Министерство науки и высшего образования РФ

Федеральное государственное автономное образовательное учреждение

высшего образования

«Московский политехнический университет»

(Московский политех)

Домашняя работа по курсу «Проектирование и администрирование баз данных»

Практическое задание 10



Выполнил:

Студент группы 221-352

Барателия Т.А.

Проверил преподаватель: Тимакин О. А.

Москва 2023г.

Процедура 1: Получение сотрудников по должности

Описание: Данная процедура принимает на вход параметр - должность, и выводит список фамилий сотрудников, занимающих указанную должность. Для этого происходит поиск в таблице k\_staff по значению параметра и вывод соответствующих фамилий.

DELIMITER //

CREATE PROCEDURE GetEmployeesByPosition(IN position VARCHAR(45))

BEGIN

SELECT staff\_name FROM k\_staff WHERE staff\_post = position;

END //

DELIMITER ;

Процедура 2: Информация о платежах по номеру счета

Описание: Данная процедура принимает на вход параметр - номер счета, и выводит информацию о платежах, внесенных по указанному счету. Сначала происходит поиск в таблице k\_bill по значению параметра для получения соответствующего счета, а затем извлекаются связанные счету платежи из таблицы k\_payment. Информация о номере, дате и сумме платежей выводится на экран.

DELIMITER //

CREATE PROCEDURE GetPaymentsByBillNumber(IN billNumber INT)

BEGIN

SELECT payment\_num, payment\_date, payment\_sum

FROM k\_payment

WHERE k\_bill\_bill\_num = billNumber;

END //

DELIMITER ;

Процедура 3: Уценка самого дорогого товара

Описание: Данная процедура выполняет поиск самого дорогого товара в таблице k\_price и устанавливает его новую цену с учетом заданной в процедуре величины уценки в процентах. Сначала происходит поиск товара с наибольшей ценой в таблице k\_price, затем его цена пересчитывается с учетом указанного процента уценки и обновляется в таблице.

DELIMITER //

CREATE PROCEDURE DiscountMostExpensiveItem(IN discountPercent DECIMAL(5,2))

BEGIN

UPDATE k\_price

SET price\_sum = price\_sum \* (1 - discountPercent / 100)

WHERE price\_sum = (

SELECT MAX(price\_sum) FROM k\_price

);

END //

DELIMITER ;

Процедура 4: Назначение нового начальника отдела

Описание: Данная процедура принимает на вход два параметра - отдел и начальника. Она изменяет запись в таблице k\_dept, указывая нового начальника для указанного отдела. Процедура проверяет, существует ли отдел в таблице k\_dept, затем находит соответствующего сотрудника в таблице k\_staff по фамилии начальника и обновляет запись в таблице k\_dept с новым значением начальника.

DELIMITER //

CREATE PROCEDURE AssignDepartmentHead(IN departmentNumber INT, IN staffNumber INT)

BEGIN

UPDATE k\_dept

SET k\_staff\_staff\_num = staffNumber

WHERE dept\_num = departmentNumber;

END //

DELIMITER ;

Процедура 5: Перевод сотрудника на другую должность

Описание: Данная процедура принимает на вход два параметра - фамилию сотрудника и новую должность. Она изменяет запись в таблице k\_staff, указывая новую должность для указанного сотрудника. Процедура проверяет, существует ли сотрудник с указанной фамилией в таблице k\_staff, затем обновляет запись с новым значением должности.

DELIMITER //

CREATE PROCEDURE ChangeEmployeePosition(IN employeeSurname VARCHAR(45), IN newPosition VARCHAR(45))

BEGIN

UPDATE k\_staff

SET staff\_post = newPosition

WHERE staff\_name = employeeSurname;

END //

DELIMITER ;

Процедура 6: Удаление отдела и перенос сотрудников

Описание: Данная процедура принимает на вход параметр - отдел, и выполняет удаление указанного отдела из таблицы k\_dept. Перед удалением отдела, процедура выполняет перенос всех сотрудников из этого отдела в другой отдел.

DELIMITER //

CREATE PROCEDURE ChangeDepartmentName(IN oldName VARCHAR(45), IN newName VARCHAR(45))

BEGIN

UPDATE k\_dept

SET dept\_full\_name = newName

WHERE dept\_full\_name = oldName;

END //

DELIMITER ;

Процедура 7: Начисление пени

Описание: Данная процедура предназначена для начисления пени на просроченные платежи. Она принимает на вход параметр - количество дней просрочки и производит расчет и добавление пени в таблицу k\_penalty для каждого просроченного платежа.

DELIMITER //

CREATE PROCEDURE ApplyPenalty()

BEGIN

UPDATE k\_bill

SET bill\_peni = bill\_sum \* 0.1

WHERE bill\_term < CURDATE() AND bill\_peni = 0;

END //

DELIMITER ;

Вызов всех процедур

CALL GetEmployeesByPosition('Менеджер');

CALL GetPaymentsByBillNumber(12345);

CALL DiscountMostExpensiveItem(20);

CALL AssignDepartmentHead(1, 2);

CALL ChangeEmployeePosition('Иванов', 'Аналитик');

CALL ChangeDepartmentName('Старый отдел', 'Новый отдел');

CALL ApplyPenalty();