**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

(СПбГУТ)

ФАКУЛЬТЕТ ИНФОРМАЦИОНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ **(ИТПИ)**

Кафедра программной инженерии и вычислительной техники **(ПИиВТ)**

Дисциплина: «Базы данных»

Практическое задание №1.

**Тема: «Учет работы сотрудников страхового общества. Добавление триггеров»**

Вариант 24.

Выполнил:

  Студент группы ИКПИ-23

Харлова А.А.

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Принял:

Дятлов Д.А.

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_

2024 г.

**Задание**

К базе данных, созданной в индивидальном задании, добавить триггеры.

Описание задания индивидульного задания приведено ниже.

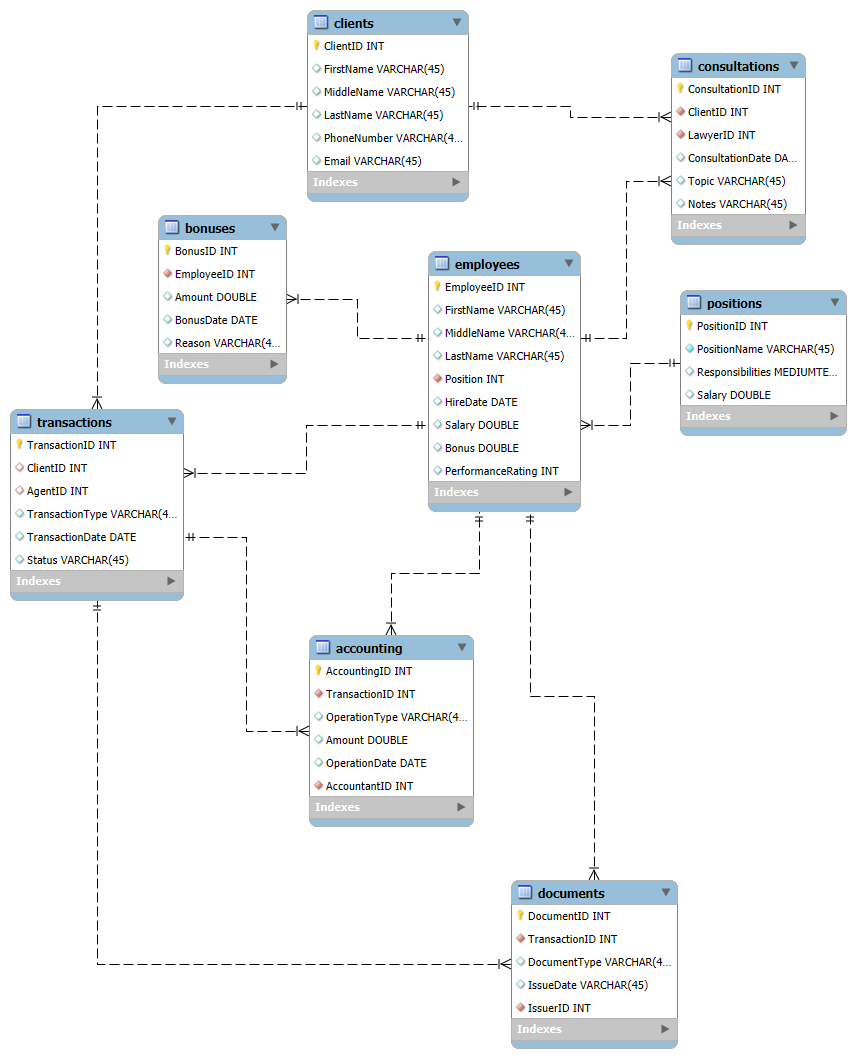
Фирма производит операции по купле-продаже и аренде, оформлению документов и консультации юристов.

Система учеты работы и поощрения сотрудников предусматривает оценку деятельности сотрудников и в зависимости от ее успешности устанавливает объем поощрений.

Основная работа выполняется агентами, которые являются посредниками между продавцами и покупателями (акредиторами).

Деятельностью фирмы руководит директор, который оформляет все необходимые документы, бухгалтер — ведет бухгалтерский учет в фирме, а юрист дает консультации по вопросам, касающимся деятельности фирмы.

**EER-диаграмма**

****

**Создание триггеров**

1. **Триггеры для таблицы *Bonuses***

Изменение поля *Bonus* таблицы *Employees*. Для реализации триггеров, которые будут автоматически пересчитывать итоговую сумму всех бонусов сотрудника в таблице *Employees* при любом изменении таблицы *Bonuses* (вставка, обновление или удаление), можно создать три триггера: AFTER INSERT, AFTER UPDATE и AFTER DELETE.

**after\_bonus\_insert**

Этот триггер будет срабатывать после вставки новой записи в таблицу *Bonuses* и увеличивать общую сумму бонусов сотрудника в таблице *Employees*.

DELIMITER //

CREATE TRIGGER after\_bonus\_insert

AFTER INSERT ON bonuses

FOR EACH ROW

BEGIN

UPDATE employees

SET Bonus = Bonus + NEW.Amount

WHERE EmployeeID = NEW.EmployeeID;

END //

DELIMITER ;

**after\_bonus\_update**

Этот триггер будет срабатывать после обновления записи в таблице *Bonuses* и корректировать общую сумму бонусов сотрудника в таблице *Employees*.

DELIMITER //

CREATE TRIGGER after\_bonus\_update

AFTER UPDATE ON bonuses

FOR EACH ROW

BEGIN

-- Уменьшение суммы бонусов на старое значение

UPDATE employees

SET Bonus = Bonus - OLD.Amount

WHERE EmployeeID = OLD.EmployeeID;

-- Увеличение суммы бонусов на новое значение

UPDATE employees

SET Bonus = Bonus + NEW.Amount

WHERE EmployeeID = NEW.EmployeeID;

END //

DELIMITER ;

**after\_bonus\_delete**

Этот триггер будет срабатывать после удаления записи из таблицы *Bonuses* и уменьшать общую сумму бонусов сотрудника в таблице *Employees*.

DELIMITER //

CREATE TRIGGER after\_bonus\_delete

AFTER DELETE ON bonuses

FOR EACH ROW

BEGIN

-- Уменьшение суммы бонусов на удаленное значение

UPDATE employees

SET Bonus = Bonus - OLD.Amount

WHERE EmployeeID = OLD.EmployeeID;

END //

DELIMITER ;

1. **Триггер для таблицы *transactions***

Установка статуса "Отменено" для транзакции, если она отменена.Этот триггер устанавливает статус "Отменено" в таблице *accounting* для всех операций, связанных с транзакцией, если транзакция отменена.

DELIMITER //

CREATE TRIGGER before\_transaction\_update

BEFORE UPDATE ON transactions

FOR EACH ROW

BEGIN

IF NEW.Status = 'Отменено' THEN

UPDATE accounting

SET OperationType = 'Отменено'

WHERE TransactionID = NEW.TransactionID;

END IF;

END //

DELIMITER ;

1. **Триггер для таблицы *consultations***

Проверка, что консультации могут проводить только юристы. Этот триггер проверяет, что консультации могут проводить только сотрудники с должностью "Юрист" (PositionID = 3). Если это не так, выбрасывается ошибка.

DELIMITER //

CREATE TRIGGER before\_consultation\_insert

BEFORE INSERT ON consultations

FOR EACH ROW

BEGIN

DECLARE employee\_position INT;

SELECT Position INTO employee\_position FROM employees WHERE EmployeeID = NEW.LawyerID;

IF employee\_position != 3 THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Только юристы могут проводить консультации';

END IF;

END //

DELIMITER ;

1. **Триггер для таблицы *transactions***

Проверка, что сделками занимаются только агенты. Этот триггер проверяет, что сделками занимаются только сотрудники с должностью "Агент" (PositionID = 4). Если это не так, выбрасывается ошибка.

DELIMITER //

CREATE TRIGGER before\_transaction\_insert

BEFORE INSERT ON transactions

FOR EACH ROW

BEGIN

DECLARE employee\_position INT;

SELECT Position INTO employee\_position FROM employees WHERE EmployeeID = NEW.AgentID;

IF employee\_position != 4 THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Только агенты могут заниматься сделками';

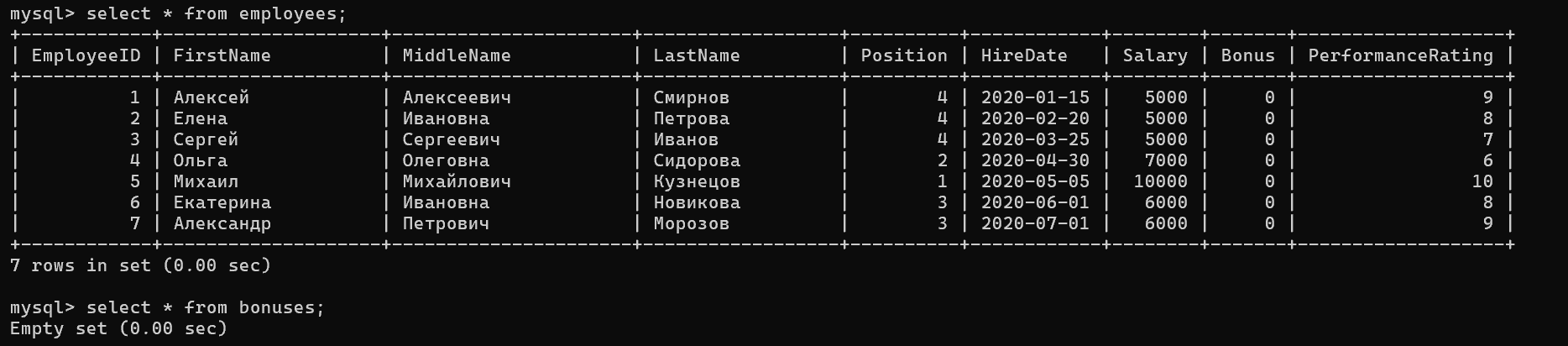
END IF;

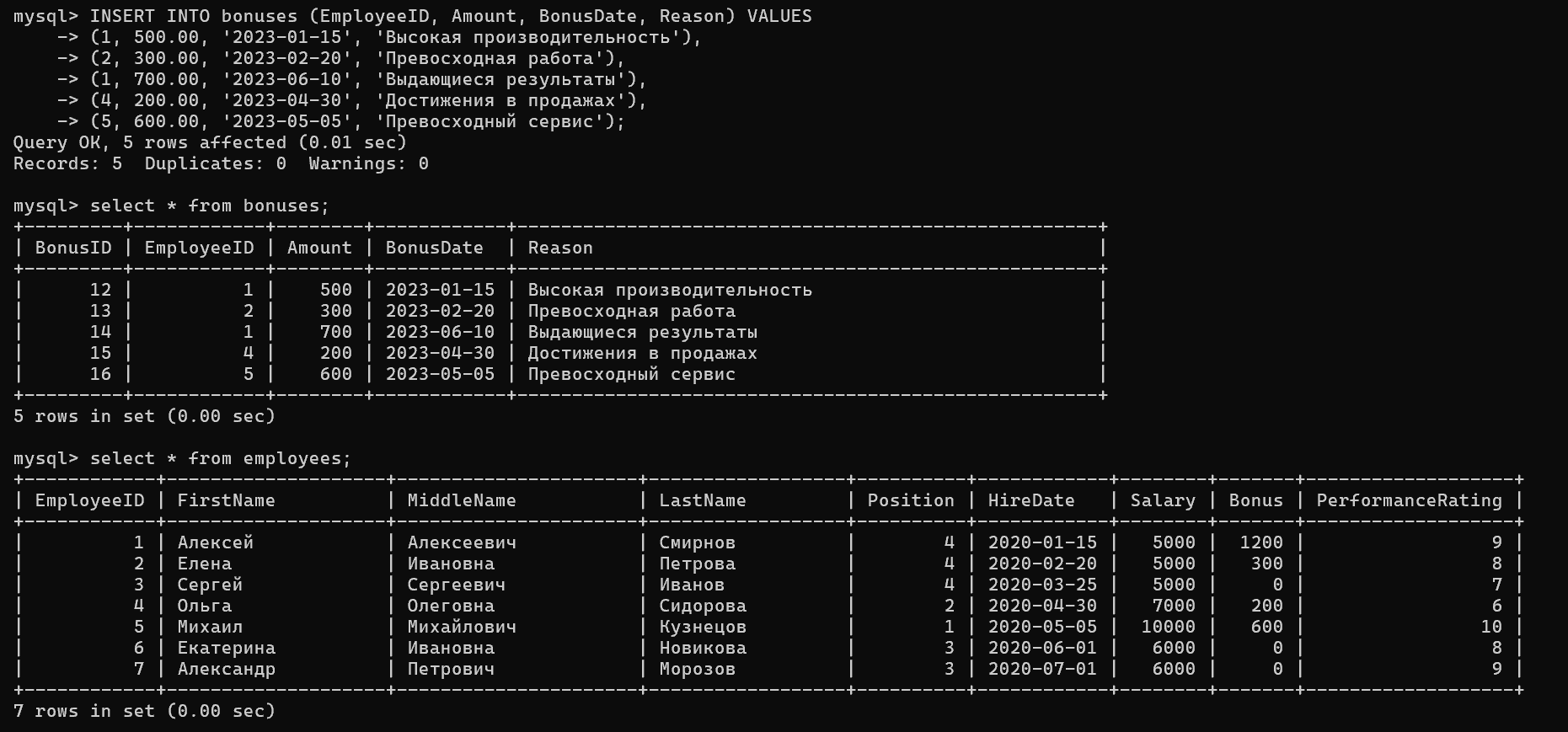
END //

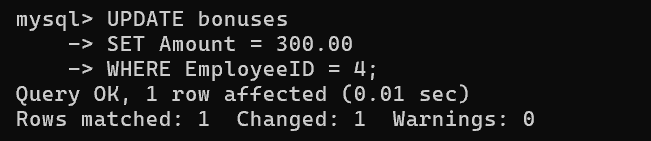
DELIMITER ;

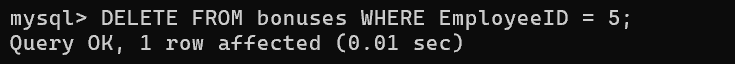
**Пример работы**

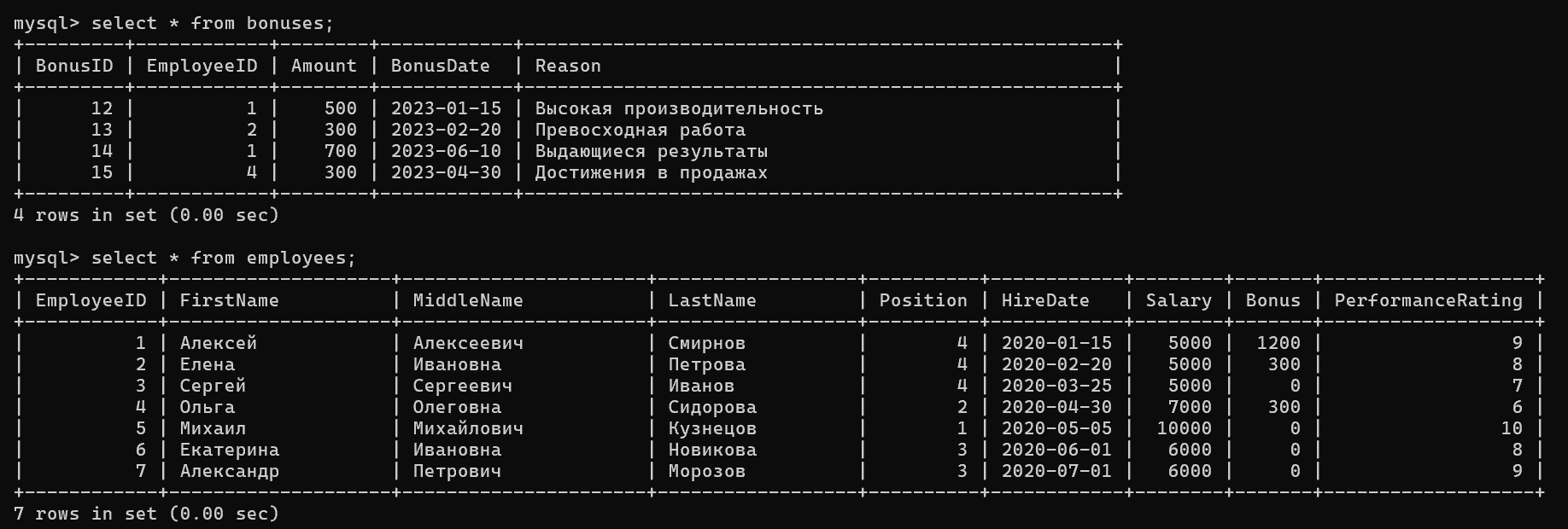
1. Триггеры для таблицы *bonuses*



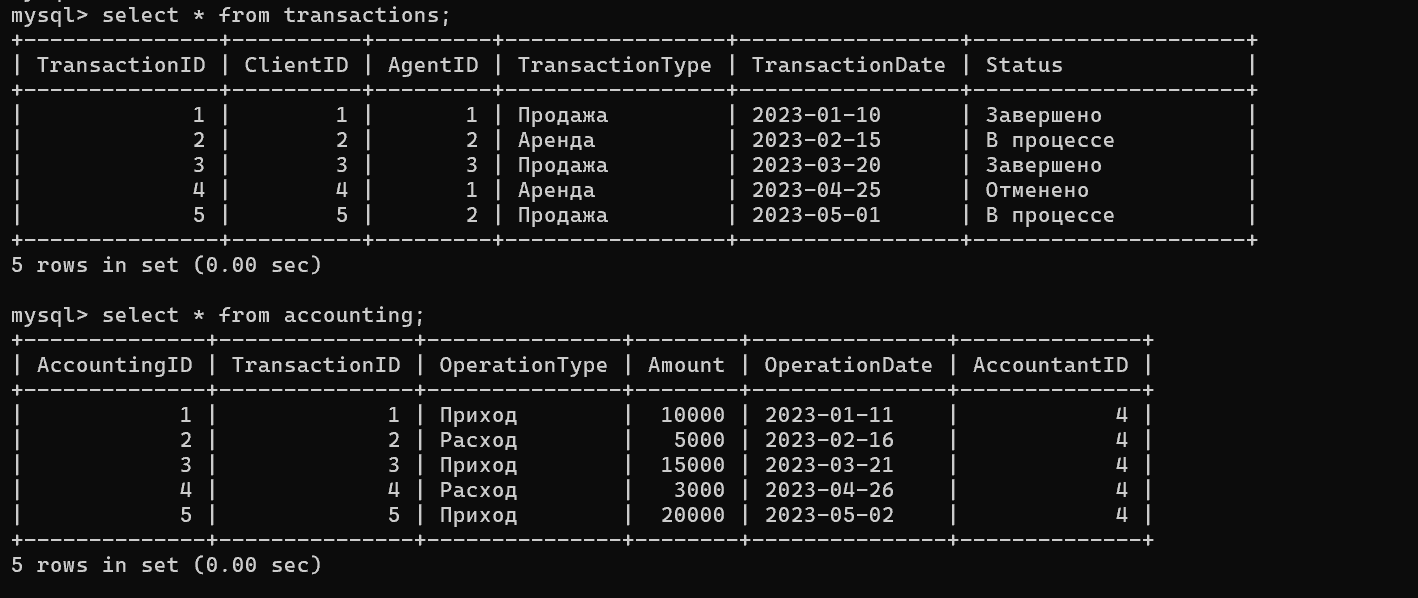


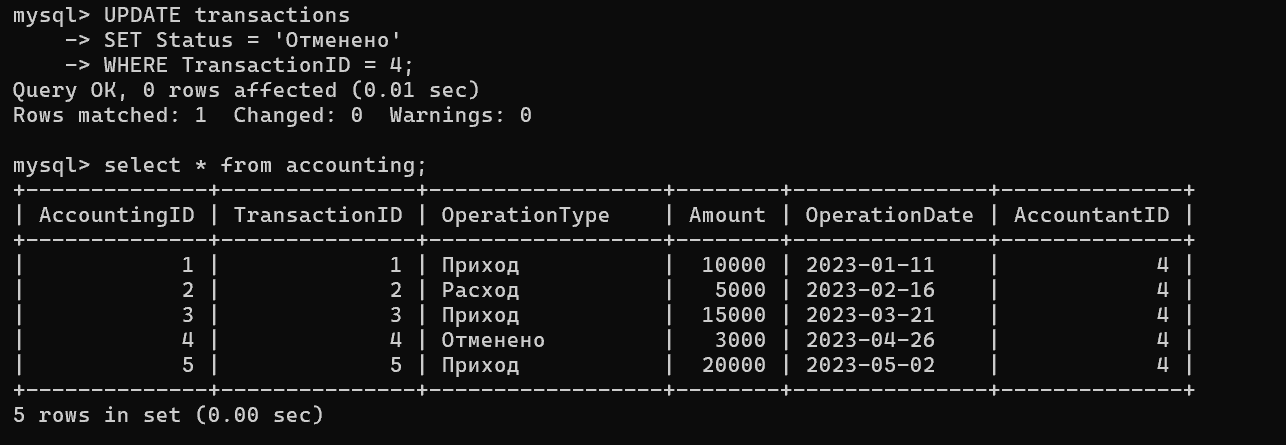




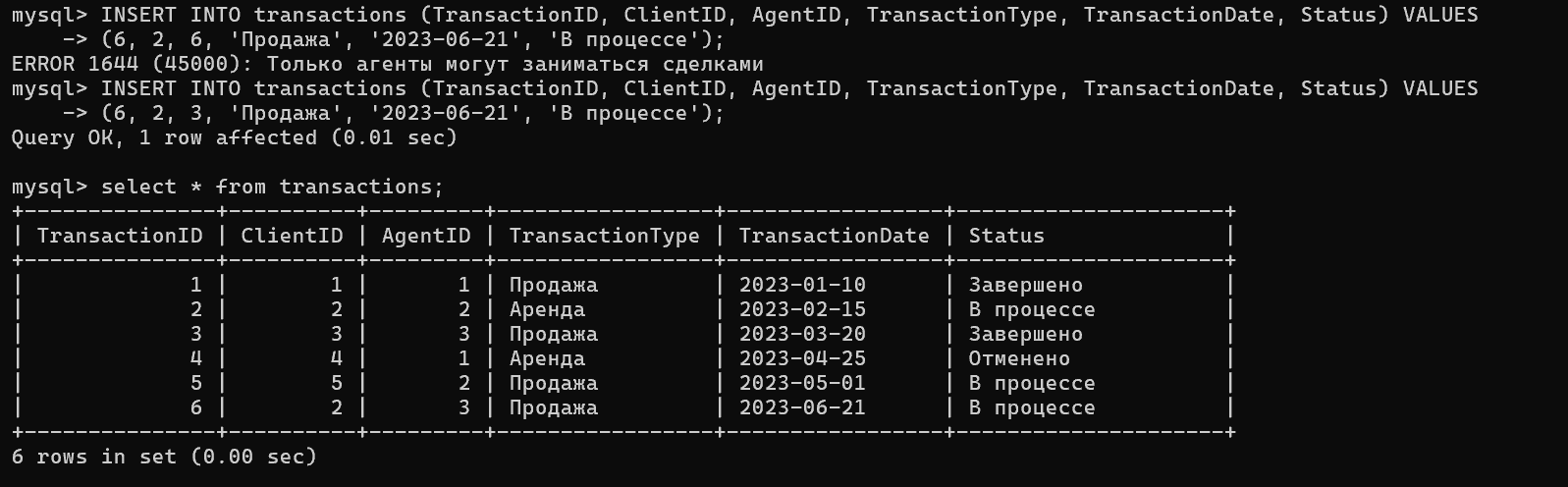


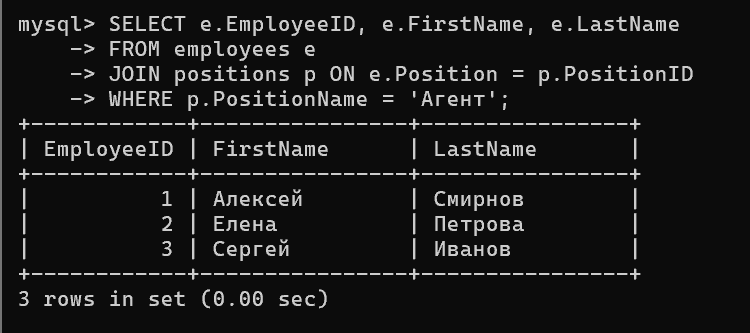
1. Триггеры для таблицы *transactions*





Команда для вставки новой транзакции с неправильным *AgentID*:





1. Триггеры для таблицы *consultations*

Команда для вставки новой транзакции с неправильным *LawerID*:

