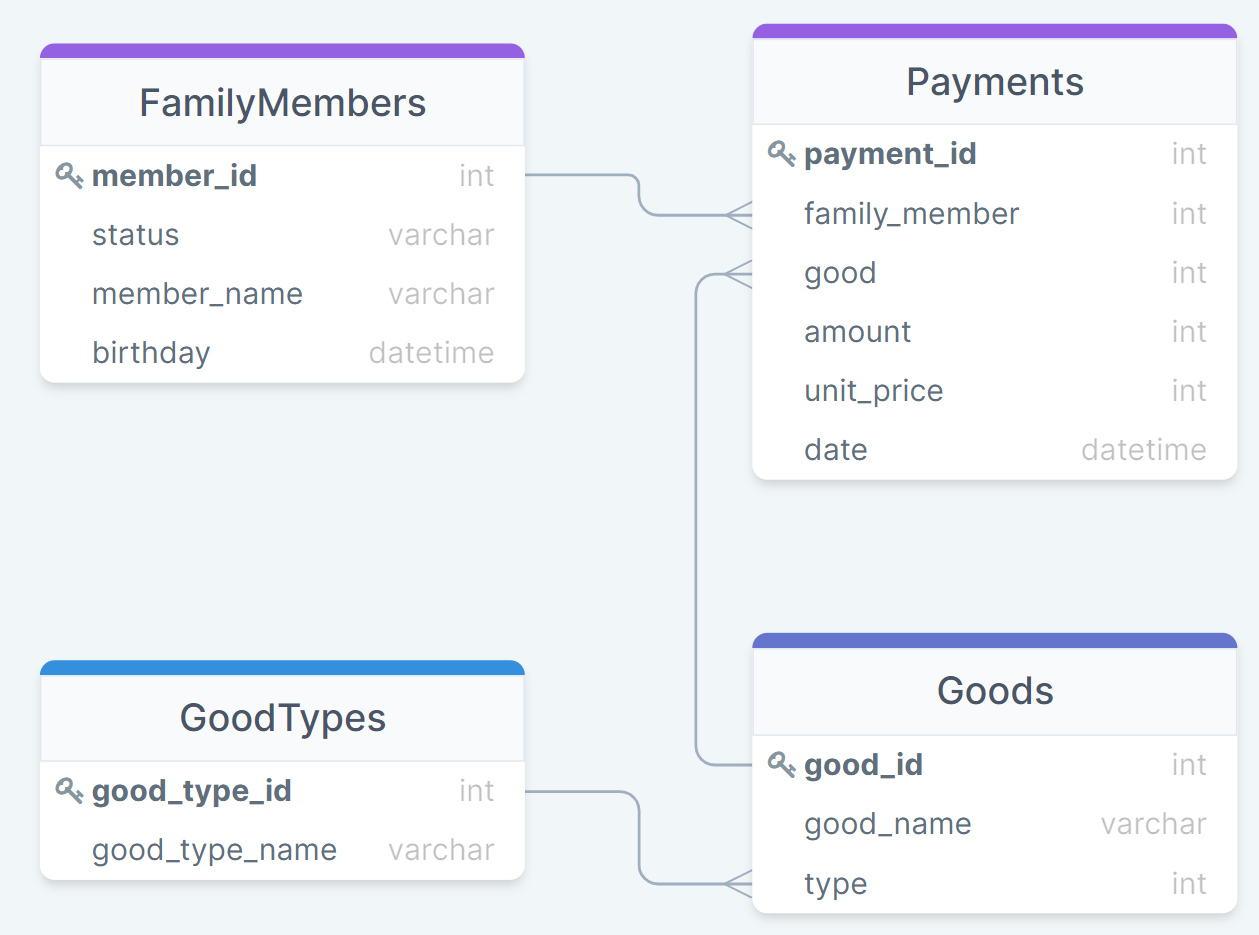


|  |  |  |
| --- | --- | --- |
| № | Task | Query |
| [1](https://sql-academy.org/ru/trainer/tasks/1) | Вывести имена всех когда-либо обслуживаемых пассажиров авиакомпаний | SELECT name  FROM Passenger; |
| [2](https://sql-academy.org/ru/trainer/tasks/2) | Вывести названия всех авиакомпаний | SELECT name  FROM Company; |
| [3](https://sql-academy.org/ru/trainer/tasks/3) | Вывести все рейсы, совершенные из Москвы | SELECT \*  FROM Trip  WHERE town\_from = "Moscow"; |
| [4](https://sql-academy.org/ru/trainer/tasks/4) | Вывести имена людей, которые заканчиваются на "man" | SELECT name  FROM Passenger  WHERE name LIKE '%man'; |
| [5](https://sql-academy.org/ru/trainer/tasks/5) | Вывести количество рейсов, совершенных на TU-134 | SELECT COUNT(\*) AS count  FROM Trip  WHERE plane = "TU-134"; |
| [6](https://sql-academy.org/ru/trainer/tasks/6) | Какие компании совершали перелеты на Boeing | SELECT DISTINCT Company.name AS name  FROM Company INNER JOIN Trip ON Company.id = Trip.company  WHERE Trip.plane = "Boeing"; |
| [7](https://sql-academy.org/ru/trainer/tasks/7) | Вывести все названия самолётов, на которых можно улететь в Москву (Moscow) | SELECT DISTINCT plane  FROM Trip  WHERE town\_to = "Moscow"; |
| [8](https://sql-academy.org/ru/trainer/tasks/8) | В какие города можно улететь из Парижа (Paris) и сколько времени это займёт? | SELECT town\_to, CAST((time\_in - time\_out) AS time(0)) AS flight\_time  FROM Trip  WHERE town\_from = "Paris"; |
| [9](https://sql-academy.org/ru/trainer/tasks/9) | Какие компании организуют перелеты с Владивостока (Vladivostok)? | SELECT DISTINCT name  FROM Company INNER JOIN Trip ON Company.id = Trip.company  WHERE Trip.town\_from = "Vladivostok"; |
| [10](https://sql-academy.org/ru/trainer/tasks/10) | Вывести вылеты, совершенные с 10 ч. по 14 ч. 1 января 1900 г. | SELECT \*  FROM Trip  WHERE time\_out BETWEEN "1900-01-01 10:00:00" AND "1900-01-01 14:00:00" |

|  |  |  |
| --- | --- | --- |
| № | Task | Query |
| [11](https://sql-academy.org/ru/trainer/tasks/11) | Вывести пассажиров с самым длинным именем | SELECT name  FROM Passenger  WHERE LENGTH(name) = (SELECT MAX(LENGTH(name)) FROM Passenger); |
| [12](https://sql-academy.org/ru/trainer/tasks/12) | Вывести id и количество пассажиров для всех прошедших полётов | SELECT trip, COUNT(trip) AS count  FROM Pass\_in\_trip  GROUP BY trip; |
| [13](https://sql-academy.org/ru/trainer/tasks/13) | Вывести имена людей, у которых есть полный тёзка среди пассажиров | SELECT name  FROM Passenger  GROUP BY name  HAVING COUNT(\*) > 1; |
| [14](https://sql-academy.org/ru/trainer/tasks/14) | В какие города летал Bruce Willis | SELECT town\_to  FROM Trip INNER JOIN Pass\_in\_trip  ON Trip.id = Pass\_in\_trip.trip INNER JOIN Passenger  ON Pass\_in\_trip.passenger = Passenger.id  WHERE Passenger.name = "Bruce Willis"; |
| [15](https://sql-academy.org/ru/trainer/tasks/15) | Во сколько Стив Мартин (Steve Martin) прилетел в Лондон (London) | SELECT time\_in  FROM Trip INNER JOIN Pass\_in\_trip  ON Trip.id = Pass\_in\_trip.trip INNER JOIN Passenger  ON Pass\_in\_trip.passenger = Passenger.id  WHERE Passenger.name = "Steve Martin" AND Trip.town\_to = "London"; |
| [16](https://sql-academy.org/ru/trainer/tasks/16) | Вывести отсортированный по количеству перелетов (по убыванию) и имени (по возрастанию) список пассажиров, совершивших хотя бы 1 полет. | SELECT name, COUNT(name) AS count  FROM Passenger INNER JOIN Pass\_in\_trip ON Passenger.id = Pass\_in\_trip.passenger  GROUP BY name  HAVING COUNT(\*) >= 1  ORDER BY count DESC, name ASC; |



|  |  |  |
| --- | --- | --- |
| № | Task | Query |
| [17](https://sql-academy.org/ru/trainer/tasks/17) | Определить, сколько потратил в 2005 году каждый из членов семьи | SELECT member\_name, status, SUM(unit\_price \* amount) AS costs  FROM FamilyMembers INNER JOIN Payments  ON FamilyMembers.member\_id = Payments.family\_member  WHERE date BETWEEN "2005-01-01" AND "2006-01-01"  GROUP BY member\_name, status; |
| [18](https://sql-academy.org/ru/trainer/tasks/18) | Узнать, кто старше всех в семьe | SELECT member\_name  FROM FamilyMembers  WHERE birthday = (SELECT MIN(birthday) FROM FamilyMembers); |
| [19](https://sql-academy.org/ru/trainer/tasks/19) | Определить, кто из членов семьи покупал картошку (potato) | SELECT DISTINCT status  FROM FamilyMembers INNER JOIN Payments  ON FamilyMembers.member\_id = Payments.family\_member INNER JOIN Goods  ON Payments.good = Goods.good\_id  WHERE good\_name = "potato"; |
| [20](https://sql-academy.org/ru/trainer/tasks/20) | Сколько и кто из семьи потратил на развлечения (entertainment). Вывести статус в семье, имя, сумму | SELECT status, member\_name, SUM(amount \* unit\_price) AS costs  FROM FamilyMembers INNER JOIN Payments  ON FamilyMembers.member\_id = Payments.family\_member INNER JOIN Goods  ON Payments.good = Goods.good\_id INNER JOIN GoodTypes  ON Goods.type = GoodTypes.good\_type\_id  WHERE GoodTypes.good\_type\_name = "entertainment"  GROUP BY status, member\_name; |
| [21](https://sql-academy.org/ru/trainer/tasks/21) | Определить товары, которые покупали более 1 раза | SELECT good\_name  FROM Goods INNER JOIN Payments  on Goods.good\_id = Payments.good  GROUP BY good\_name  HAVING COUNT(\*) > 1; |

|  |  |  |
| --- | --- | --- |
| № | Task | Query |
| [22](https://sql-academy.org/ru/trainer/tasks/22) | Найти имена всех матерей (mother) | SELECT member\_name  FROM FamilyMembers  WHERE status = "mother"; |
| [23](https://sql-academy.org/ru/trainer/tasks/23) | Найдите самый дорогой деликатес (delicacies) и выведите его стоимость | SELECT good\_name, unit\_price  FROM GoodTypes INNER JOIN Goods  ON good\_type\_id = type INNER JOIN Payments  ON good\_id = good  WHERE good\_type\_name = "delicacies"  ORDER BY unit\_price DESC  LIMIT 1 |
| [24](https://sql-academy.org/ru/trainer/tasks/24) | Определить кто и сколько потратил в июне 2005 | SELECT member\_name, SUM(amount \* unit\_price) AS costs  FROM FamilyMembers INNER JOIN Payments  ON FamilyMembers.member\_id = Payments.family\_member  WHERE date BETWEEN "2005-06-01" AND "2005-06-30"  GROUP BY member\_name; |
| [25](https://sql-academy.org/ru/trainer/tasks/25) | Определить, какие товары имеются в таблице Goods, но не покупались в течение 2005 года | SELECT DISTINCT good\_name  FROM Goods  WHERE good\_name NOT IN (  SELECT DISTINCT good\_name  FROM Goods INNER JOIN Payments  ON Goods.good\_id = Payments.good  WHERE date BETWEEN "2005-01-01" AND "2005-12-31"); |
| [26](https://sql-academy.org/ru/trainer/tasks/26) | Определить группы товаров, которые не приобретались в 2005 году | SELECT DISTINCT good\_type\_name  FROM GoodTypes  WHERE good\_type\_name NOT IN (  SELECT DISTINCT good\_type\_name  FROM GoodTypes INNER JOIN Goods  ON GoodTypes.good\_type\_id = Goods.type INNER JOIN Payments  ON Goods.good\_id = Payments.good  WHERE date BETWEEN "2005-01-01" AND "2005-12-31"); |
| [27](https://sql-academy.org/ru/trainer/tasks/27) | Узнать, сколько потрачено на каждую из групп товаров в 2005 году. Вывести название группы и сумму | SELECT good\_type\_name, SUM(amount \* unit\_price) AS costs  FROM GoodTypes INNER JOIN Goods  ON GoodTypes.good\_type\_id = Goods.type INNER JOIN Payments  ON Goods.good\_id = Payments.good  WHERE date BETWEEN "2005-01-01" AND "2005-12-31"  GROUP BY good\_type\_name; |

|  |  |  |
| --- | --- | --- |
| № | Task | Query |
| [31](https://sql-academy.org/ru/trainer/tasks/31) | Вывести всех членов семьи с фамилией Quincey | SELECT \*  FROM FamilyMembers  WHERE member\_name LIKE '%Quincey'; |
| [32](https://sql-academy.org/ru/trainer/tasks/32) | Вывести средний возраст людей (в годах), хранящихся в базе данных. Результат округлите до целого в меньшую сторону | SELECT FLOOR(AVG(TIMESTAMPDIFF(YEAR, birthday, CURRENT\_TIMESTAMP))) AS age  FROM FamilyMembers; |
| [33](https://sql-academy.org/ru/trainer/tasks/33) | Найдите среднюю стоимость икры. В базе данных хранятся данные о покупках красной (red caviar) и черной икры (black caviar) | SELECT AVG(unit\_price) AS cost  FROM Payments INNER JOIN Goods  ON Payments.good = Goods.good\_id  WHERE good\_name IN( "red caviar", "black caviar"); |
|  |  |  |
|  |  |  |
|  |  |  |