Практика №6.2
Выполнение задания на основе индивидуального задания

Дисциплина	Базы данных для индустриальных задач
Институт	Перспективных технологий и индустриального программирования
Кафедра	Индустриального программирования
Вид учебного материала	Практика
Преподаватель	Евдошенко Олег Игоревич
Семестр	1 семестр, 2025-2026

Вариант 1. БД спортивного клуба.

Задание №1. 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, преобразующая значение ФИО (один входной параметр) в фамилию с инициалами (например, "Иванов Иван Сергеевич" в "Иванов И.С."). При невозможности преобразования функция возвращает строку '##############'.
3. Функция преобразования номера телефона в строку вида '8-XXX-XXX-XX-XX'. Входной параметр – строка с номером телефона в виде 11-и, 10-и или семизначной последовательности.

Задание №2. Создание процедур.

1. Процедура, выводящая список всех спортсменов с указанием их тренеров:

ФИО спортсмена1
- ФИО тренера1 начало_тренировок окончание_тренировок
- ФИО тренера2 начало_тренировок окончание_тренировок
- …

ФИО спортсмена2
- ФИО тренера1 начало_тренировок окончание_тренировок
- ФИО тренера2 начало_тренировок окончание_тренировок
- …

2. Процедура, выдающая пары спортсменов для парных видов спорта:

Вид_спорта1
1. ФИО_спортсмена1 – ФИО_спорсмена2 (ФИО_тренера)
2. ФИО_спортсмена1 – ФИО_спорсмена2 (ФИО_тренера)
…

Если спортсменов тренируют разные тренеры, выдавать две фамилии. Не допускать повторов.

3. Процедура, рассчитывающая текущий рейтинг тренеров на основании текущего рейтинга спортсменов. При расчете должны учитываться также рейтинги спортсменов, которых тренер тренировал в течение последнего года

Задание №3. Создание триггеров.

1. Автоматическое формирование таблицы перехода спортсмена от одного тренера к другому (при изменении поля "Тренер" в таблице "Спортсмены").
2. Увеличение рейтинга тренера при изменении сведений об уровне мастерства его спортсмена (при повышении уровня): плюс 20 баллов к рейтингу.
3. Проверка значений всех полей отношения "Спортсмены", для которых могут быть определены домены.

Вариант 2. БД поликлиники.

Задание №1. Создание функций.

1. Функция, принимающая на вход номер полиса и возвращающая строку с этим номером, разбитую на две части пробелом после 6-го символа. Если исходный номер содержит меньше 16-ти цифр, возвращать '###########'.
2. Функция, преобразующая значение ФИО (один входной параметр) в фамилию с инициалами (например, "Иванов Иван Сергеевич" в "Иванов И.С."). При невозможности преобразования функция возвращает строку '##############'.
3. Функция, определяющая, является ли человек пенсионером по полу и дате рождения. Возвращает строку "пенсионер" или пустую строку.

Задание №2. Создание процедур.

1. Процедура, выдающая расписание приемов на текущую дату:
   Дата
   врач1 время1 пациент1 адрес_пациента
   время2 пациент2 адрес_пациента
   …
   врач2 время1 пациент1 адрес_пациента
   время2 пациент2 адрес_пациента
   …
2. Процедура, выдающая по специализации врача и дате список незанятых приемов.
   Специализация
   ФИО_врача1 время_начала_приема время_окончания_приема
   время_начала_приема время_окончания_приема
   …
   ФИО_врача1 время_начала_приема время_окончания_приема
   …
3. Процедура, определяющая наступление эпидемии. Правило такое: если за последнюю неделю количество заболевших с диагнозами ОРЗ, ОРВИ и грипп за каждый следующий день увеличивалось не менее чем на 50% по сравнению с предыдущим днем, то эпидемия наступила.

Задание №3. Создание триггеров.

1. Проверка значений всех полей отношения "Пациенты", для которых могут быть определены домены.
2. Если при вводе данных дата поступления не указана, устанавливать текущую дату.
3. При удалении данных о пациенте – перенос этих данных в архив.

Вариант 3. БД собственников квартир.

Задание №1. 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. Если вторая дата не определена, считать до текущей даты.
2. Функция, проверяющая правильность поля "Пол". Параметры: отчество и пол. Возвращаемое значение – пустая строка или строка "не соответствует", если отчество не соответствуют установленному полу. Основные правила: мужской пол – отчество оканчивается на 'ИЧ'; женский пол – окончание отчества 'НА'.
3. Функция, преобразующая значение ФИО в фамилию с инициалами (например, "Иванов Иван Сергеевич" в "Иванов И.С.").

Задание №2. Создание процедур.

1. Процедура, выводящая на экран по номеру дома список квартир, по которым нет информации о собственниках. Алгоритм работы: найти максимальный номер квартиры и вывести все номера с 1-го, которых нет в таблице "Владение".
2. Процедура, проставляющая дату окончания владения по дате выдачи свидетельства о смерти.
3. Процедура, выдающая на экран по номеру дома список квартир, у которых в настоящее более 10 собственников или квартир, у которых с начала первого владения более 20 собственников, в том числе, бывших.

Задание №3. Создание триггеров.

1. Триггер, проверяющий, что дата окончания владения или не определена, или больше даты начала владения.
2. Триггер, переносящий в архив все записи, удаляемые из таблицы "Владение".
3. 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).

Вариант 4. БД транспортного предприятия.

Задание №1. Создание функций.

1. Функция, возвращающая строку со временем прибытия по времени отправления и времени в пути.
2. Функция, возвращающая 1, если сегодня есть указанный рейс, и 0 в противном случае. Параметр: периодичность рейса (ежедн., четн., нечет., день недели).
3. Функция, преобразующая значение ФИО в фамилию с инициалами (например, "Иванов Иван Сергеевич" в "Иванов И.С.").

Задание №2. Создание процедур.

1. Процедура, выдающая расписание работы водителя на текущий месяц.
2. Процедура, проверяющая, что у водителей нет недопустимых рейсов. Недопустимым считается, если у водителя:
    - общее время в пути превышает 6 часов в день;

- более 3-х рейсов в день;
- промежуток между двумя рейсами менее 1 часа;
- конечный пункт предыдущего и следующего рейсов не совпадают.

3. Процедура, рассчитывающая общую стоимость выполненных за месяц рейсов. Входные параметры – водитель и дата, из которой надо извлечь год и месяц. Если рейс начинается в один месяц, а заканчивается в следующем, учитывать только дату прибытия.

Задание №3. 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.
3. Триггер, переносящий сведения об удаляемых рейсах в архив (в специальную таблицу).

Вариант 5. БД библиотеки.

Задание №1. 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, возвращающая для значения поля "Место издания":
   "М" – строку "Москва",
   "Л" – строку "Ленинград",
   "Мн" – строку "Минск",
   "К" – строку "Киев",
   "СПб" – строку "Санкт-Петербург",
   а для всех остальных значений – исходное место издания без изменения.
3. Функция, преобразующая значение ФИО в фамилию с инициалами (например, "Тургенев Иван Сергеевич" в "Тургенев И.С."). Два параметра: строка ФИО и флаг. Если флаг равен 0, то при невозможности преобразования функция возвращает исходную строку; если он равен 1, то возвращает строку '###########'.

Задание №2. Создание процедур.

1. Процедура списания книг, изданных более 50 лет назад (перенос в архив). При этом если какого-либо из опубликованных в книге произведений нет больше ни в одной книге, то данные об этом произведении удаляются в архив.
2. Процедура вывода содержания книги по ее названию и шифру в виде:

   Шифр. Название [старое издание]
   год выпуска, издательство
   Содержание:

   1. автор1 произведение1 тип
      автор2
   2. (без автора) произведение2 тип
   3. автор1 произведение3
   4. автор1 произведение4
      автор2
      автор3

...
3. Процедура, выдающая на экран библиографическое описание книги (по ее названию и шифру) в одном из следующих форматов:
   – для книг с одним произведением и одним или несколькими авторами:
     Фамилия1 И.О., Фамилия2 И.О. Название книги. Название произведения (тип произведения). – Издательство, год выпуска. – N с.
   – для книг с одним произведением и без автора:
     Название книги. Название произведения (тип произведения). – Издательство, год выпуска. – N с.
   – для книг с несколькими произведениями:
     Название книги. (сборник) – Издательство, год выпуска. – N с.
   (Название произведения выводится, если название книги не совпадает с названием произведения, тип выводится всегда. N – количество страниц).

Задание №3. Создание триггеров.

1. Автоматизация переноса удаляемой книги в архив.
2. Проверка значений всех полей отношения "Каталог книг", для которых могут быть определены домены (в т.ч., год издания и количество страниц).
3. При изменении данных о книгах – копирование старых значений в специальную таблицу. Сохранять в этой таблице дату изменения и имя пользователя.

Вариант 6. БД страховой компании.

Задание №1. Создание функций.

1. Функция, принимающая в качестве входного параметра дату начала договора, длительность и признак: ‘день’, если длительность измеряется в днях, ‘месяц’ – в месяцах и ‘год’ – в годах. Функция должна вернуть дату окончания срока действия договора. Например, для даты 20.10.2014 с продолжительностью 1 год функция должна вернуть 19.10.2015, а для 01.02.2013 – 31.01.2014.
2. Функция, возвращающая возраст страхователя на момент завершения действия договора страхования. Входные параметры: дата окончания договора и дата рождения страхователя.
3. Функция, рассчитывающая стоимость полиса по стоимости вида страхования. Входные параметры: вид страхования и срок заключения договора.

Задание №2. Создание процедур.

1. Процедура, выводящая результаты расчетов со страхователем по номеру страхового полиса:
   № страхового_полиса дата_заключения_договора – дата_окончания_договора
   ФИО_страхователя возраст_страхователя
   Вид _трахования Стоимость_полиса Сумма_страховой_премии
   Дата_наступления_страхового_случая1 сумма_страховой_выплаты1
   Дата_наступления_страхового_случая2 сумма_страховой_выплаты2
   …
   Общая сумма выплат по полису
2. Процедура переноса данных о страховых случаях по закончившимся полисам в архив. Добавить в таблицу «Страхователи» поле «Общая сумма выплат» и при удалении данных указывать в этом поле сумму, выплаченную по данному полису.
3. Процедура, определяющая прибыльность отдельных видов страхования. Выдает отчет вида:
   Вид страхования1

Срок1 количество_полисов1 сумма_премии1 средняя_прибыль(убыток)
Срок2 количество_полисов2 сумма_премии2 средняя_прибыль(убыток)
…
Вид страхования1
Срок1 количество_полисов1 сумма_премии1 средняя_прибыль(убыток)
Срок2 количество_полисов2 сумма_премии2 средняя_прибыль(убыток)
…

Данные группируются по виду страхования, продолжительности и сумме страховой премии, прибыль (убыток) рассчитывается как разница между стоимостью полиса и страховыми выплатами.

Задание №3. Создание триггеров.

1. Проверка значений всех полей отношения "Страхователи", для которых могут быть определены домены.
2. Регистрация изменений: при модификации или удалении записи из таблицы "Виды страхования" старое содержимое этой записи дублируется в другую таблицу с указанием даты модификации и пользователя, изменившего запись.
3. Проверка: дата наступления страхового случая должна находиться в промежутке между датами начала и окончания действия договора.

Вариант 7. БД расписания уроков.

Задание №1. Создание функций.

1. Функция, преобразующая значение ФИО в фамилию с инициалами (например, "Иванов Иван Сергеевич" в "Иванов И.С."). При невозможности преобразования функция возвращает строку '##############'.
2. Функция, возвращающая время начала урока по его номеру.
3. Функция, возвращающая одной строкой перечень предметов, которые может вести учитель. Входной параметр – идентификатор учителя.

Задание №2. Создание процедур.

1. Процедура, выводящая расписание занятий определенного класса на неделю:
   Класс
   пн: 1. Предмет кабинет ФИО_учителя
   2. Предмет кабинет ФИО_учителя
   …
   вт: 1. Предмет кабинет ФИО_учителя
   2. Предмет кабинет ФИО_учителя
   …
2. Процедура, выводящая расписание для определенного учителя.
   ФИО_учителя
   пн: 1. Предмет кабинет класс
   2. (окно)
   3. Предмет кабинет класс
   …
   вт: 1. Предмет кабинет
   …

   Если возникает ошибка (2 предмета в одно время), то процедура должна выводить сообщение об ошибке.
3. Процедура, выводящая нагрузку учителей по каждому предмету:
   ФИО_учителя1 категория всего_часов

предмет1 количество_часов
предмет2 количество_часов
…

ФИО_учителя2 категория всего_часов
предмет1 количество_часов
…

Если нагрузки по предмету нет, выводится 0.

Задание №3. Создание триггеров.

1. Проверка значений всех полей отношения "Расписание", для которых могут быть определены домены. В частности, предмет должен соответствовать уровню класса, а максимальное количество уроков для начальной школы – 5, для средней – 7. Класс – это номер класса (1-11) плюс буква (а, б, в или г).
2. Фиксация в специальной таблице изменений таблицы «Специализация» с указанием даты внесения изменений и пользователя, который их вносил.

Вариант 8. БД отдела кадров.

Задание №1. Создание функций.

1. Функция, определяющая по дате рождения, является ли человек юбиляром в текущем году, и выдающая для юбиляра возраст (юбилейную дату, например, "45"), а в противном случае – пустую строку.
2. Функция, преобразующая ФИО в фамилию с инициалами (например, "Иванов Иван Сергеевич" в "Иванов И.С."). Принимает два параметра – фамилия и имя-отчество.
3. Функция, преобразующая оклад (число) в строку. Если число целое, то формат "X руб.", если дробное – "X руб. Y коп."

Задание №2. Создание процедур.

1. Процедура, выводящая список всех сотрудников – юбиляров текущего года (с указанием даты юбилея и возраста).
2. Процедура, выводящая список вакансий в виде:
   Отдел_1 должность1 оклад количество_вакантных_ставок
   должность2 оклад количество_вакантных_ставок
   …
   Отдел_2 должность1 оклад количество_вакантных_ставок
   должность2 оклад количество_вакантных_ставок
   …

3. Процедура, проверяющая отсутствие превышения лимита по ставкам для должностей. Результат (нарушения) выводит в специальную таблицу "Ошибки" (отдел – должность – количество ставок всего – количество занятых ставок – разницу между 3-м и 4-м значением). Процедура предварительно очищает таблицу. Если нарушений нет, выводит сообщение "Превышения нет".

Задание №3. Создание триггеров.

1. Проверка значений всех полей отношения "Сотрудники", для которых могут быть определены домены (проверить пол по отчеству, возраст не менее 16-ти лет, дата приема на работу не меньше текущей и т.д.)
2. Замена значений поля "Образование":

"ср.", "сред." и т.п. на "среднее",

"ср. спец.", "среднее спец." и т.п. на "среднее специальное",
"выс.", "высш." и т.п. на "высшее",
"н. высшее", "н. высш." и т.п. на "незаконченное высшее".
(Если значение не входит в список допустимых, триггер должен выдавать ошибку).

3. Регистрация изменений, вносимых в таблицу "Сотрудники" (дублирование старой записи в специальной таблице с указанием даты изменения и пользователя, который их проводит).

Вариант 9. БД бухгалтерии.

Задание №1. Создание функций.

1. Функция расчета общего стажа работы (в годах и месяцах). Принимает 2 аргумента: стаж на прежних местах работы и дату поступления на данную работу. Возвращает число, в котором дробная часть – это количество полных месяцев. В аргументе "стаж на прежних местах работы" дробная часть – это тоже количество месяцев (не может быть больше 11).
2. Функция, преобразующая строку в дату. Возможные форматы 'DD.MM.YYYY' или 'YYYY.MM.DD', причем вместо точки можно использовать - или /. Если исходная строка не является датой, функция должна возвращать '##########'.
3. Функция, принимающая на входе дату рождения и возвращающая 1, если человек с такой датой рождения является несовершеннолетним, и 0 в противном случае. Функция возвращает -1, если указанная дата больше текущей.

Задание №2. Создание процедур.

1. Процедура, повышающая заработную плату:
   – до размера в 1.5 прожиточного минимума, если оклад меньше, чем (прожиточный минимум)*1.5;
   – на 10%, если оклад больше, чем 1.5 прожиточного минимума;
   – на 15%, если должность относится к первой категории.
   Входной параметр – прожиточный минимум.
2. Процедура расчета зарплаты, выдающая на экран результаты в виде:
   отдел1 1. фамилия1 сумма
   2. фамилия2 сумма
   …
   отдел2 1. фамилия1 сумма
   2. фамилия2 сумма
   …
   Сумма рассчитывается как (оклад-13%). Для сотрудников, имеющих несовершеннолетних детей, сумма налогообложения уменьшается на МРОТ*13% на каждого ребенка.
3. Процедура начисления 13-й зарплаты. Размер 13-й зарплаты равен (оклад плюс надбавка), надбавка зависит от стажа работы на данном предприятии:
   10% от оклада, если стаж от 1 до 5 лет;
   25% от оклада, если стаж от 5 до 10 лет;
   40% от оклада, если стаж от 10 до 20 лет;
   60% от оклада, если стаж от 20 до 30 лет;
   100% от оклада, если стаж свыше 30 лет.

Задание №3. Создание триггеров.

1. Проверка значений всех полей отношения "Сотрудники", для которых могут быть определены домены (в т.ч., (возраст сотрудника)-(стаж на прежних работах)-(стаж работы на данном предприятии) не может быть меньше 16, а дата поступления на работу должна быть не больше текущей даты).
2. Проверка: в каждом отделе может быть только один начальник отдела или директор (триггер уровня предложения).
3. Регистрация изменений, вносимых в таблицу "Сотрудники" (дублирование старой записи в специальной таблице с указанием даты изменения и пользователя, который их проводит).

Вариант 10. БД деканата.

Задание №1. Создание функций.

1. Функция, возвращающая по дате строку, в которой указаны время начала и завершения экзамена или консультации. 2 параметра: дата 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, преобразующая две строки формата 'DD.MM.YYYY' и 'HH24:MI' в дату. Время должно находиться в интервале от 9.00 до 18.00 и должно быть кратно получасу. Если исходные строки не могут быть преобразованы в дату или время выходит за указанные границы, функция должна возвращать NULL.
3. Функция, возвращающая сокращенные названия должностей:
   'ст. препод.' для старшего преподавателя,
   'доц.' для доцента,
   'проф.' для профессора.
   Для всех остальных функция возвращает входной параметр без изменений.

Задание №2. Создание процедур.

1. Процедура, выводящая расписание сессии. Список должен выглядеть так:
   факультет1
   группа1 дисципл1 препод. консультация (дата, время, ауд.) экз. (дата, время, ауд.)
   дисципл2 препод. консультация (дата, время, ауд.) экз. (дата, время, ауд.)
   дисципл3 препод. консультация (дата, время, ауд.) экз. (дата, время, ауд.)
   группа2 дисципл1 препод. консультация (дата, время, ауд.) экз. (дата, время, ауд.)
   дисципл2 препод. консультация (дата, время, ауд.) экз. (дата, время, ауд.)
   …
   факультет2
   …
2. Процедура назначения консультаций: накануне каждого экзамена, в свободной аудитории, не позднее 17.00. Время консультации – два часа; время экзамена – пять часов. Считайте, что все номера аудиторий указаны в этой же таблице "Сессия".
3. Процедура, выдающая ведомость на оплату экзаменационной сессии (каждая фамилия должна встречаться один раз). Профессор за экзамен получает 5000 руб., доцент – 4000 руб., старший преподаватель – 3000 руб.

Задание №3. Создание триггеров.

1. Проверка значений всех полей отношения "Сессия", для которых могут быть определены домены.
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.

3. Проверка: в каждый день у преподавателя может быть только один экзамен (но может быть еще консультация).

Вариант 11. БД спортивного клуба.

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, преобразующая значение ФИО (один входной параметр) в фамилию с инициалами (например, "Иванов Иван Сергеевич" в "Иванов И.С."). При невозможности преобразования функция возвращает строку '##############'.
3. Функция преобразования номера телефона в строку вида '8-XXX-XXX-XX-XX'. Входной параметр – строка с номером телефона в виде 11-и, 10-и или семизначной последовательности.

Задание №2. Создание процедур.

1. Процедура, выводящая список всех спортсменов, которые участвовали в соревнованиях:

Название соревнования1	место1	ФИО спортсмена	ФИО тренера
	место2	ФИО спортсмена	ФИО тренера
…			
Название соревнования2	место1	ФИО спортсмена	ФИО тренера
…			

2. Процедура, проверяющая, что места участникам соревнований расставлены в соответствии с завоеванными очками, и выдающая нарушения расстановки мест. Проверяется только соответствие правилу "чем больше очков, тем выше место".
3. Процедура, устанавливающая стипендию спортсменам. Параметр – размер стипендиального фонда. Правила такие: спортсмен, не участвовавший в соревнованиях в прошлом году, стипендию не получает. А остальные получают стипендию пропорционально рейтингу, но так, чтобы общая сумма совпадала с размером стипендиального фонда.

Задание №3. Создание триггеров.

1. Фиксация в специальной таблице перехода спортсмена от одного тренера к другому (при изменении поля "Тренер" в таблице "Спортсмены").
2. 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.
3. Проверка значений всех полей отношения "Спортсмены", для которых могут быть определены домены.

Вариант 12. БД диссертаций.

Задание №1. Создание функций.

1. Функция, преобразующая значение ФИО в фамилию с инициалами (например, "Иванов Иван Сергеевич" в "Иванов И.С."). При невозможности преобразования функция возвращает строку '##############'.
2. Функция, выдающая полное название учетной степени по параметрам "Тип" и "Раздел науки". Например, для типа "докторская" и раздела "Технические науки" функция должна вернуть "доктор технических наук".
3. Функция, выдающая возраст по двум датам: дате рождения и дате, на которую интересует возраст. Если вторая дата не указана, то возраст на текущую дату. Вызывать функцию для двух полей: дата рождения автора и дата защиты диссертации.

Задание №2. Создание процедур.

1. Процедура, выдающая отчет по диссертациям, защищенным в определенном году:
   Год
   Раздел1
   Направление1
   ФИО1 дата защиты название диссертации
   ФИО2 дата защиты название диссертации
   …
   Направление2
   ФИО1 дата защиты название диссертации
   …
   Раздел2
   Направление1
   ФИО1 дата защиты название диссертации
   …
   Сначала для каждого научного направления выводятся данные о докторских диссертациях, затем – о кандидатских. Параметр – год защиты.
2. Процедура поиска авторов, которые внесены в таблицу "Авторы" дважды. Идентификация автора происходит по совпадению ФИО и даты рождения (паспортные данные могут измениться). Если при этом диссертации (кандидатская и докторская) защищены по одному направлению, то это один и тот же человек. Данные о нем объединяются: в таблице "Авторы" остается одна строка с более поздней датой выдачи паспорта.
3. Выдает список авторов с указанием ученой степени. Если автор защитил кандидатскую и докторскую диссертации по одному разделу, то он является доктором наук. Если разделы разные, то ученые степени перечисляются через запятую (например, 'кандидат экономических наук, доктор технических наук').

Задание №3. Создание триггеров.

1. Проверка значений всех полей отношения "Диссертации", для которых могут быть определены домены (в т.ч., дата защиты и дата утверждения не могут быть больше текущей даты).
2. Замена при добавлении данных сокращенных значений поля "Тип" отношения "Диссертации" на полные ("канд" на "кандидатская" и т.п.).
3. Триггер, переносящий в архив (в специальную таблицу) изменения сведений об авторах.

Вариант 13. БД больницы.

Задание №1. Создание функций.

1. Функция, возвращающая строку "больше месяца", если со времени поступления пациента прошло более 1 месяца (один месяц – 30 дней). Параметр – дата поступления.
2. Функция, преобразующая значение ФИО в фамилию с инициалами (например, "Иванов Иван Сергеевич" в "Иванов И.С."). При невозможности преобразования функция возвращает строку '##############'.
3. Функция, принимающая на вход номер полиса и возвращающая строку с этим номером, разбитую на две части пробелом после 6-го символа. Если исходный номер содержит меньше 16-ти цифр, возвращать '############'.

Задание №2. Создание процедур.

1. Процедура, распределяющая больных временно отсутствующего лечащего врача по другим лечащим врачам в соответствии со специализацией врачей. Процедура не должна допускать перекоса при распределении пациентов (т.е. всех пациентов – одному врачу). В качестве параметра принимает идентификатор врача.
2. Процедура, выводящая список палат с указанием количества коек и статуса палаты:
   "пустая", если в палате никто не лежит,
   "свободных мест нет", если палата заполнена,
   "мужская", если в палате лежат только мужчины,
   "женская", если в палате лежат только женщины,
   "ошибка в данных" во всех остальных случаях.
   Параметр – название отделения.
3. Процедура, создающая отчет "Оборачиваемость коек": отделение – количество коек – коэффициент занятости. Коэффициент занятости высчитывается как отношение суммарной продолжительности пребывания пациентов в отделении в течение года, к произведению общего количества дней в году на количество коек в данном отделении.

Задание №3. Создание триггеров.

1. Проверка значений всех полей отношения "Пациенты", для которых могут быть определены домены.
2. Если при вводе данных дата поступления не указана, устанавливать текущую дату.
3. При удалении данных о пациенте – перенос этих данных в архив.

Вариант 14. БД «Отдел поставок».

Задание №1. Создание функций.

1. Функция, возвращающая строку с указанием остатка товара. Параметры – остаток и единица измерения. Пример: параметры 10 и 'шт' – возвращает строку '10 шт.'. Если второй параметр не указан, то единица измерения – кг.
2. Функция, преобразующая значение ФИО в фамилию с инициалами (например, "Иванов Иван Сергеевич" в "Иванов И.С."). При невозможности преобразования функция возвращает строку '##############'.
3. Функция, возвращающая строку с указанием цены товара. Параметр – цена. Пример: параметр 10.75 – возвращает строку '10 руб. 75 коп.'. Если цена – целое число, то должна вернуть строку '10 руб.'.

Задание №2. Создание процедур.

1. Процедура, выводящая на экран список комплектующих в виде:
   артикул название поставщик дата поставки цена количество

…

Упорядочить по дате поставки и поставщикам. Использовать функции из предыдущей Задание

2. Процедура, переносящая в архив информацию о поставках товаров, которые были получены больше месяца назад. Архив должен содержать не значения ключей, а реальные названия комплектующих и поставщиков.
3. Процедура, выводящая на экран список комплектующих с учетом изменения цены:

   артикул1 название дата_поставки1 поставщик цена
   дата_поставки2 поставщик цена разница_с_прежней_ценой
   дата_поставки3 поставщик цена разница_с_прежней_ценой
   артикул2 название дата_поставки1 поставщик цена
   …

   Учитывать сведения из архива (поставки за последний год).

Задание №3. Создание триггеров.

1. Если при вводе данных дата поставки не указана, устанавливать текущую дату.
2. Проверка значений всех полей отношения "Поставки", для которых могут быть определены домены.
3. При изменении данных о комплектующих – копирование старых значений в специальную таблицу с указанием даты внесения изменений.

Вариант 15. БД фитнес-клуба.

Задание №1. Создание функций.

1. Функция, преобразующая числовое значение параметра ("Время начала занятий", например) в строку формата 'hh24:mi'.
2. Функция, возвращающая время окончания занятия в формате 'hh24:mi' по двум числовым параметрам: "Время начала занятий" и "Продолжительность".
3. Функция, возвращающая строку "абонемент закончился", если время действия абонемента на текущую дату закончилось.

Задание №2. Создание процедур.

1. Процедура, распечатывающая расписание занятий по номеру зала:

   Зал №___ Дата распечатки
   Понедельник Начало занятия1 – Окончание занятия1 – Вид занятия
   Начало занятия2 – Окончание занятия2 – Вид занятия
   Начало занятия3 – Окончание занятия3 – Вид занятия
   Вторник Начало занятия1 – Окончание занятия1 – Вид занятия
   Начало занятия2 – Окончание занятия2 – Вид занятия
   …

   Процедура должна проверять, что время разных занятий в одном зале не пересекается.
2. Процедура переноса сведений о клиентах в архив, если клиент перестал заниматься больше двух месяцев назад.
3. Процедура, проверяющая нагрузку тренеров. Если тренер ведет больше 3-х занятий в день, или время его занятий в разных залах совпадает, или перерыв между двумя занятиями меньше 15 минут или больше 3-х часов, то такие сведения выдаются на экран. Если таких ситуаций нет, процедура выдает "Без нарушений".

Задание №3. Создание триггеров.

1. Если при вводе данных не указана продолжительность занятий, устанавливать 1.5.
2. Проверка значений всех полей отношения "Клиенты", для которых могут быть определены домены.
3. Автоматизация переноса в архив всех изменений о клиентах (с указанием пользователя, вносящего изменения, и даты).