# Отчёт по работе №2

# Язык SQL-DML

### Выборка данных

* однотабличная выборка
  1. Вывести ФИО персонала, упорядочив в алфавитном порядке по фамилии и в обратном порядке по имени

**SELECT last\_name, first\_name, pather\_name**

**FROM auto\_personnel**

**ORDER BY -- упорядочить**

**last\_name, -- в алфавитном порядке**

**first\_name DESC; -- в обратном порядке**



* 1. Посчитать количество автомобилей, у которых первая буква номера «р»

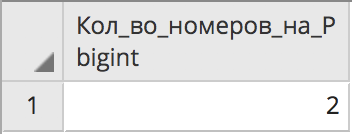
Таблица автомобилей



**SELECT Count(\*) AS Кол\_во\_номеров\_на\_Р**

**FROM auto**

**WHERE num LIKE 'P%';**

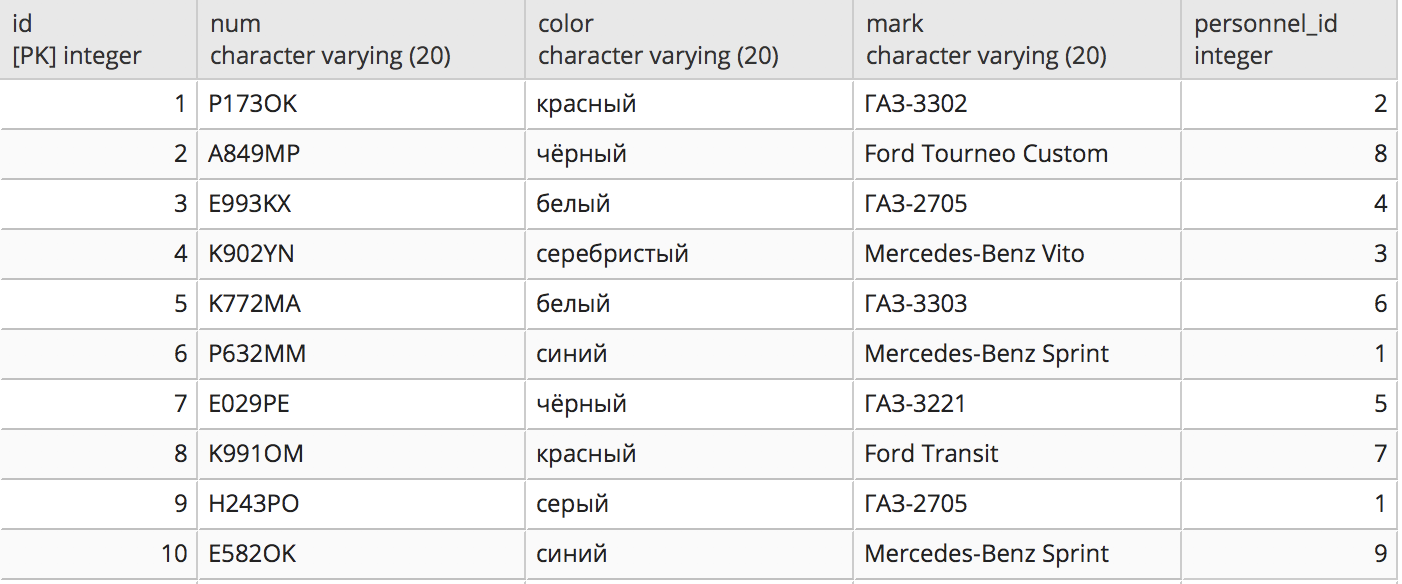


* выборка с подзапросами
  1. Вывести номера автомобилей водителя с заданной фамилией

Таблица водителей



Таблица автомобилей



**SELECT num**

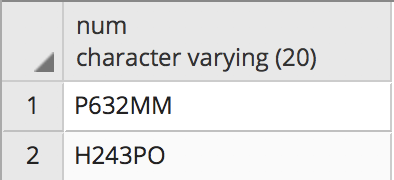
**FROM auto**

**WHERE personnel\_id =**

**( SELECT id**

**FROM auto\_personnel**

**WHERE last\_name = 'Назаров');**



* 1. Вывести наименования маршрутов, на которых есть не вернувшиеся автомобили

Таблица журнал

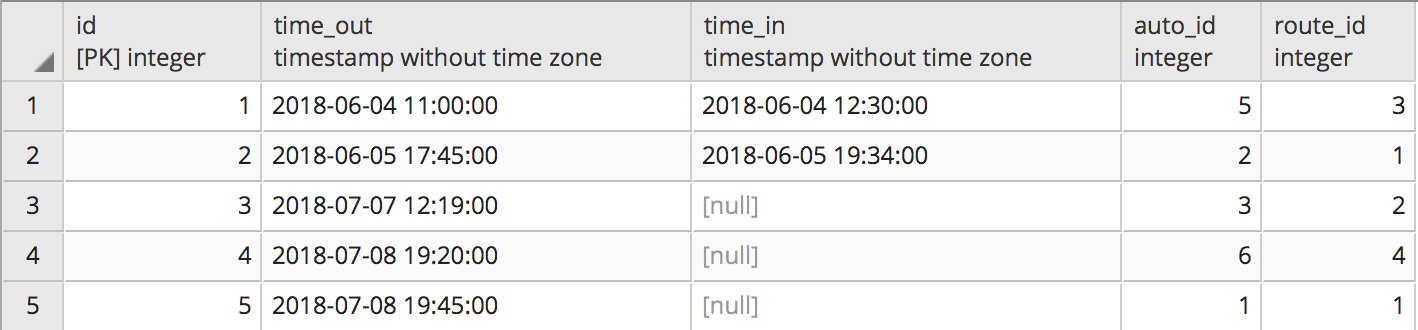
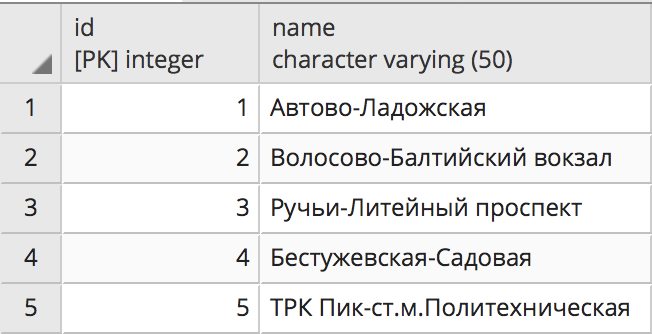


Таблица маршрутов



**SELECT name**

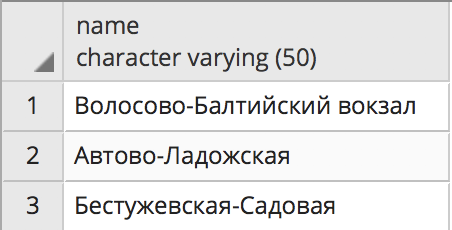
**FROM routes**

**WHERE id IN**

**( SELECT route\_id**

**FROM journal**

**WHERE time\_in is null)**



* соединение таблиц (join)
  1. Вывести время отправки из журнала оператора и наименования маршрутов, включая маршруты, по которым рейсов не было

**SELECT journal.time\_out, routes.name**

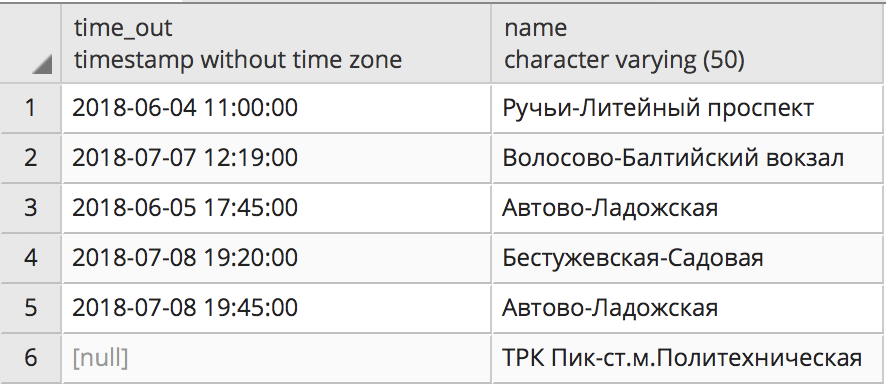
**FROM**

**journal -- Левая таблица**

**RIGHT OUTER JOIN**

**routes -- Правая таблица**

**ON journal.route\_id = routes.id**



* 1. Вывести времена отправки/прибытия из журнала оператора, наименования маршрутов поездок, включая маршруты, по которым не было рейсов, и номера автомобилей, включая автомобили которые не участвовали в рейсах.

**SELECT journal.time\_out, journal.time\_in, routes.name, auto.num**

**FROM**

**journal**

**RIGHT OUTER JOIN**

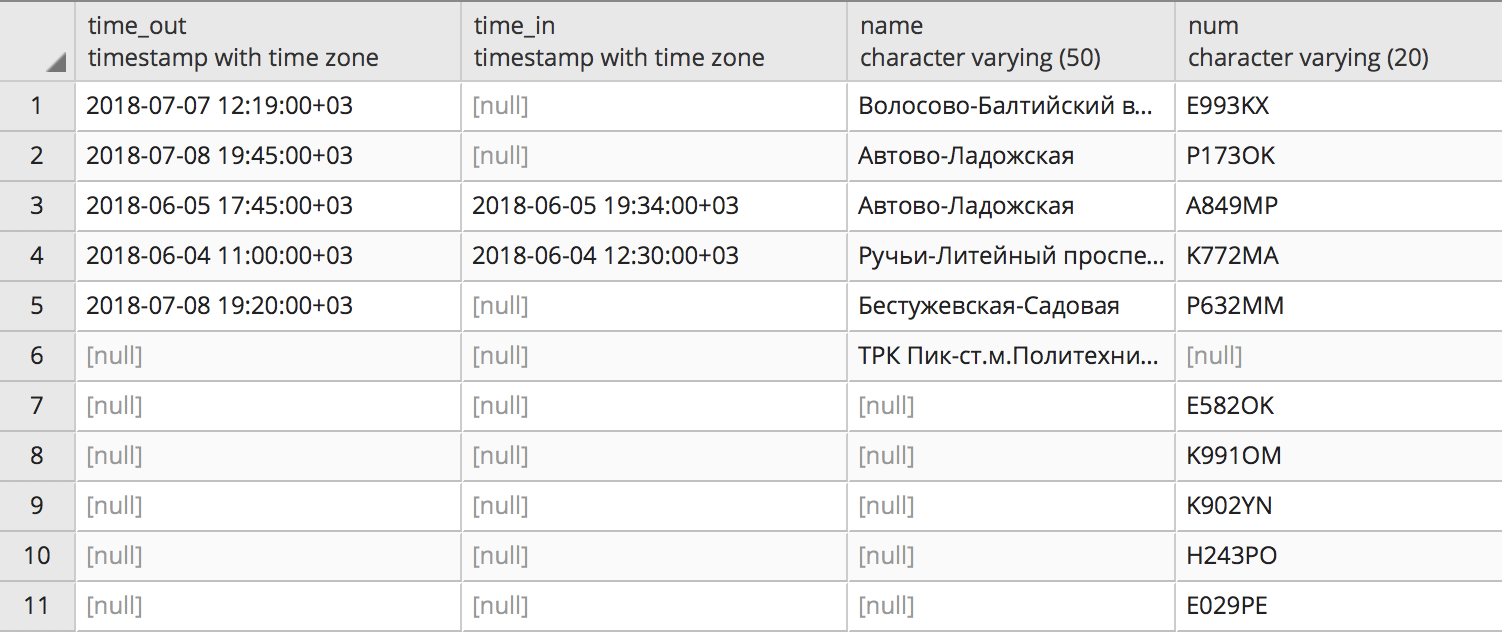
**routes**

**ON journal.route\_id = routes.id**

**FULL OUTER JOIN**

**auto**

**ON journal.auto\_id = auto.id**



* для реализации проекта
  1. Вывести самое короткое время проезда по заданному наименованием маршруту и номер автомобиля, который поставил рекорд.

**SELECT MIN(journal.time\_in-journal.time\_out) AS min\_time, auto.num**

**FROM journal, auto**

**WHERE route\_id =**

**( SELECT id**

**FROM routes**

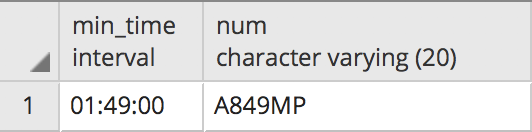
**WHERE name = 'Автово-Ладожская'**

**)**

**AND auto.id = journal.auto\_id**

**GROUP BY auto.num**

**LIMIT 1**



* 1. Количество автомобилей находящихся в рейсе по заданному наименованием маршруту

Таблица маршрутов

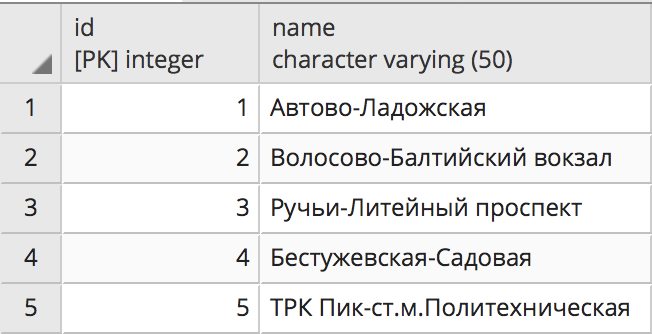
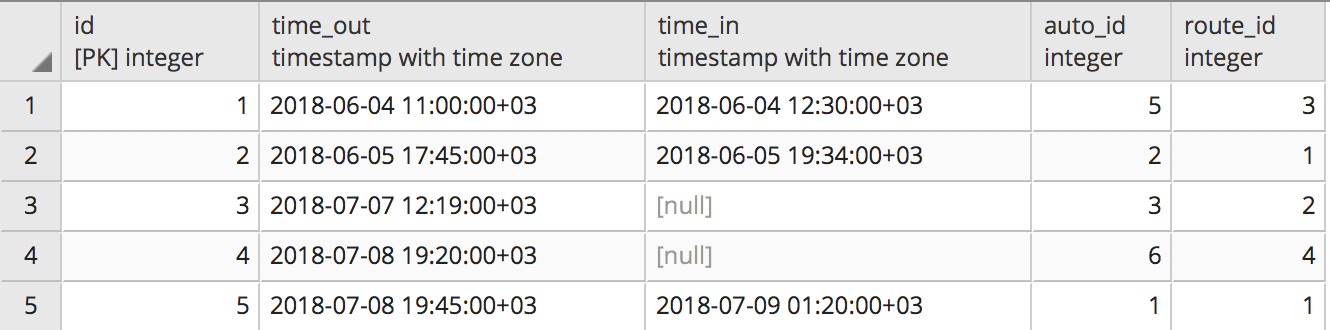


Таблица журнал



**SELECT COUNT(auto.id)**

**FROM auto, journal**

**WHERE auto.id = journal.auto\_id**

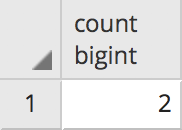
**AND journal.route\_id =**

**( SELECT id**

**FROM routes**

**WHERE name = 'Автово-Ладожская'**

**)**



Hints: select, count, join, where, in, exists, order by, group by, having

### Вставка данных

* однотабличная вставка
  1. Добавить водителю с заданной фамилией два автомобиля

Таблица водителей

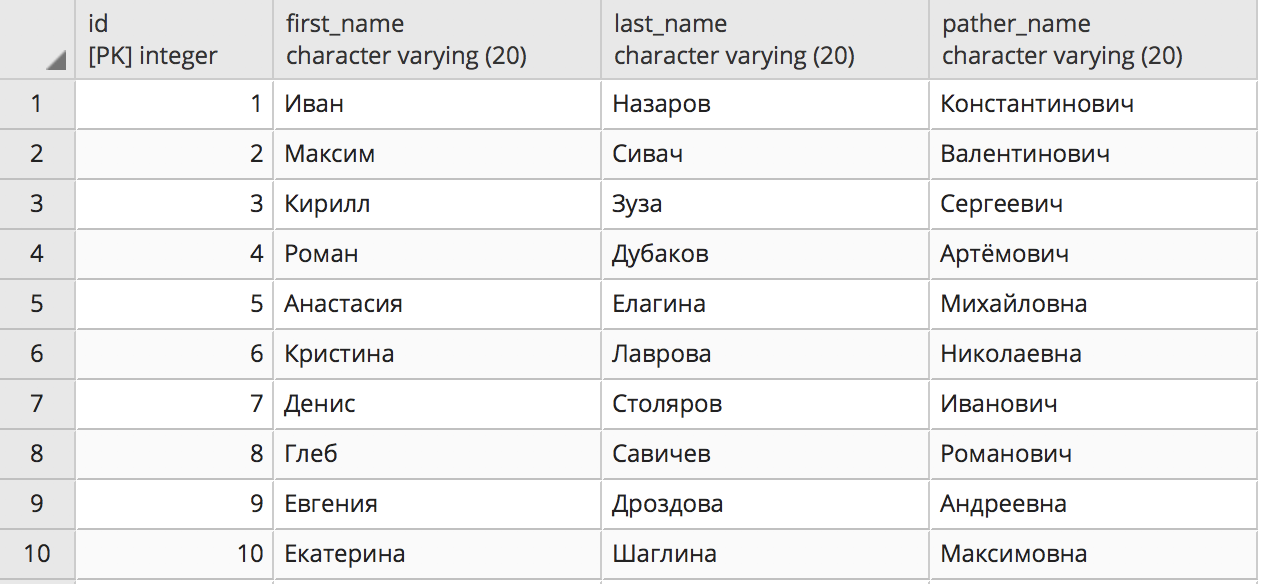


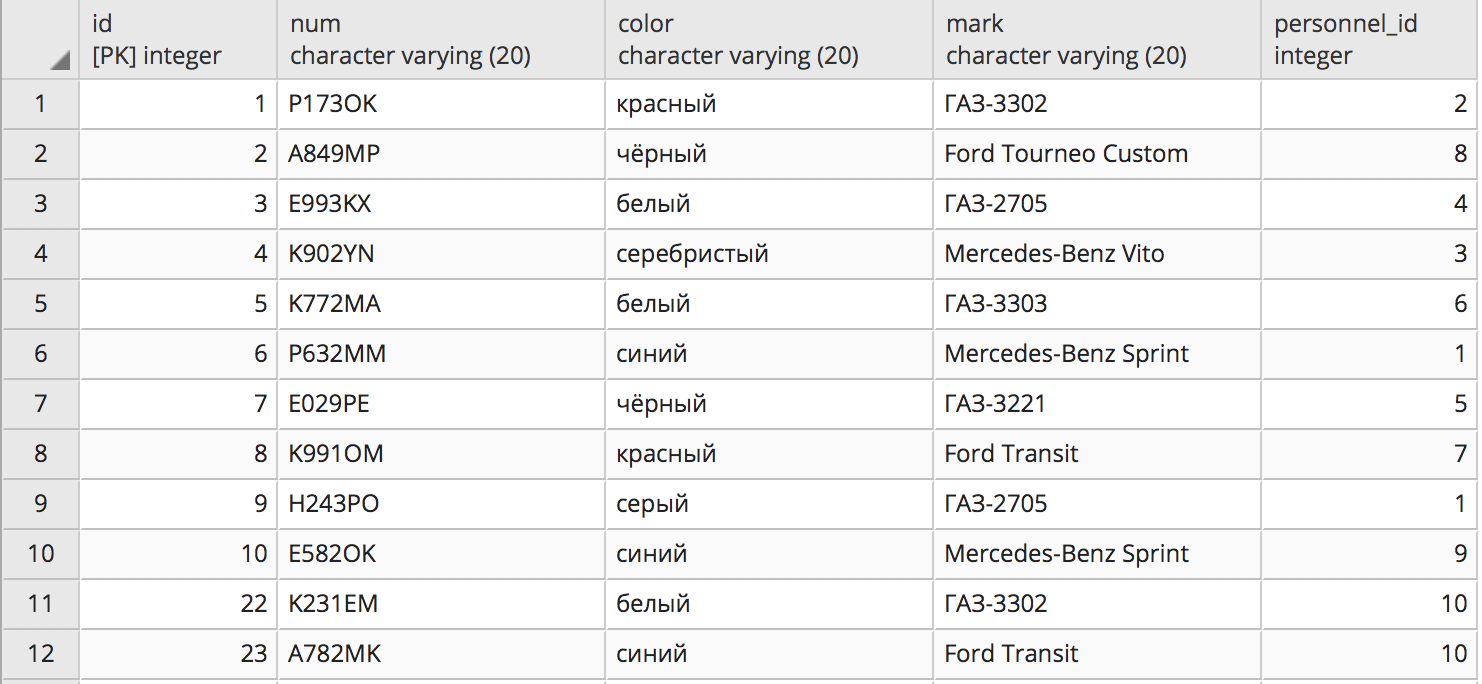
Таблица автомобилей



**INSERT INTO auto (num, color, mark, personnel\_id)**

**VALUES('K231EM', 'белый', 'ГАЗ-3302', (SELECT id FROM auto\_personnel WHERE last\_name = 'Шаглина')),**

**('A782MK', 'синий', 'Ford Transit', (SELECT id FROM auto\_personnel WHERE last\_name = 'Шаглина'))**



* 1. Отправить водителя с заданной фамилией на желтой машине в рейс по маршруту с известным id.

Таблица водители

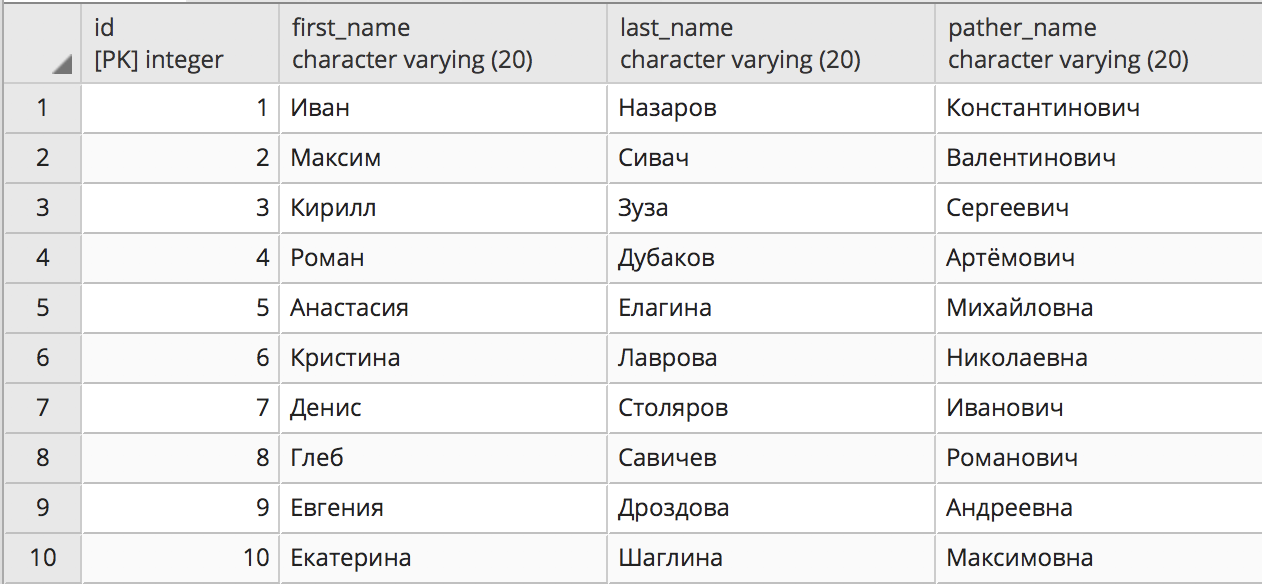
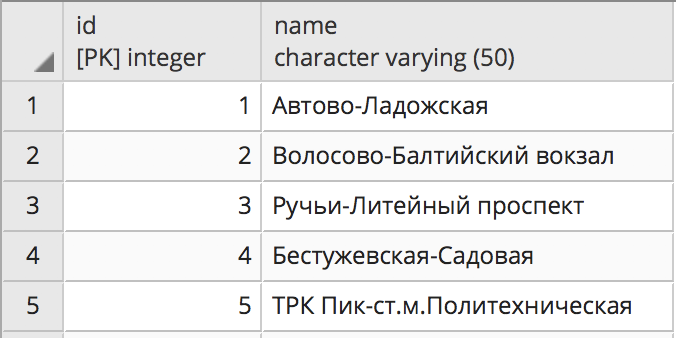


Таблица машины



Таблица маршруты



**INSERT INTO journal (time\_out, auto\_id, route\_id)**

**VALUES('now', (SELECT id**

**FROM auto**

**WHERE color = 'жёлтый' AND personnel\_id =**

**(SELECT id**

**FROM auto\_personnel**

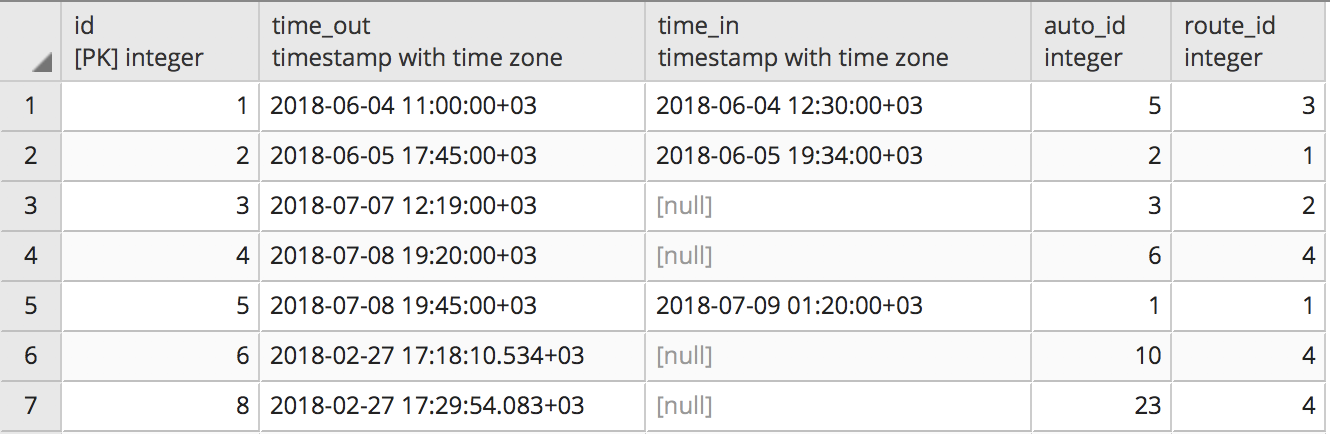
**WHERE last\_name = 'Шаглина'**

**)),**

**(SELECT id**

**FROM routes**

**WHERE id = 4));**



* многотабличная вставка в рамках транзакции
  1. Добавить запись в журнал, в случае, если автомобилей в данном рейсе больше 2-ух, транзакцию откатить

**START TRANSACTION;**

**DO $$**

**DECLARE**

**routeid INTEGER := 1;**

**BEGIN**

**IF (SELECT COUNT(auto\_id) > 2 FROM journal WHERE route\_id = routeid) THEN**

**INSERT INTO journal(time\_out, auto\_id, route\_id) VALUES('now', 4, routeid);**

**END IF;**

**END$$;**

**COMMIT;**

**START TRANSACTION;**

**DO $$**

**DECLARE**

**routeid INTEGER := 1;**

**BEGIN**

**IF (SELECT COUNT(auto\_id) > 2 FROM journal WHERE route\_id = routeid) THEN**

**INSERT INTO journal(time\_out, auto\_id, route\_id) VALUES('now', 4, routeid);**

**END IF;**

**END$$;**

**ROLLBACK;**

Hints: insert, where, in, exists, commit, rollback

### Удаление данных

* удаление по фильтру и удаление из связанных таблиц
  1. Удалить маршрут и все, связанные с ним, записи в журнале

Таблица маршрутов

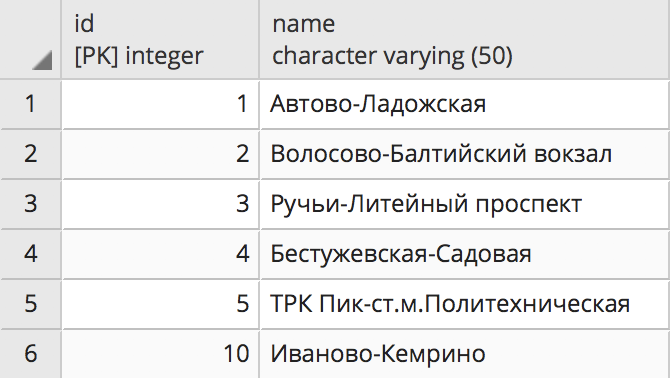
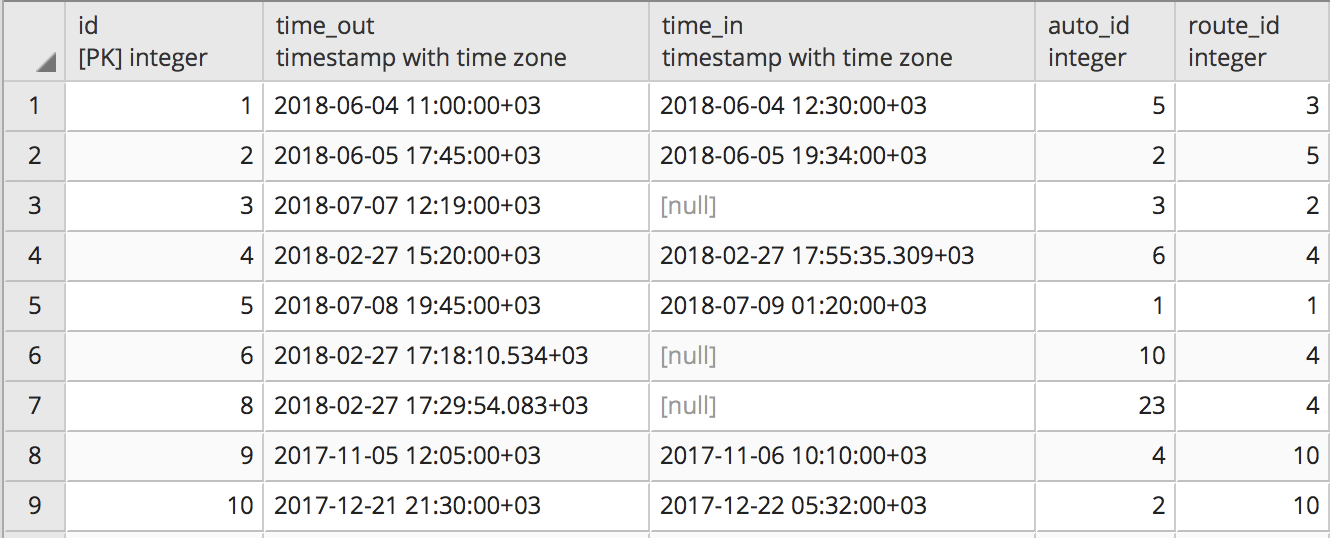


Таблица журнал



Удаляем маршрут с id = 10

**DELETE FROM routes USING journal**

**WHERE routes.id = journal.route\_id AND routes.id = 10**

Таблица маршрутов

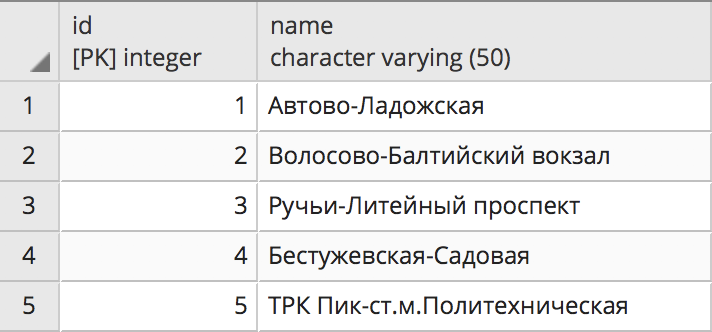
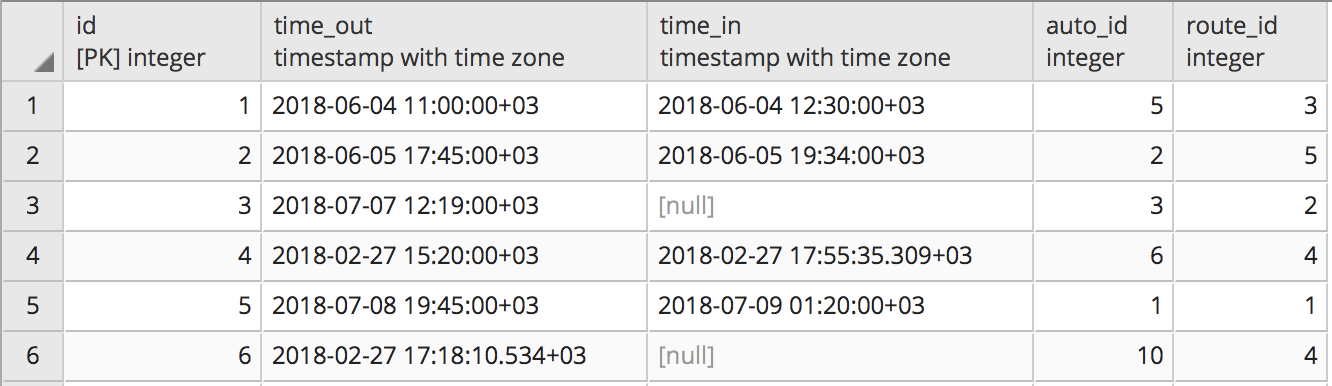


Таблица журнал



* 1. Удалить персонал, не имеющий автомобилей

Таблица водители

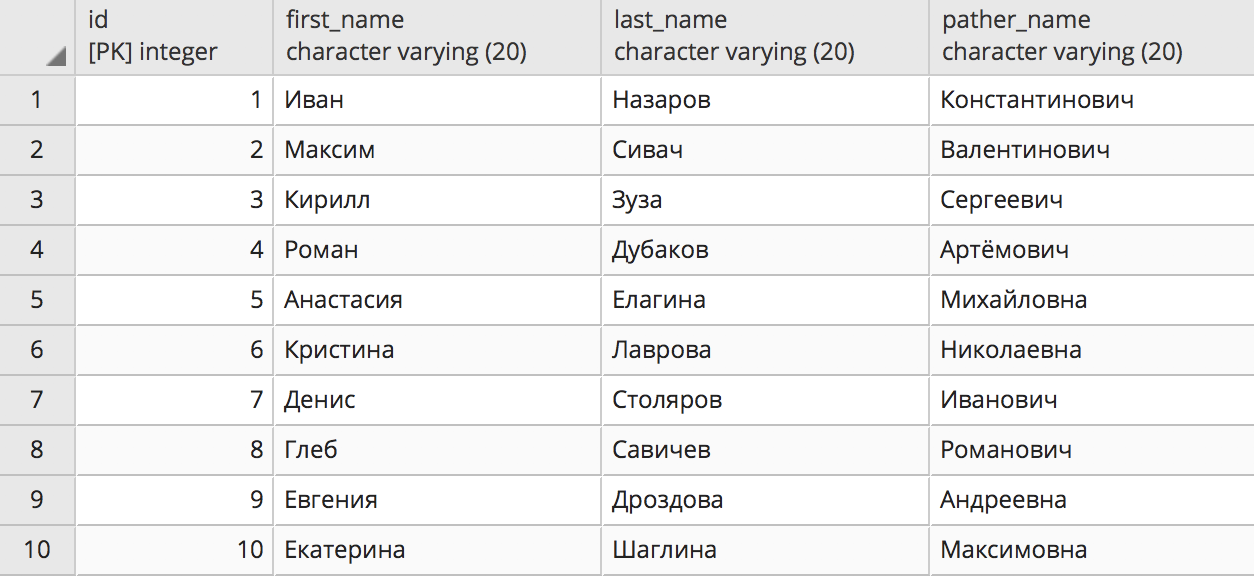
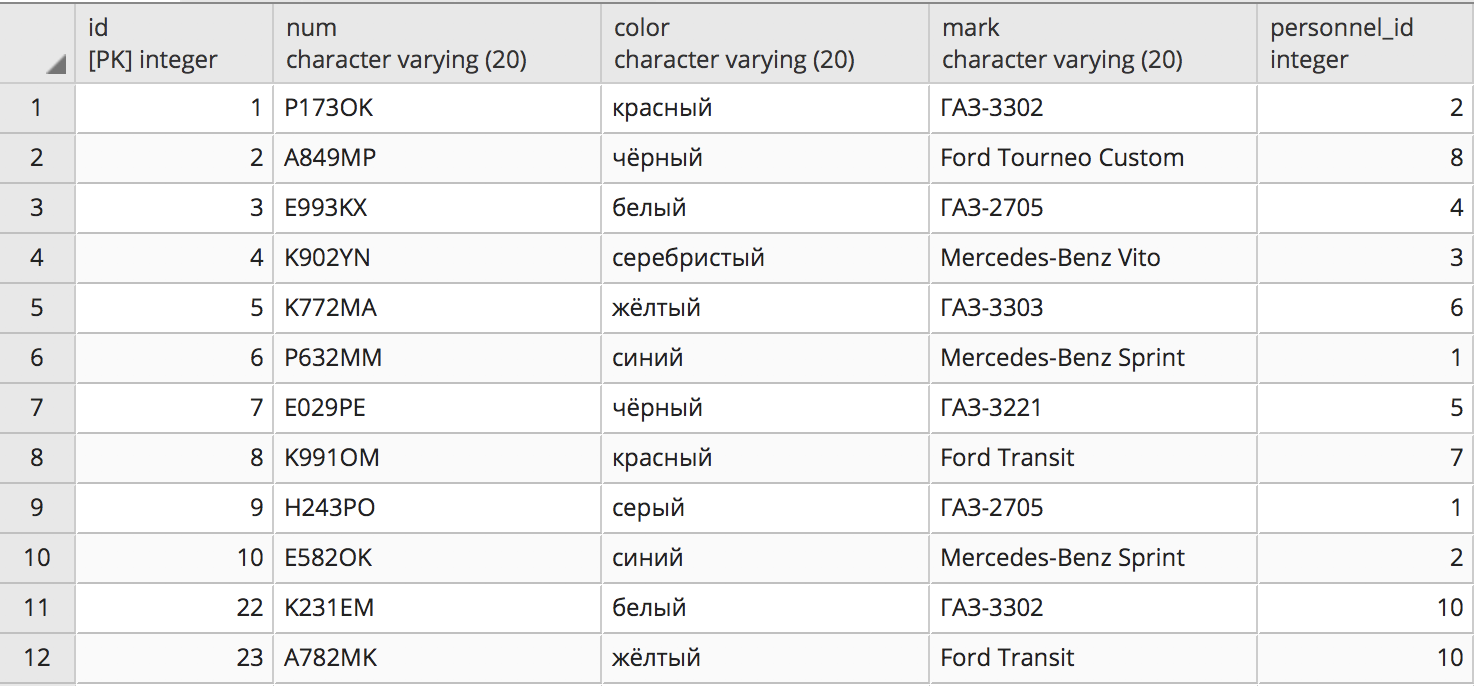


Таблица автомобили



**DELETE FROM auto\_personnel**

**WHERE id NOT IN (**

**SELECT auto.personnel\_id**

**FROM auto, auto\_personnel**

**WHERE auto\_personnel.id = auto.personnel\_id**

**);**

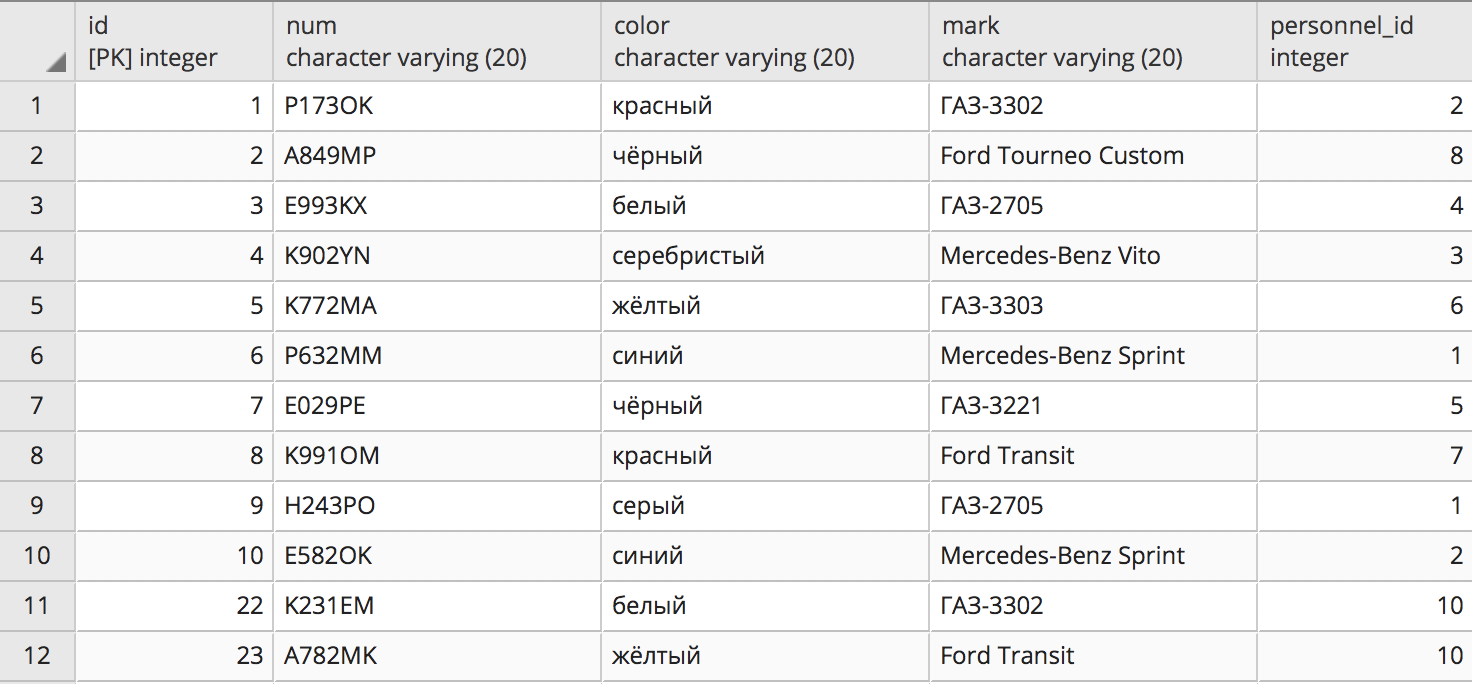


* удаление в рамках транзакции
  1. Удалить в рамках транзакции заданного водителя и его автомобили

Таблица водители



Таблица автомобили



Удаляем водителя с id = 10

**START TRANSACTION;**

**DELETE FROM auto\_personnel WHERE id = 10;**

**DELETE FROM auto WHERE personnel\_id = 10;**

**COMMIT;**

ИЛИ

**START TRANSACTION;**

**DELETE FROM auto\_personnel USING auto**

**WHERE auto\_personnel.id = auto.personnel\_id AND auto\_personnel.id = 10;**

**COMMIT;**

Таблица водители



Таблица автомобили



* 1. то же, что и п1, но транзакцию откатить

**BEGIN TRANSACTION;**

**DELETE FROM auto\_personnel WHERE id = 10;**

**DELETE FROM auto WHERE personnel\_id = 10;**

**ROLLBACK;**

ИЛИ

**BEGIN TRANSACTION;**

**DELETE FROM auto\_personnel USING auto**

**WHERE auto\_personnel.id = auto.personnel\_id AND auto\_personnel.id = 10;**

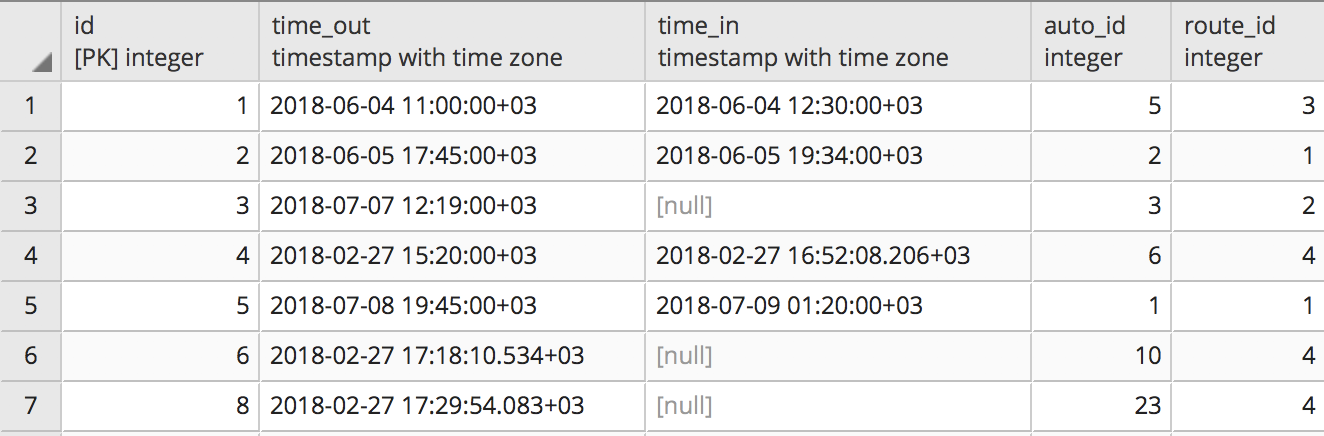
**ROLLBACK;**

Hints: delete, where, in, exists, commit, rollback

### Модификация данных

* модификация по фильтру
  1. Изменить время прибытия у заданной строки журнала оператора

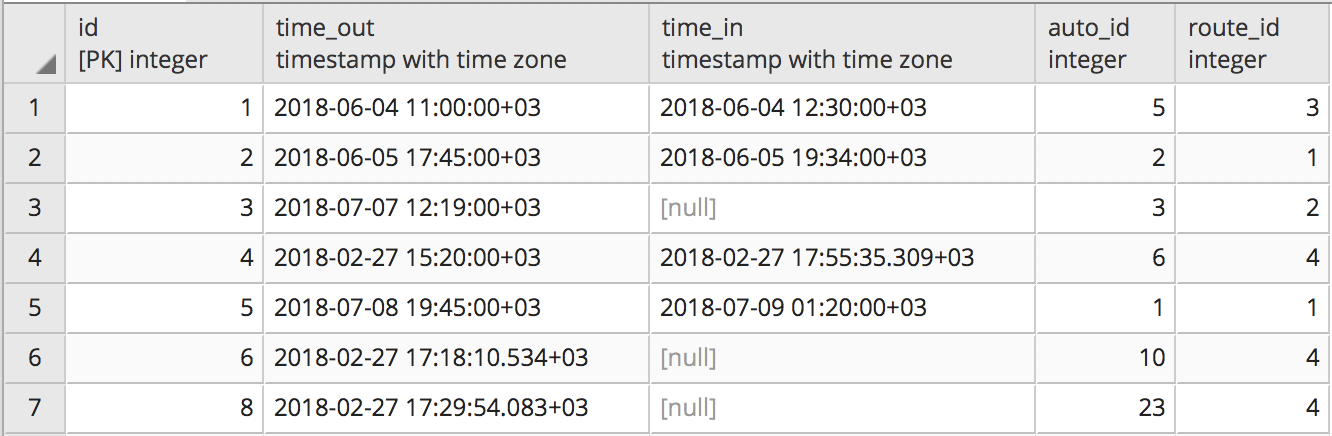
Таблица журнал



**UPDATE journal**

**SET time\_in = 'now'**

**WHERE id = 4**



* 1. Изменить маршрут, по которому следует автомобиль с заданным номером

Таблица автомобили

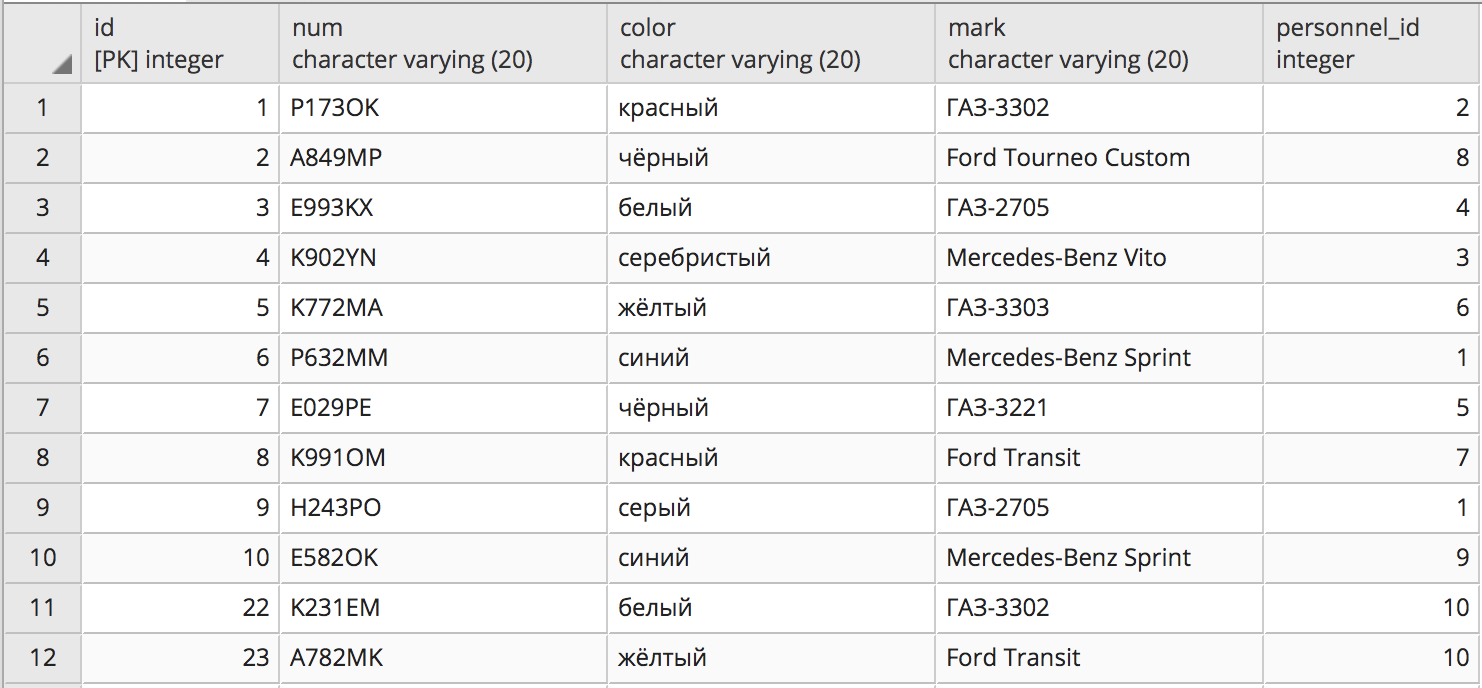
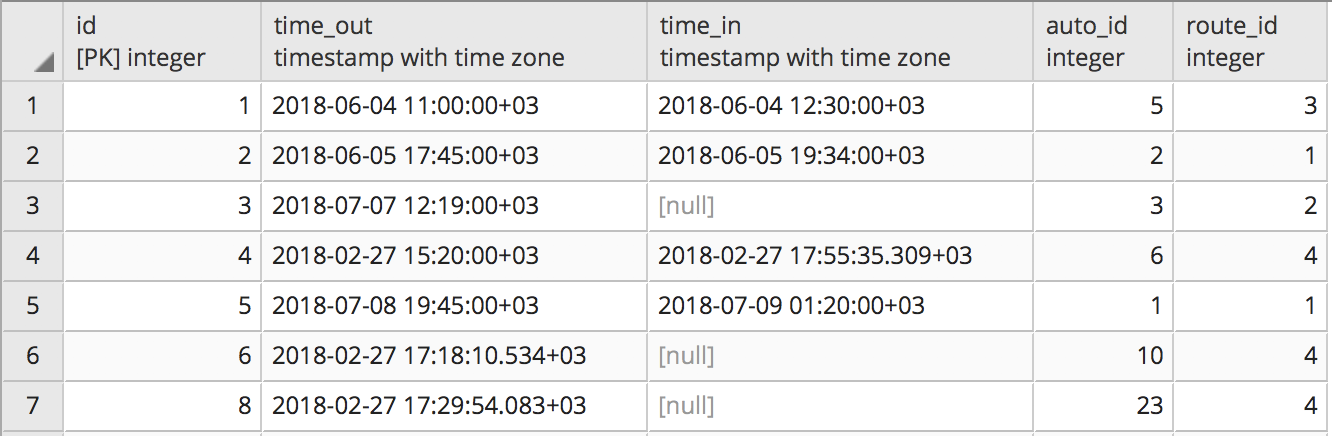


Таблица журнал



**UPDATE journal**

**SET route\_id = 5**

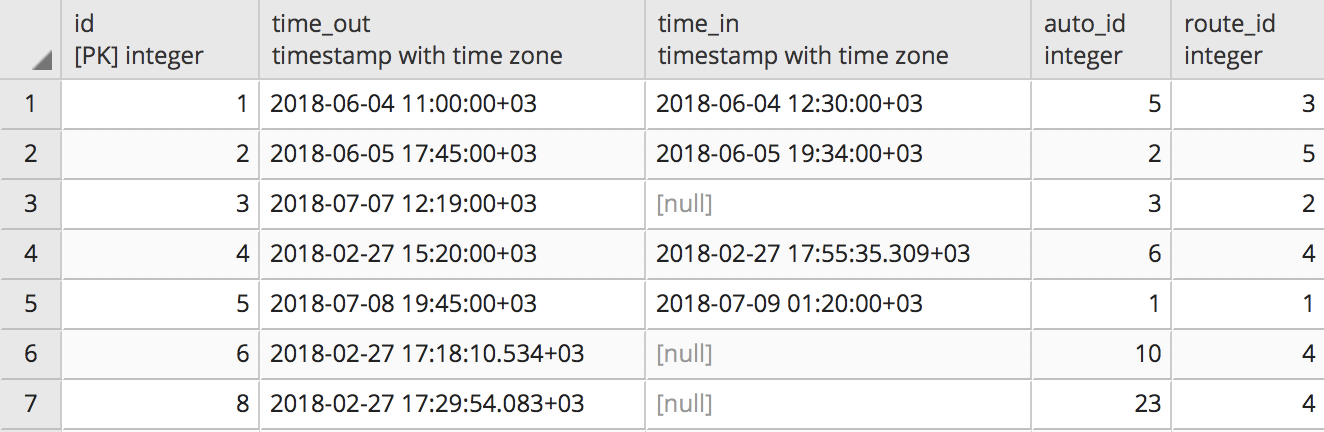
**WHERE auto\_id = (**

**SELECT id**

**FROM auto**

**WHERE num = 'A849MP'**

**);**



* модификация в рамках транзакции
  1. В рамках транзакции поменять заданный маршрут всех поездок в журнале на другой и удалить заданный маршрут.

Таблица маршрутов

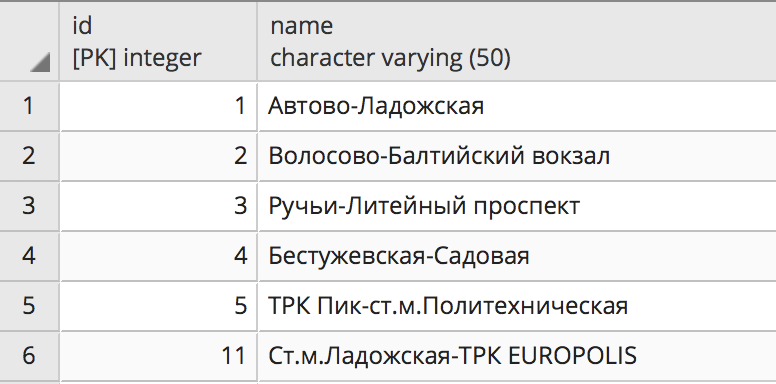
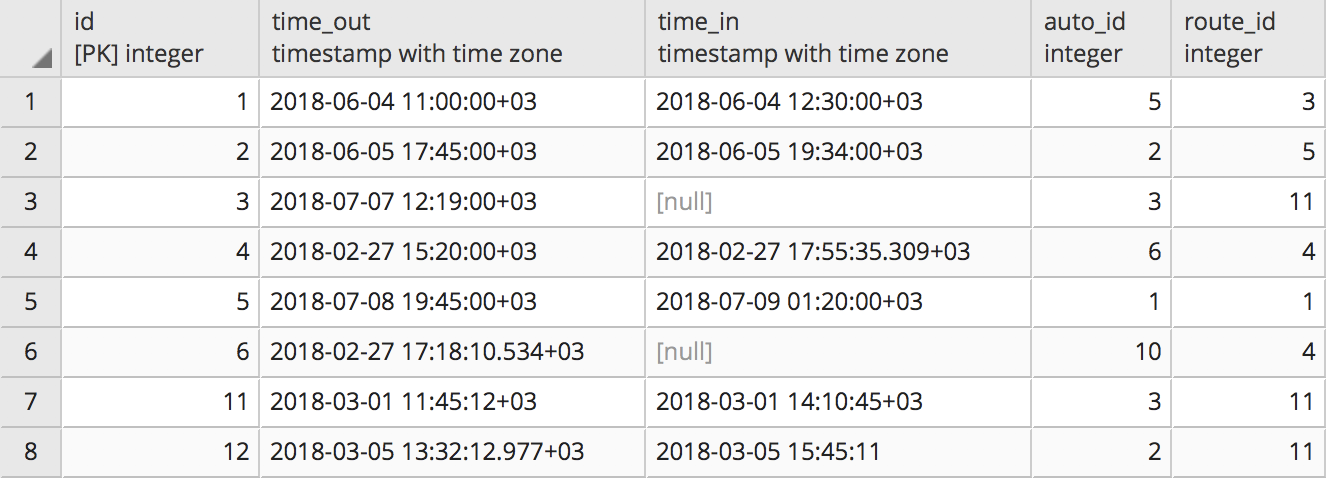


Таблица журнал



Маршрут с id = 11

**START TRANSACTION;**

**UPDATE journal SET route\_id = 1 WHERE route\_id = 11;**

**DELETE FROM routes WHERE id = 11;**

**COMMIT;**

ИЛИ

**START TRANSACTION;**

**DO $$**

**DECLARE**

**oldid INTEGER := 11;**

**newid INTEGER := 1;**

**BEGIN**

**UPDATE journal SET route\_id = newid WHERE route\_id = oldid;**

**DELETE FROM routes WHERE id = oldid;**

**END$$;**

**COMMIT;**

Таблица маршрутов

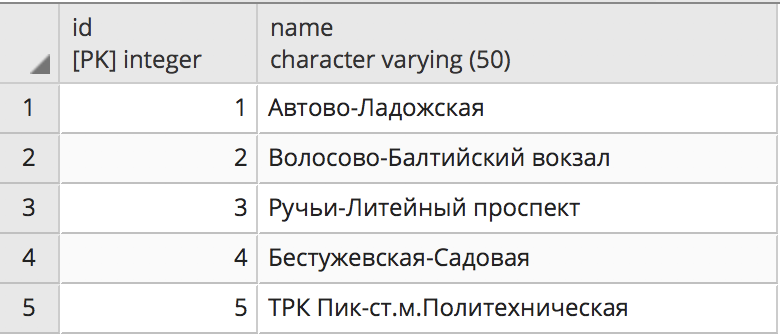
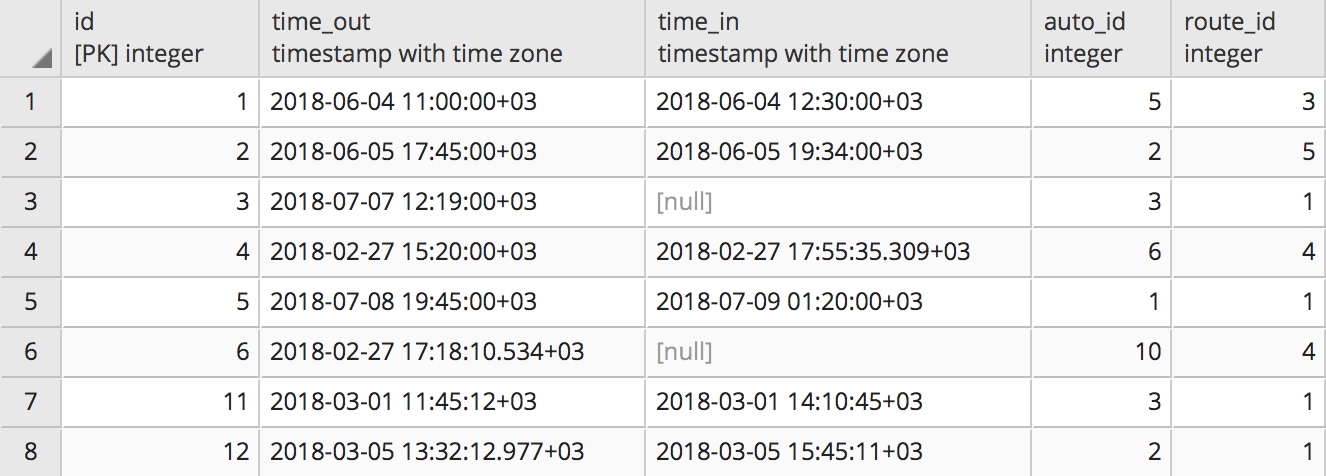


Таблица журнал



* 1. то же, что и п1, но транзакцию откатить

**BEGIN TRANSACTION;**

**DO $$**

**DECLARE**

**oldid INTEGER := 11;**

**newid INTEGER := 1;**

**BEGIN**

**UPDATE journal SET route\_id = newid WHERE route\_id = oldid;**

**DELETE FROM routes WHERE id = oldid;**

**END$$;**

**ROLLBACK;**