

Лабораторные работы по курсу "Базы данных" (4-й модуль 2-го курса).

ВНИМАТЕЛЬНО ПРОЧИТАЙТЕ ЗАДАНИЕ!

Лабораторные работы заключаются в создании процедурных объектов БД: двух функций и двух триггеров на языке PL/pgSQL в СУБД PostgreSQL (версии не ниже 9.5).

Сдача лабораторной работы заключается в демонстрации работы созданных процедурных объектов и ответах на вопросы по сути работы.

Отчет по лабораторной работе должен включать ФИО студента, номер группы, название предмета, номер варианта, задание, текст создания процедурных объектов и скриншоты, демонстрирующие их работу. Для триггера нужно вызвать соответствующую команду, которая является событием триггера и вызывает его срабатывание, и команду, подтверждающую выполнение действий триггера (выбор данных из архивной таблицы, например).

Для выполнения лабораторных работ необходимо установить СУБД PostgreSQL, которую можно скачать с официального сайта <https://www.postgresql.org>.

Задания выполняются по тому же варианту, который был выдан в начале занятий.

Перед созданием процедурных объектов необходимо создать таблицы, запустив SQL-код, который был написан при выполнении первой лабораторной работы (если эту работу вы делали под управлением другой СУБД).

1. Лабораторная работа №6 заключается в создании функций.
2. Лабораторная работа №7 заключается в создании триггеров.

Задания на лабораторные работы

Вариант 1. БД отдела кадров института.

Л. р. №6. Создание функций.

1. Функция, определяющая по дате рождения, является ли человек юбиляром в текущем месяце, и выдающая для юбиляра возраст (юбилейную дату, например, "45"), а в противном случае –NULL. Параметр – дата рождения.
2. Функция, вычисляющая премии для сотрудников, празднующих юбилей в текущем месяце. Входные параметры – базовый размер премии. Размер премии для сотрудника рассчитывать по следующей формуле: за каждые 100 часов лекций нагрузки – один базовый размер премии для доктора наук, 0.8 для доцента и 0.5 для всех остальных. Результат работы функции добавляется в таблицу «Премии» (поля ФИО сотрудника, дату юбилея, возраст и размер премии), которая в начале работы очищается от старых данных. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Преподаватели", для которых могут быть определены домены (в частности, значения полей "Пол", "Должность", "Ученая степень", "Ученое звание", возраст не может быть меньше 20 лет).
2. Регистрация изменений, вносимых в таблицу "Преподаватели" (дублирование старой записи в специальной таблице с указанием даты изменения и пользователя, который их проводит).

Вариант 2. БД бухгалтерии.

Л. р. №6. Создание функций.

1. Функция, определяющая стаж работы человека. Параметры: стаж на прежних местах работы и дата поступления на работу. Возвращает число полных лет стажа на текущую дату.
2. Функция начисления премий сотрудникам. Входной параметр – базовая ставка. Размер премии вычисляется как произведение базовой ставки на оклад и на коэффициент, который зависит от стажа работы на данном предприятии:
 - 0.1, если стаж от 1 до 5 лет;
 - 0.2, если стаж от 5 до 10 лет;
 - 0.3, если стаж от 10 до 20 лет;
 - 0.5, если стаж от 20 до 30 лет;
 - 1, если стаж свыше 30 лет.

Результат работы функции добавляется в таблицу «Премии» (поля ФИО сотрудника, стаж, размер премии), которая в начале работы очищается от старых данных. Людям, проработавшим менее 1 года, премия не выплачивается (в таблицу они не добавляются). Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Сотрудники", для которых могут быть определены домены (в т.ч., ИНН может содержать только цифры, (возраст сотрудника)-(стаж на прежних работах)-(стаж работы на данном предприятии) не может быть меньше 16 лет, а дата поступления на работу должна быть не больше текущей даты). Если при вводе данных дата поступления не указана, устанавливать текущую дату.
2. Установка значения поля "пол", если оно не установлено. Правила: если отчество оканчивается на '-НА', то пол женский, если на '-ИЧ', то мужской. В противном случае триггер должен генерировать ошибку.

Вариант 3. БД бухгалтерии.

Л. р. №6. Создание функций.

1. Функция, определяющая, является ли человек на текущую дату пенсионером. Параметры: пол и дата рождения. Возвращает строку "пенсионер" или пустую строку.
2. Функция начисления премий сотрудникам. Входной параметр – базовая ставка. Размер премии вычисляется как произведение базовой ставки на оклад и на коэффициент, который зависит от стажа работы на данном предприятии:
 - 0.1, если стаж от 1 до 5 лет;
 - 0.2, если стаж от 5 до 10 лет;
 - 0.3, если стаж от 10 до 20 лет;
 - 0.5, если стаж от 20 до 30 лет;
 - 1, если стаж свыше 30 лет.

Результат работы функции добавляется в таблицу «Премии» (поля ФИО сотрудника, стаж, размер премии), которая в начале работы очищается от старых данных. Стаж рассчитывается с учетом трудовой книжки. Людям, проработавшим менее 1 года, премия не выплачивается (и в таблицу они не добавляются). Пенсионерам выплачивается половина премии. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Сотрудники", для которых могут быть определены домены (в т.ч., ИНН может содержать только цифры, а дата поступления на работу должна быть не больше текущей даты). Если при вводе данных дата поступления не указана, устанавливать текущую дату.
2. Формирование таблицы "Трудовая книжка": при изменении занимаемой должности в таблице "Сотрудники", предыдущая должность добавляется в таблицу "Трудовая

книжка" с указанием даты начала работы в этой должности (старое значение даты вступления в должность из таблицы "Сотрудники").

Вариант 4. БД спортивного клуба.

Л. р. №6. Создание функций.

1. Функция, определяющая по дате рождения и текущей дате, к какой категории относится спортсмен:

Д-1	Дети I	9 лет и моложе
Д-2	Дети II	10-11 лет
Ю-1	Юниоры I	12-13 лет
Ю-2	Юниоры II	14-15 лет
М	Молодежь	16-18 лет
ВЗ	Взрослые	19 лет и старше
С	Сеньоры	35 лет и старше
2. Функция, проверяющая соответствие места и завоеванных очков на соревновании. Параметр: идентификатор соревнования. Если два спортсмена получили одинаковое количество очков на одном соревновании, давать обоим одно и то же место.

Л. р. №7. Создание триггеров.

1. Увеличение рейтинга спортсмена при добавлении сведений о его участии в соревнованиях: за 1-е место – плюс 20 баллов к рейтингу, за 2-е место – плюс 15, за 3-е место – плюс 10, за простое участие – плюс 2 балла.
2. Проверка значений всех полей отношения "Спортсмены", для которых могут быть определены домены, например: номер домашнего телефона не должен совпадать с номером мобильного телефона; текущий рейтинг и стипендия не могут быть отрицательными; спортсмен не может быть моложе пяти лет.

Вариант 5. БД диссертаций.

Л. р. №6. Создание функций.

1. Функция, выдающая полное название ученой степени по параметрам "Тип" и "Раздел науки". Например, для типа "докторская" и раздела "Технические науки" функция должна вернуть "доктор технических наук".
2. Функция, выдающая для автора его ученую степень. Параметр – идентификатор автора. Если автор защитил кандидатскую и докторскую диссертации по одному разделу науки, то он является доктором наук. Если разделы разные, то ученые степени перечисляются через запятую (например, 'кандидат экономических наук, доктор технических наук'). Сначала – кандидат, потом доктор. Докторских степеней может быть несколько, кандидатская – одна. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Диссертации", для которых могут быть определены домены (дата защиты и дата утверждения не могут быть больше текущей даты; дата защиты меньше даты утверждения, если последняя определена; тип диссертации – кандидатская или докторская).
2. Триггер, переносящий в архив (в специальную таблицу) предыдущие значения данных об авторах при их изменении: могут поменяться любые сведения, кроме даты рождения. При попытке изменить дату рождения триггер выдает ошибку.

Вариант 6. БД научного фонда.

Л. р. №6. Создание функций.

1. Функция, принимающая в качестве параметра даты начала и завершения гранта и возвращающая строку "не начался", если текущая дата меньше даты начала гранта, строку "закончился", если текущая дата больше даты завершения, и пустую строку в других случаях.
2. Функция, возвращающая для руководителя гранта количество грантов, в которых он участвует параллельно с текущим грантом. Параметры: идентификатор руководителя и номер текущего гранта. Параллельным с текущим грантом считается тот грант, период выполнения которого совпадает хотя бы частично с периодом выполнения текущего гранта. Например, периоды 01.02.2019-30.10.2020 и 01.01.2020-30.12.2022 пересекаются. Функция должна возвращать 2 числа: количество грантов, в которых человек является руководителем (кроме текущего гранта), и количество грантов, в которых он является участником. Если человек не является руководителем текущего гранта, функция должна выдавать ошибку.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Гранты", для которых могут быть определены домены (дата начала меньше даты завершения; сумма финансирования не меньше 100000; первые две цифры идентификатора гранта равны двум последним цифрам первого года выполнения гранта, например, 15-01-00678 для даты '01.01.2015').
2. Триггер, переносающий в архив (в специальную таблицу) предыдущие значения данных об авторах при их изменении: могут поменяться любые сведения, кроме даты рождения. При попытке изменить дату рождения триггер выдает ошибку.

Вариант 7. БД собственников квартир.

Л. р. №6. Создание функций.

1. Функция, принимающая в качестве параметров две даты и возвращающая строку "несовершеннолетний", если между этими датами прошло менее 18-и лет и одного дня (человек становится совершеннолетним на следующий день после того, как ему исполняется 18 лет). Если вторая дата не определена, считать до текущей даты.
2. Функция, выводящая по номеру здания в специальную таблицу список квартир в этом доме, у которых в настоящее время более 10 собственников, или квартир, у которых в числе собственников есть несовершеннолетние. Параметры – номер здания (ПК) и режим работы: 0 – функция должна перед началом работы очищать таблицу, 1 – новые данные просто добавляются в таблицу. Таблица имеет следующие поля: номер здания, начало владения, ФИО собственника, доля, которой он владеет. Если таких квартир нет, выдавать сообщение об этом.

Л. р. №7. Создание триггеров.

1. Триггер, проверяющий правильность формирования полей "Серия документа" и "Номер документа". Маска серии:
 - для свидетельства о рождении или смерти – X...X-КК, где X...X – латинское число, КК – две русские буквы.
 - для российского паспорта – 4 цифры.Маска номера: шесть цифр (номер хранится в виде строки, т.к. в нем могут быть ведущие нули).
2. Проверка значений всех полей отношения "Владение", для которых могут быть определены домены (даты не больше текущей; дата окончания владения больше даты начала или не определена; номер квартиры содержит только цифры или цифры и одну

букву в конце). Если при вводе данных дата начала владения не указана, устанавливать текущую дату.

Вариант 8. БД пункта проката автомобилей.

Л. р. №6. Создание функций.

1. Функция, принимающая два параметра – дату и продолжительность аренды. Функция должна возвращать предположительную дату возврата автомобиля.
2. Функция, добавляющая в специальную таблицу «Отчет» сведения о каждом автомобиле определенной марки на текущую дату. Параметр – марка автомобиля. Функция должна перед началом работы очищать таблицу. Для автомобилей, находящихся в прокате, функция должна вносить номер, год выпуска, цвет, пробег, ФИО арендатора, начало и предполагаемую дату возврата автомобиля. Для автомобилей, не находящихся в прокате, функция должна вносить номер, год выпуска, цвет, пробег, дату последнего возврата из аренды и продолжительность аренды за последний год (в днях). Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Триггер, устанавливающий стоимость проката после установки даты возврата автомобиля как произведение стоимости аренды на ее реальную продолжительность.
2. Проверка значений всех полей отношения "Договоры на аренду", для которых могут быть определены домены: дата начала договора не меньше текущей, дата возврата автомобиля не определена, стоимость договора равна продолжительности, умноженной на стоимость аренды; продолжительность больше 0.

Вариант 9. БД библиотеки (книги).

Л. р. №6. Создание функций.

1. Функция, возвращающая пустую строку или строку "старое издание" для учебников, выпущенных 20 и более лет назад, для справочников, выпущенных 10 и более лет назад и для остальных книг, если они выпущены более 30 лет назад. Параметры: год издания и тип издания (значение поля 'примечание').
2. Функция добавления в специальную таблицу «Отчет» списка книг по указанной рубрике. Таблица должна содержать следующие поля: название рубрики, авторы, название книги, год выпуска, место издания, примечание (старое издание). Параметры – название рубрики и режим: 0 – функция должна перед началом работы очищать таблицу, 1 – новые данные просто добавляются в таблицу. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Каталог книг", для которых могут быть определены домены (в т.ч., год издания не может быть больше текущего года и меньше 1950; количество страниц – от 10 до 2000). Если год издания не указан, устанавливать текущий год. Если значение поля примечание – "учебник", поле «Авторы» не может быть пустым.
2. При изменении данных о рубриках – копирование старых значений в специальную таблицу. Сохранять в этой таблице дату изменения и имя пользователя, который произвел изменения.

Вариант 10. БД библиотеки (журналы).

Л. р. №6. Создание функций.

1. Функция, возвращающая количество страниц на основании значения поля "Страницы". Если значение поля 'x-y', то значение вычисляется как $(y-x+1)$. Если значение имеет вид 'x', возвращается 1. Если обнаружена ошибка ($y < x$ или вообще невозможно выделить числа), то возвращается -1.
2. Функция добавления в специальную таблицу «Отчет» списка книг указанного автора. Таблица должна содержать следующие поля: идентификатор автора, ФИО автора, название публикации, название журнала, год, номер выпуска, количество страниц, соавторы. Параметры – идентификатор автора и режим: 0 – функция должна перед началом работы очищать таблицу, 1 – новые данные просто добавляются в таблицу. Использовать ранее созданную функцию. Если у автора не было статей, выдавать сообщение об этом.

Л. р. №7. Создание триггеров.

1. Установка правильного формата поля "Страницы": оно должно содержать значение типа 'x-y', причем x должен быть меньше y . Если формат поля 'с.х-у', то 'с.' убирается. Если x равен y , то значение преобразуется к виду 'x'. Если формат нарушен (нельзя выделить одно или два числа), то триггер выдает сообщение об ошибке.
2. Фиксация всех изменений (по UPDATE и DELETE) отношения "Рубрикатор" в отдельной таблице (с указанием пользователя, который произвел изменения, и даты внесения изменений).

Вариант 11. БД гостиницы.

Л. р. №6. Создание функций.

1. Функция, возвращающая количество дней, прожитых постояльцем в гостинице на основании двух дат (день приезда – день отъезда считать как один день). Если второй параметр не передается, считать до текущей даты.
2. Функция подготовки счёта клиенту за предоставленные услуги. Функция помещает данные счета в таблицу «Счет» с полями: ФИО клиента, дата оказания услуги, количество, стоимость. Учитываются предоставленные услуги и проживание. Параметр – номер паспорта клиента. Если в один день клиенту было оказано несколько одинаковых услуг, выводить их одной строкой, указывая количество. Функция предварительно должна очищать таблицу Счет. Последняя добавляемая строка: ФИО, дата вселения, дата выезда, общая стоимость. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Копирование в архив (специальную таблицу) всех изменений таблицы "Услуги". Указывать в архиве дату внесения изменений и имя пользователя, который вносил изменения.
2. Проверка значений всех полей отношения "Постояльцы", для которых могут быть определены домены: дата вселения не меньше текущей; признак 0, 1 или 2; дата выселения не определена или больше даты вселения. Если значение поля "дата вселения" не определено, устанавливает текущую дату. Если признак равен 2, то дата выезда должна быть определена, иначе триггер выдает ошибку.

Вариант 12. БД адвоката.

Л. р. №6. Создание функций.

1. Функция, возвращающая список статей для клиента. Параметр – серия и номер паспорта. Статьи выдавать через запятую. Если статей нет, возвращать строку «данные не найдены».

2. Функция формирования отчета об эффективности защиты по законченным делам за указанный год путем добавления данных в специальную таблицу «Отчет». Таблица должна содержать следующие поля: дело, ФИО, дата рождения, список статей, минимальный срок, максимальный срок, срок по приговору. Параметр – год, за который требуется отчет. Значение "максимальный срок" брать как максимальный срок из всех статей, по которым обвинялся клиент, значение "минимальный срок" брать как минимальный срок из всех статей для данного клиента. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Клиенты", для которых могут быть определены домены (например, номер камеры от 101 до 599, размер гонорара не может быть отрицательным). Если на момент начала ведения дела обвиняемому меньше 18 лет, **добавлять** в поле "Примечание" 'несовершеннолетний' (если это не указано). Если значение поля "результат" – 'оправдан', а срок по приговору не равен 0, выдавать сообщение об ошибке.
2. Копирование в архив (специальную таблицу) всех изменений таблицы "Статьи". Указывать в архиве дату внесения изменений и имя пользователя, который вносил изменения.

Вариант 13. БД складского предприятия.

Л. р. №6. Создание функций.

1. Функция, возвращающая количество дней, прошедших между двумя датами. Если первая дата больше второй, функция возвращает -1.
2. Функция, создающая отчет о движении товаров за период путем добавления данных в специальную таблицу «Отчет». Таблица должна содержать следующие поля: название товара, единица измерения, количество поступившего товара, количество реализованного товара. Параметры: начало и окончание периода (даты). Если окончание периода не определено, считать его равным текущей дате. Если начало периода больше окончания, процедура должна выдавать соответствующее сообщение. Использовать ранее созданную функцию. Для каждого товара должна выводиться одна строка с суммарным количеством поставленного и реализованного товара.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Товары на складе", для которых могут быть определены домены (например, дата поставки не может быть больше текущей, остаток товара не больше количества поступившего товара, не меньше 0; количество поступившего товара больше 0). Если при вводе данных дата поставки не указана, устанавливать текущую дату.
2. Перенос в архив (специальную таблицу) сведений о поставках товаров, которые уже реализованы (при установке значения «Остаток товара» равным 0).

Вариант 14. БД магазина.

Л. р. №6. Создание функций.

1. Функция, возвращающая новую цену товара, если до завершения срока реализации осталось менее одного дня для скоропортящегося товара и менее одного 3-х дней для всех остальных. Цена уменьшается на 20%, но не более чем на 50 рублей. Параметры: цена товара, дата поставки и срок реализации.

2. Функция, создающая отчет об изменении цен на товары путем добавления данных в специальную таблицу «Изменение цен». Таблица должна содержать следующие поля: текущая дата, категория, артикул товара, название товара, единица измерения, остаток, старая цена, новая цена. Параметр: режим работы: 0 – функция должна перед началом работы очищать таблицу, 1 – новые данные просто добавляются в таблицу. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Поставки", для которых могут быть определены домены: количество товара и цена единицы поставки больше 0, дата поставки не может быть больше текущей, остаток товара не может превышать количество поставленного товара и быть меньше 0. Если при вводе данных дата поставки не указана, устанавливать текущую дату.
2. При изменении данных о товарах – копирование старых значений в специальную таблицу.

Вариант 15. БД деканата.

Л. р. №6. Создание функций.

1. Функция, возвращающая по параметрам "время" и "тип занятия" время окончания занятия, например:
 - для лекции или семинара в '09.00' вернет '10.20', т.е. занятие длится час двадцать;
 - для лабораторной работы в '09.00' вернет '10.30', т.е. полтора часа.
2. Функция, создающая отчет о некорректных данных в расписании для преподавателей кафедры путем добавления данных в специальную таблицу «Отчет». Таблица должна содержать следующие поля: ФИО преподавателя, день недели, количество занятий, начало первого занятия, начало следующего занятия. Расписание считается некорректным, если у одного преподавателя более 4-х занятий в день или если время разных занятий у одного преподавателя совпадает или пересекается. Первое и следующее занятия – это те занятия, которые пересекаются по времени (каждая пара – в отдельной строке). Если таких занятий нет, то эти поля пустые; если занятий не больше 4-х – соответствующее поле пустое. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Расписание занятий", для которых могут быть определены домены. Проверить тип занятий, день недели, аудиторию, занятия не могут начинаться раньше 9.00 и заканчиваться позднее, чем в 21.00.
2. Триггер, копирующий в архив (специальную таблицу) все данные об изменениях в таблице "Преподаватели" с указанием даты изменения и пользователя, который эти изменения внес.

Вариант 16. БД владельцев автомобилей.

Л. р. №6. Создание функций.

1. Функция, возвращающая сумму штрафа по трем параметрам: "Дата выписки", "Сумма штрафа", "Дата оплаты". Если дата оплаты не определена, возвращать NULL. Если между датой выписки и датой оплаты прошло более 20-ти дней, возвращать полную сумму штрафа, иначе – сумму, деленную пополам. Если сумма не определена или равна 0, выдавать сообщение об ошибке.
2. Функция, создающая отчет об уплате штрафов владельцем за некоторый период. Функция добавляет данные об уплате в специальную таблицу «Отчет», предварительно ее очищая. Параметры: идентификатор владельца, начало и окончание периода выписки

штрафа (даты). Если начало периода больше окончания, функция должна выдавать соответствующее сообщение. Если окончание периода не определено, считать его равным текущей дате. Поля таблицы: ФИО владельца, номер протокола, дата выписки штрафа, сумма выписанного штрафа, дата оплаты, оплаченная сумма. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Автомобили", для которых могут быть определены домены: год выпуска не может быть позже текущей даты и позже даты постановки на учет, дата снятия с учёта не может быть позже текущей даты; "Государственный номерной знак": буква – три цифры – две буквы – 2 или 3 цифры, перед добавлением записи необходимо перевести буквы в верхний регистр. Допустимые буквы: АВЕКМНОРСТУХ.
2. При удалении штрафа добавлять этот штраф в специальную архивную таблицу, если дата оплаты установлена. Если дата оплаты не определена, то возвращать сообщение об ошибке.

Вариант 17. БД кинологического клуба.

Л. р. №6. Создание функций.

1. Функция, возвращающая список наград по идентификатору собаки (через запятую). Параметры – идентификатор собаки и период (в годах), за который нужны награды. Если период не указан, считать его равным одному году. Формат: название награды¹ (дата), название награды² (дата),... Награды выводить в порядке получения.
2. Функция, формирующая список собак определенной породы путем добавления данных в специальную таблицу «Список». Таблица должна содержать следующие поля: Название породы, Кличка, Пол, Возраст, Награды. Параметр – название породы. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Если при вводе данных не указана дата получения награды, устанавливать текущую дату. Если указана дата больше текущей – генерировать ошибку.
2. Проверка значений всех полей отношения "Собаки", для которых могут быть определены домены (дата рождения не больше текущей; пол 'м' или 'ж'; значение поля "отец" не определено или относится к самцу, поля "мать" – не определено или относится к самке).

Вариант 18. БД рецептов блюд.

Л. р. №6. Создание функций.

1. Функция, возвращающая состав блюда одной строкой. Параметр: идентификатор блюда. Формат результата: название ингредиента (количество единица измерения), ... Например: 'мука (1 ст.), яйцо (2 шт.), вода (100 мл), соль (3 г)'.
2. Функция подбора блюд по значениям входных параметров. Параметры: тип блюда, ингредиенты (одной строкой, через запятую), сезон, максимальное время приготовления. Если сезон не указан, определять по текущей дате. Если не указано время, не учитывать его при выборе. Блюдо подходит, если в его рецепте есть хотя бы один ингредиент из перечисленных в параметре (и совпадает сезон, тип и время не больше указанного). Результат работы функции добавляется в таблицу «Подбор», которая в начале работы очищается от старых данных. Поля таблицы совпадают с полями таблицы «Блюда» плюс поле «состав». Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Блюда", для которых могут быть определены домены (количество порций больше 0, вес больше 10 (граммов), калорийность больше 0, сезон – список значений). Если при вводе данных не указано количество порций, устанавливать 1.
2. Автоматизация переноса в архив (специальную таблицу) данных о блюде при удалении его из основной таблицы.

Вариант 19. БД фитнес-клуба.

В таблице "Расписание занятий" тип поля "Продолжительность" надо изменить на interval.

Л. р. №6. Создание функций.

1. Функция, возвращающая по ФИО фамилию с инициалами. Параметр: ФИО (одной строкой). При невозможности преобразования должна возвращать исходное значение.
2. Функция, создающая отчет о некорректных данных в расписании для тренеров путем добавления данных в специальную таблицу «Отчет». Таблица должна содержать следующие поля: ФИО тренера, день недели, количество занятий, начало первого занятия, начало следующего занятия. Расписание считается некорректным, если у одного тренера более 4-х тренировок в день или если время разных занятий у одного тренера совпадает или пересекается. Первое и следующее занятия – это те занятия, которые пересекаются по времени, или перерыв между которыми меньше 20 минут (каждая пара пересекающихся занятий – в отдельной строке). Если таких занятий нет, то эти поля пустые; если занятий не больше 4-х – соответствующее поле пустое. Использовать ранее созданную функцию

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Клиенты", для которых могут быть определены домены (возраст не менее 14 лет; пол 'м' или 'ж'; вес не менее 40 кг, рост от 140 см до 220 см, окончание действия абонемента больше начала его действия). Если при вводе данных не указана дата начала действия абонемента, устанавливать текущую дату.
2. Автоматизация переноса в архив (специальную таблицу) всех изменений о клиентах (с указанием пользователя, вносящего изменения, и даты).

Вариант 20. БД спортивного клуба.

Л. р. №6. Создание функций.

1. Функция, определяющая, сколько времени с точностью до месяца прошло между двумя датами. Аргументы: первая дата, вторая дата. Вторая дата должна быть позже первой, в противном случае выдаётся ошибка. Если в функцию передаётся только один аргумент, второй приравнивается к текущей дате.
2. Функция, выводящая в специальную таблицу «Отчет» список всех тренеров, тренировавших спортсменов по определенному виду спорта. Параметры – вид спорта и режим: 0 – функция должна перед началом работы очищать таблицу, 1 – новые данные просто добавляются в таблицу. Таблица должна содержать следующие поля: вид спорта, ФИО спортсмена, ФИО тренера, дата начала тренировки, дата окончания тренировки, продолжительность тренировок. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. При изменении текущего тренера добавлять предыдущего тренера в таблицу "Предыдущие тренеры", в качестве даты окончания тренировок устанавливать текущую дату.

2. Проверка значений всех полей отношения "Спортсмены", для которых могут быть определены домены, например: номер домашнего телефона не должен совпадать с номером мобильного телефона; спортсмен не может быть партнёром самому себе; текущий рейтинг не может быть отрицательным; спортсмен не может быть моложе пяти лет. Если дата прихода спортсмена в клуб не определена, устанавливать текущую дату.

Вариант 21. БД аптеки.

Л. р. №6. Создание функций.

1. Функция, возвращающая строку «необходимо закупить», если количество упаковок, имеющихся в аптеке, меньше, чем минимальный запас, строку «подходит к концу», если количество упаковок, имеющихся в аптеке, отличается от минимального запаса менее чем на 30%; пустую строку в остальных случаях. Аргументы функции: количество лекарства в аптеке и минимальный запас.
2. Функция, выводящая в специальную таблицу «Отчет» список всех лекарств с пометкой о необходимости закупки или о подходящем к концу запасе лекарства. Параметр – режим: 0 – функция должна перед началом работы очищать таблицу, 1 – новые данные просто добавляются в таблицу. Таблица должна содержать следующие поля: название лекарства, фирма-производитель, остаток, примечание. Использовать ранее созданную функцию, результат которой записывается в поле Примечание.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Лекарства", для которых могут быть определены домены, например: количество таблеток в упаковке и цена – положительные числа. Для каждого поля выдавать свое сообщение об ошибке.
2. При удалении данных о лекарствах – перенос этих данных в архив (в специальную таблицу) с указанием даты переноса и пользователя, который это сделал.

Вариант 22. БД поликлиники.

Л. р. №6. Создание функций.

1. Функция, определяющая по дате рождения и текущей дате, к какой категории относится пациент:

Н	Новорождённый	от 1 до 10 дней
Г	Грудной ребёнок	от 10 дней до 1 года
Д-0	Раннее детство	от 1 до 2 лет
Д-1	I период детства	от 3 до 7 лет
Д-2	II период детства	от 8 до 12 лет
Пд	Подростковый возраст	от 13 до 16 лет
Ю	Юношеский возраст	от 17 до 21 года
Ср	Средний возраст	от 22 до 60 лет
Пж	Пожилой человек	от 61 до 75 лет
Ст	Старческий возраст	от 76 до 90 лет
Д	Долгожитель	старше 90 лет
2. Функция, записывающая в специальную таблицу ошибки в расписании приемов врачей. Параметры – начало и конец периода, за который выводятся данные. Корректные данные удовлетворяют следующим ограничениям: 1) все приемы происходят с 8.00 до 20.00, интервал между приемами – 15 минут; 2) один и тот же врач в одно время принимает только одного пациента (или ни одного, если никто не записался); 3) разница по времени между первым и последним приемом одного врача в один день должна быть меньше 8-ми

часов; 4) два разных приема не могут проходить в одно время в одном кабинете. Таблица с ошибками должна содержать следующие поля: ФИО врача, специализация, дата и время приема, описание ошибки.

Л. р. №7. Создание триггеров.

1. При удалении информации о пациентах, занесение данных о них в архив (специальную таблицу) с указанием даты переноса.
2. При добавлении записи в отношении «Назначения» проверять, не назначен ли пациенту визит к другому врачу в это же время.

Вариант 23. БД транспортного предприятия.

Л. р. №6. Создание функций.

1. Функция, определяющая продолжительность транспортного рейса. Параметры: дата и время отправления, дата и время прибытия.
2. Функция, записывающая в специальную таблицу ошибки в расписании рейсов. Параметры – начало и конец периода, за который выводятся данные. Корректные данные удовлетворяют следующим ограничениям: 1) рейс не может длиться более 8-ми часов; 2) интервал между двумя последовательными рейсами – не менее 30 минут, если рейсы длятся менее 4-х часов, не менее 2-х часов, если хотя бы один из рейсов длится более 4-х часов; 3) рейсы одного водителя не могут пересекаться по времени; 4) следующий рейс одного водителя должен начинаться из того же населенного пункта, в который он приехал предыдущим рейсом. Таблица с ошибками должна содержать следующие поля: ФИО водителя, дата/время отправления предыдущего рейса, пункт отправления предыдущего рейса, дата/время отправления следующего рейса, пункт отправления следующего рейса. Если нарушается правило (1), то два последних поля пустые.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Рейсы", для которых могут быть определены домены, например: время отправления не может быть раньше времени прибытия, длина маршрута не может быть отрицательной или быть равной нулю, пункт отправления не может быть равен пункту прибытия.
2. При изменении данных о водителе – перенос старой информации о нём в специальную таблицу (архив).

Вариант 24. БД библиотеки.

Л. р. №6. Создание функций.

1. Функция, выдающая список авторов произведений, опубликованных в книге. Параметр: шифр издания. Результат – строка вида 'А. Стругацкий, Б. Стругацкий'. Если авторов более 3-х, то вместо 4-го и последующих выдавать 'и др.'.
2. Функция, записывающая в специальную таблицу данные о книгах, вышедших в определенном издательстве. Параметры – название издательства и год издания. Таблица должна содержать следующие поля: Название издательства, Год издания, Шифр издания, Название книги, Автор(ы), Название произведения, Количество страниц. Если произведений в книге несколько, выводить «сборник». Таблица в начале работы очищается от старых данных. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Каталог книг", для которых могут быть определены домены, например: год издания больше 1900 и не больше следующего года (т.е. для текущей даты 01.10.2020 год издания может быть равен 2021), количество страниц должно быть положительным числом. Примечание переводится в нижний регистр.
2. При обновлении таблицы "Каталог книг" старые записи сохранять в отдельную таблицу с указанием даты изменения и именем пользователя, который внес изменение.

Вариант 25. БД страховой компании.

Л. р. №6. Создание функций.

1. Функция, подсчитывающая сумму страховой премии. Входной аргумент – стоимость полиса. Стоимость страховой премии рассчитывается по правилу:

Сумма страховой премии	Стоимость полиса от страховой премии
10 000 – 29 999 руб	0.75 %
30 000 – 69 999 руб	0.70 %
70 000 – 149 999 руб	0.60 %
150 000 – 299 999 руб	0.50 %
Более 299 999 руб	0.45 %

2. Функция, записывающая в специальную таблицу некорректные данные о страховании. Параметры – начало и конец периода, за который выводятся данные. Корректные данные удовлетворяют следующим ограничениям: 1) стоимость полиса зависит от суммы страховой премии и проверяется ранее написанной функцией; 2) сумма страховых выплат по полису не может превышать сумму страховой премии; 3) первый оплаченный страховой случай по полису не может быть ранее, чем через 6 дней после заключения договора. Таблица с ошибками должна содержать следующие поля: Номер страхового полиса, Паспортные данные, ФИО страхователя, Дата рождения, Название вида страхования, ФИО сотрудника, заключившего договор, Дата заключения договора, Дата окончания договора, Сумма страховой премии, Стоимость полиса, Сумма страховых выплат, Дата первого страхового случая. Если нарушается только правило (1), то два последних поля пустые; если нарушается только правило (2), то поля «Стоимость полиса» и «Дата первого страхового случая» пустые; если нарушается только правило (3), то поля «Стоимость полиса» и «Сумма страховых выплат» пустые.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Страхователи", для которых могут быть определены домены, например: договор не может оканчиваться раньше, чем начинается, суммы страховой премии и стоимость полиса не могут быть отрицательными и быть равны нулю; страхователь должен быть совершеннолетним на момент заключения договора; стоимость полиса зависит от суммы страховой премии и проверяется ранее написанной функцией.
2. Триггер, проверяющий, что дата страхового случая больше даты заключения договора плюс 6 дней, но не меньше даты окончания договора.

Вариант 26. БД расписание уроков.

Л. Р. №6. Создание функций.

1. Функция, возвращающая по параметру ФИО фамилию учителя с инициалами. Если не удастся выделить 3 слова, функция возвращает исходное значение.
2. Функция, выводящая в специальную таблицу "окна" для учителей (пропуски между уроками). Перед началом работы функция должна очищать таблицу. Таблица должна содержать следующие поля: день недели – Фамилия И.О. учителя – номер начала "окна" – номер окончания "окна". Например, если у учителя есть 2-й, 4-й и 7-й уроки, то его "окна" – 3-3 и 5-6. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Расписание", для которых могут быть определены домены (значения полей "Кабинет", "Класс", "Номер урока", "День недели").
2. Триггер, определяющий перед добавлением записей в отношении «Расписание», не назначены ли уже занятия на этот урок в том кабинете, в котором планируется проводить занятия.

Вариант 27. БД кинофильмов.

В таблице "Кинофильмы" тип поля "Продолжительность" надо изменить на interval.

Л. р. №6. Создание функций.

1. Функция, возвращающая перечень создателей фильма в виде строки "А.Алов, В.Наумов". Параметры: идентификатор фильма и вид участия (режиссер, актер и т.д.) Для актеров возвращаемая строка примет следующий вид: "Е.Леонов (Сарафанов), Н.Караценцов (Бусыгин)" – в скобках указывается роль.
2. Функция, выводящая в специальную таблицу "Подбор" фильмы по определенному жанру. Параметры: жанр и период выхода, например, ('Фантастика', 2018, 2020). Если второй и/или последний параметр не указан, считать их равным текущему году. Перед началом работы функция должна очищать таблицу. Таблица должна содержать следующие поля: Название фильма, Страна, Год выхода, Продолжительность, Режиссеры, Сценаристы, Операторы, Композиторы, Актеры. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка и изменение данных в таблице "Кинофильмы". Продолжительность фильма больше 10 минут. Если при вводе данных год выхода фильма не указан, устанавливать текущий год. Если указан год больше текущего – генерировать ошибку. Содержимое поля "Страна" изменять следующим образом:
РФ, Российская Федерация – "Россия";
Великобритания – "Англия";
Голландия – "Нидерланды";
КНР – "Китай".
2. При удалении данных о фильмах записывать их в архив (в специальную таблицу). Добавлять туда дату внесения данных и имя пользователя, который удалил фильм.

Вариант 28. БД деканата (Сессия).

Л. р. №6. Создание функций.

1. Функция, возвращающая по дате/времени строку, в которой указаны время начала и завершения экзамена или консультации. В функцию передаются 2 параметра: дата/время и тип (0 – экзамен, 1 – консультация). Продолжительность экзамена – 5 часов, консультации – 2 часа. (Например, для времени 9.00 и типа 'экзамен' функция должна вернуть '9.00 – 14.00').

2. Функция, выводящая в специальную таблицу расписание сессии. Использовать ранее созданную функцию. В таблице должны быть следующие поля: Группа, Название дисциплины, ФИО преподавателя, дата консультации, время консультации, аудитория консультации, дата экзамена, время экзамена, аудитория экзамена. Перед началом работы функция должна очищать таблицу.

Л. р. №7. Создание триггеров.

1. Проверка значений всех полей отношения "Сессия", для которых могут быть определены домены: курс – от 1 до 6, аудитории – от 101 до 522, время – от 9.00 до 17.00, кратно получасу; экзамен не должен заканчиваться позднее 19.00.
2. Регистрация изменений: при модификации или удалении записи из таблицы "Дисциплины" старое содержимое этой записи дублируется в другую таблицу с указанием даты модификации и пользователя, изменившего запись.

Вариант 29. БД транспортного предприятия.

Примечание: в таблице "Маршруты" изменить тип поля "время отправления" на INTERVAL.

Л. р. №6. Создание функций.

1. Функция, принимающая два параметра – дату и периодичность рейса (ежедн., четн., нечет., день недели). Функция должна возвращать null, если на указанную дату нет рейса (в соответствии с периодичностью), или исходную дату, если рейс есть.
2. Функция, формирующая таблицу «Рейсы» на будущий период. Параметр – дата, из которой берется год и месяц, на который надо составить расписание. Этот месяц должен относиться к будущему времени (например, в апреле можно составить расписание на май). Поле "Номер маршрута" берется из таблицы "Маршруты", "Дата выезда" вычисляется на основании периодичности рейса с помощью ранее созданной функции. Поле "Водитель" остается пустым, количество проданных билетов равно 0. Функция не должна добавлять (или изменять) те рейсы, которые уже есть в таблице.

Л. р. №7. Создание триггеров.

1. Триггер, проверяющий, что количество проданных на рейс билетов не превышает количество мест в автобусе.
2. Проверка значений всех полей отношения "Маршруты", для которых могут быть определены домены: пункт отправления не равен пункту прибытия; время в пути не менее получаса; цена билета – от 50 до 1500 рублей.

Вариант 30. БД детского клуба

В таблице "Расписание занятий" тип поля "Продолжительность" надо изменить на interval.

Л. р. №6. Создание функций.

1. Функция, принимающая четыре параметра – дату, день недели, время начала и продолжительность занятий. Если дата является указанным днем недели, то функция должна возвращать время окончания занятий, иначе – NULL.
2. Функция, выводящая в специальную таблицу «Свободные помещения» данные о незанятых помещениях на будущей неделе. Параметр – интервал: минимальное время, в течение которого помещение должно быть свободным. Таблица должна содержать следующие поля: Дата, Номер помещения, Начало перерыва, Окончание перерыва в занятиях. Помещение считается свободным, если там нет занятий в период с 9.00 до 20.00, и длительность перерыва не менее указанного интервала. Например, если в среду в комнате №20 занятия начинаются в 11.00 и интервал – 1 ч, то эта комната свободна с 9 до

11 ч. Использовать ранее созданную функцию. Функция должна предварительно очищать эту таблицу.

Л. р. №7. Создание триггеров.

1. Триггер, проверяющий, что добавляемый в расписание занятий кружок посещает хотя бы один ребенок.
2. Проверка значений всех полей отношения "Расписание занятий", для которых могут быть определены домены: помещение входит в список помещений; указан допустимый день недели (с учетом регистра), занятия начинаются не раньше 9.00 и заканчиваются не позднее 20.00; продолжительность занятий – от 20 минут до 2 часов.

Вариант 31. БД "Классификация СУБД".

1. Функция, выдающая список характеристик по версии СУБД. Параметр: версия СУБД. Результат – строка вида 'Операционная система: Linux, FreeBSD, QNX. Объем ОП: 8 Гб. ...'. (Здесь Гб – единица измерения для числовых данных).
2. Функция, записывающая в специальную таблицу данные о характеристиках всех версий СУБД, начиная с определенного года выхода версий. Параметры – название СУБД и год выхода. Таблица должна содержать следующие поля: Название СУБД, Дата выхода, Версия, Поддерживаемая модель данных, Список характеристик. Таблица в начале работы очищается от старых данных. Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Проверка значений полей отношения "Характеристики СУБД". Если указано значение строкового типа, то единица измерения этой характеристики не определена. Если указано значение числового типа, то должна быть определена единица измерения этой характеристики. Одновременно указывается или строковое, или числовое значение.
2. При изменении (update, delete) таблицы "Характеристики СУБД" старые записи сохранять в отдельную таблицу с указанием даты изменения и имени пользователя, который внес изменение.