**Задание 1**

1. **Уникальные названия маршрутов, для которых существуют билеты (есть запись в tickets). Вывести только названия:**

SELECT DISTINCT destination.name

FROM Destination

INNER JOIN Tickets ON Destination.id = Tickets.id\_destination;

1. **Дополните предыдущий запрос: ограничьте маршруты статусом «Без визы»:**

SELECT DISTINCT Destination.name

FROM Destination

INNER JOIN Tickets ON Destination.id = Tickets.id\_destination

INNER JOIN Status ON Destination.id\_status = Status.id

WHERE Status.name = 'Без визы';

1. **Найдите маршруты, максимальная цена которых выше общей средней. Общая средняя находится как среднее значение lowest\_price и highest\_price. Вывести названия и высшую цену:**

SELECT Destination.name, MAX(Tickets.highest\_price)

FROM Destination

INNER JOIN Tickets ON Destination.id = Tickets.id\_destination

GROUP BY Destination.name

HAVING MAX(Tickets.highest\_price) > (SELECT AVG((lowest\_price + highest\_price) / 2) FROM Tickets);

**Задание 2**

1. **Список уникальных классов (только названия):**

SELECT DISTINCT class\_name FROM visits;

1. **Количество часов, проведенных на занятиях, для каждого пользователя (фамилия, имя и количество часов):**

SELECT u.user\_surname, u.user\_name, SUM(v.hours\_spent) AS total\_hours

FROM user u

JOIN visits v ON u.id\_user = v.id\_user

GROUP BY u.user\_surname, u.user\_name;

1. **Средний возраст пользователей, посещающих класс Flex:**

SELECT AVG(u.age) AS average\_age

FROM user u

JOIN visits v ON u.id\_user = v.id\_user

WHERE v.class\_name = 'Flex';

**Задание 3**

1. **Уникальные названия всех книг, опубликованных после 1990 года:**

SELECT DISTINCT title

FROM book

WHERE year\_puplish > 1990;

1. **Для каждого автора найти сумму напечатанных страниц и вывести полное имя автора и сумму страниц:**

SELECT a.full\_name, SUM(b.pages) AS total\_pages

FROM author AS a

INNER JOIN book AS b ON a.id\_author = b.id\_author

GROUP BY a.full\_name;

1. **Подсчитать количество книг авторов каждого века и вывести век и количество книг:**

SELECT

CASE

WHEN century = 19 THEN 'XIX век'

WHEN century = 20 THEN 'XX век'

WHEN century = 21 THEN 'XXI век'

ELSE 'Другой век'

END AS century,

COUNT(\*) AS book\_count

FROM author AS a

INNER JOIN book AS b ON a.id\_author = b.id\_author

GROUP BY century;