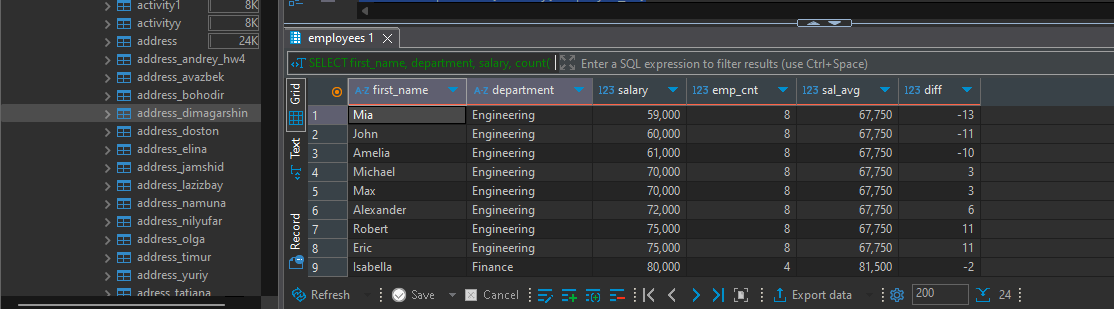
**round**(**AVG**(salary) **OVER** (**PARTITION** **BY** department), 2) **AS** *sal\_avg*,

**round**((salary - **AVG**(salary) **OVER** (**PARTITION** **BY** department)) / **avg**(salary) **OVER** (**PARTITION** **BY** department) \* 100, 0) **AS** *diff*

**FROM** course\_schema.employees

**ORDER** **BY** department, salary, employee\_id;

**Console output:**

****

**Задание 4**

Нужно посчитать фонд оплаты труда нарастающим итогом (кумулятивная сумма)

независимо для каждого департамента.

Сортировка результата: **department , salary , employee\_id**

**PostgreSQL code:**

**SELECT**

employee\_id,

first\_name,

department,

salary,

**sum**(salary) **OVER** (

**PARTITION** **BY** department

**ORDER** **BY** salary, employee\_id

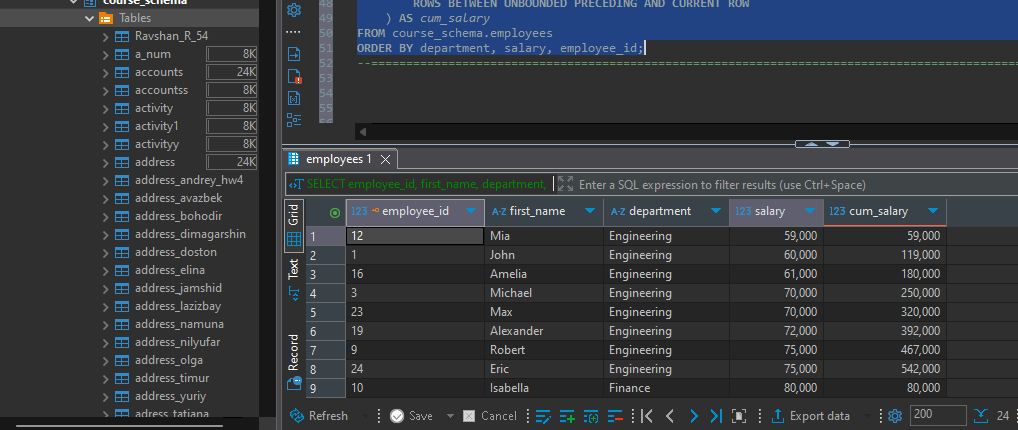
**ROWS** **BETWEEN** **UNBOUNDED** **PRECEDING** **AND** **CURRENT** **ROW**

) **AS** *cum\_salary*

**FROM** course\_schema.employees

**ORDER** **BY** department, salary, employee\_id;

**Console output:**

****