



1. **Из таблиц course\_schema.Flight\_detail и course\_schema.aircompany\_detail с помощью конструкции**

**CTE**

**Написать скрипт, который достает**

**- Название авиакомпании**

**- Номер рейса**

**- Количество пассажиров на рейсе**

**- Общее количество пассажиров, которая перевезла авиакомпания**

**with** *flight\_info\_CTE1* **as** (

**select**

*fd*.aircompany\_id,

*ac*.aircompany\_name,

*fd*.flight\_number,

*fd*.passengers

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

**join** course\_schema.aircompany\_detail *ac*

**on** *fd*.aircompany\_id = *ac*.aircompany\_id

),

*total\_passengers\_CTE2* **as** (

**select** aircompany\_id, **sum**(passengers) **as** *total\_passengers*

**from** *flight\_info\_CTE1*

**group** **by** aircompany\_id

)

**select**

*fi*.aircompany\_name,

*fi*.flight\_number,

*fi*.passengers,

*tp*.*total\_passengers*

**from** *flight\_info\_CTE1* *fi*

**join** *total\_passengers\_CTE2* *tp*

**on** *fi*.aircompany\_id = *tp*.aircompany\_id

**order** **by** *fi*.aircompany\_name, *fi*.flight\_number

**Задание 2**

**Из таблицы course\_schema.flight\_details с помощью вложенного запроса вывести номер рейса и**

**количество пассажиров, где количество пассажиров больше или равно среднему количеству**

**пассажиров в таблице course\_schema.flight\_details**

**select** flight\_number,passengers

**from** course\_schema.flight\_detail

**where** passengers >= (

**select** **avg**(passengers)

**from** course\_schema.flight\_detail)

**Задание 3**

**Получить полный список книг, которые есть в библиотеке из таблиц course\_schema.books\_archive**

**и course\_schema.books (убрать дубли)**

**Вывести весь список книг, добавить к нему колонку status, которая должна содержать текст**

**Дешевле среднего, в случае, когда цена ниже средней цены из общего списка книг и текст Дороже**

**среднего во всех иных случаях**

**with** *all\_books* **as** (

**select** book\_id, title, author, genre, price **from** course\_schema.books

**union**

**select** book\_id, title, author, genre, price **from** course\_schema.books\_archive

),

*avg\_price* **as** (

**select** **avg**(price) **as** *avg\_price* **from** *all\_books*

)

**select**

*ab*.book\_id,

*ab*.title,

*ab*.author,

*ab*.price,

**case** **when** *ab*.price < *ap*.*avg\_price* **then** 'Дешевле среднего'

**else** 'Дороже среднего'

**end** **as** *status*

**from** *all\_books* *ab*, *avg\_price* *ap*

**order** **by** *ab*.book\_id