



СОРЕВНОВАНИЯ

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [МОИ ПОСЫЛКИ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

S2. Транспортировка hard (15 баллов)

ограничение по времени на тест: 15 секунд[Ⓢ]

ограничение по памяти на тест: 1024 мегабайта

ввод: стандартный ввод

вывод: стандартный вывод

Это необычная задача — вам надо написать SQL-запрос. В качестве решения вы должны отослать один запрос к базе данных, который возвращает требуемые данные. Запрос может содержать произвольное количество подзапросов, других конструкций, быть сколь угодно навороченным, но это должен быть один запрос (в нём не должна встречаться точка с запятой для разделения разных запросов).

При проверке вашего решения используется PostgreSQL 15.1. В качестве входных данных вам предоставляется дамп состояния базы данных. Обратите внимание, что время работы вашего решения на тесте включает восстановление состояния базы данных из дампа, но это время значительно меньше ограничения по времени. Вы можете использовать сторонний сервис <http://sqlfiddle.com/> как инструмент для запуска запросов.

В этой задаче вам предстоит написать запрос к базе данных перемещений посылок между фулфилментами.

Фулфилмент — это склад, на который и от которого могут доставлять посылки.

Напишите запрос к базе данных, который возвращает информацию про время хранения посылок на фулфилментах:

- идентификатор фулфилмента — соответствующую колонку в выводе следует назвать `fulfilment_id`;
- название фулфилмента — соответствующую колонку в выводе следует назвать `fulfilment_name`;
- идентификатор посылки — соответствующую колонку в выводе следует назвать `package_id`;
- имя посылки — соответствующую колонку в выводе следует назвать `package_name`;
- время прибытия посылки в фулфилмент — соответствующую колонку в выводе следует назвать `time_in`;
- время отбытия посылки из фулфилмента — соответствующую колонку в выводе следует назвать `time_out`;
- время хранения посылки в фулфилменте (т.е. время, прошедшее от прибытия на фулфилмент до отбытия от фулфилмента) — соответствующую колонку в выводе следует назвать `storage_time`.

Строки в выводе сортируйте в первую очередь по невозрастанию `storage_time`, затем по возрастанию `time_in`, затем по возрастанию `fulfilment_id`, затем по возрастанию `package_id`.

Внимательно ознакомьтесь с примерами вывода. Ваш запрос должен иметь в точности такой же вывод, как на примерах.

Схема базы данных содержит четыре таблицы:

- `transportations` — перемещения посылок между фулфилментами (описываются полями: `id` (идентификатор транспортировки), `package_id` (идентификатор посылки), `source_id` (идентификатор фулфилмента, из которого отправляется посылка), `destination_id` (идентификатор фулфилмента, в

Route 256: Junior

Участник

→ О группе



→ Соревнования группы

- Дорешивание
- Контекст - C# (Juniors)
- Контекст - Go (Juniors)
- Песочница (C#)
- Песочница (Go)

Контекст - Go (Juniors)

Закончено

Участник

→ Пересчёт ограничений по времени

Это соревнование использует политику пересчёта ограничений по времени по языкам программирования. Система автоматически увеличивает ограничения по времени для некоторых языков в соответствии с множителями. Независимо от множителя языка, ограничение по времени не может превысить 30 секунд. Прочтите детали [по ссылке](#).

→ Языки

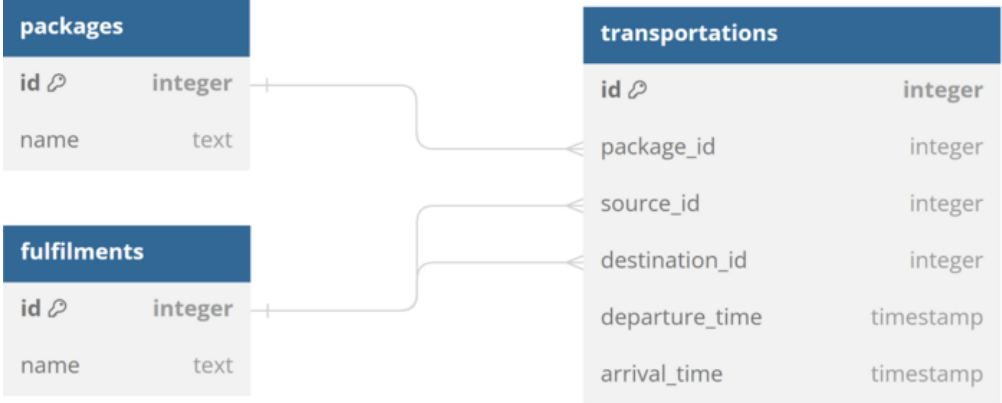
Только перечисленные языки могут быть использованы для решения задач соревнования

Контекст - Go (Juniors):

- GNU GCC C11 5.1.0
- Clang++20 Diagnostics
- Clang++17 Diagnostics
- GNU G++14 6.4.0
- GNU G++17 7.3.0
- GNU G++20 11.2.0 (64 bit, winlibs)

- который приходит посылка), departure_time (время отправки посылки из фулфилмента source_id), arrival_time (время прибытия посылки в фулфилмент destination_id));
- packages — посылки (описываются двумя полями: id и name),
 - fulfilments — фулфилменты (описываются двумя полями: id и name).

Таким образом, packages и transportations находятся в отношении «один ко многим», fulfilments и transportations находятся в отношении «многие к одному».



Любая посылка может быть перемещена в один и тот же фулфилмент несколько раз. В том числе, посылка может быть перемещена из фулфилмента в него же (source_id равен destination_id).

Изучите входные данные примера, чтобы подробно ознакомиться со схемой базы данных. Диаграмма выше иллюстрирует схему базы данных.

Входные данные

Входными данными в этой задаче является дамп базы данных. Вам он может быть полезен для ознакомления с состоянием базы данных для конкретного теста. В качестве решения вы должны отправить один SQL-запрос.

Выходные данные

Внимательно ознакомьтесь с примерами вывода. Ваш запрос должен иметь **в точности** такой же вывод, как на примерах.

Пример

входные данные

Скопировать

```
create table packages
(
    id      bigint primary key,
    name text not null
);

create table fulfilments
(
    id      bigint primary key,
    name text not null
);

create table transportations
(
    id              bigint primary key,
    package_id      bigint      not null,
    source_id       bigint      not null,
    destination_id  bigint      not null,
    departure_time  timestamp not null,
    arrival_time    timestamp not null,
    constraint fk_transportations_package_id foreign key (package_id)
references packages (id),
    constraint fk_transportations_source_id foreign key (source_id)
references fulfilments (id),
    constraint fk_transportations_destination_id foreign key
(destination_id) references fulfilments (id)
);

insert into packages
values (1, 'Package slow'),
      (2, 'Package fast'),
      (3, 'Package strange');

insert into fulfilments
```

- Microsoft Visual C++ 2017
- GNU G++17 9.2.0 (64 bit, msys 2)
- C# 8, .NET Core 3.1
- C# 10, .NET SDK 6.0
- C# Mono 6.8
- Go 1.19.5
- Java 11.0.6
- Java 17 64bit
- Java 1.8.0_241
- Delphi 7
- Free Pascal 3.0.2
- PascalABC.NET 3.8.3
- PHP 8.1.7
- PostgreSQL 15.1
- Python 2.7.18
- Python 3.8.10
- PyPy 2.7.13 (7.3.0)
- PyPy 3.6.9 (7.3.0)
- PyPy 3.9.10 (7.3.9, 64bit)

→ Набранные баллы

	Баллы
A	5
B	10
C	10
D	15
E	10
F	25
G	
H	
S1	
S2	
Всего	75

- Материалы соревнования
- problem-b-tests.zip
 - problem-c-tests.zip
 - problem-d-tests.zip
 - problem-e-tests.zip
 - problem-f-tests.zip
 - problem-g-tests.zip
 - problem-h-tests.zip
 - problem-s1-tests.zip
 - problem-s2-tests.zip

```
values (1, 'Moscow'),
(2, 'Spb'),
(3, 'Almaty'),
(4, 'Novosibirsk'),
(5, 'Vladivostok');

insert into transportations
values (1, 1, 1, 3, '2023-01-01 12:00:00', '2023-01-13 12:00:00'),
(2, 1, 3, 3, '2023-01-15 12:00:00', '2023-01-31 12:00:00'),
(3, 1, 3, 5, '2023-02-27 12:00:00', '2023-03-14 12:00:00'),
(4, 1, 5, 5, '2023-03-15 12:00:00', '2023-03-28 12:00:00'),
(5, 2, 1, 2, '2023-02-15 12:00:00', '2023-02-15 18:00:00'),
(6, 3, 2, 3, '2023-04-02 12:00:00', '2023-04-05 12:00:00'),
(7, 3, 3, 5, '2023-04-07 12:00:00', '2023-04-28 12:00:00');
```

ВЫХОДНЫЕ ДАННЫЕ

Скопировать

fulfilment_id	fulfilment_name	package_id	package_name	time_in	time_out	storage_time
	3 Almaty	1	Package slow	2023-01-13 12:00:00	2023-01-31 12:00:00	27 days
	3 Almaty	1	Package slow	2023-01-15 12:00:00	2023-02-27 12:00:00	2 days
	3 Almaty	3	Package strange	2023-04-02 12:00:00	2023-04-05 12:00:00	2 days
	5 Vladivostok	1	Package slow	2023-03-14 12:00:00	2023-03-15 12:00:00	1 day
(4 rows)						

Примечание

Разберем пример из условия.

Посылка 1 (Package slow) с 2023-01-13 12:00:00 до 2023-01-15 12:00:00 находилась в фулфилменте 3 первый раз; с 2023-01-31 12:00:00 до 2023-02-27 12:00:00 находилась в фулфилменте 3 второй раз; с 2023-03-14 12:00:00 до 2023-03-15 12:00:00 находилась в фулфилменте 2.

Посылка 3 (Package strange) с 2023-04-05 12:00:00 до 2023-04-07 12:00:00 находилась в фулфилменте 3.

