



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

**Рубцовский индустриальный институт (филиал)**  
федерального государственного бюджетного образовательного  
учреждения высшего образования  
«Алтайский государственный технический университет им. И.И. Ползунова»  
(РИИ АлтГТУ)

**Е.А. ДУДНИК**

## **ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ БАЗ ДАННЫХ**

Методические указания  
для самостоятельной работы студентам направления подготовки  
«Информатика и вычислительная техника»

Рубцовск 2021

## ББК 81.2

Дудник, Е.А. Информационная безопасность баз данных: методические указания для самостоятельной работы студентам направления подготовки «Информатика и вычислительная техника» /Е.А. Дудник; Рубцовский индустриальный институт. – Рубцовск: РИИ, 2021. – 32 с. [ЭР].

Данные учебно-методические рекомендации предназначены для студентов направлений подготовки «Информатика и вычислительная техника», изучающих дисциплину «Информационная безопасность баз данных». Представлен лабораторный практикум по вариантам.

Рассмотрены и одобрены на  
заседании кафедры ПМ Рубцовского  
индустриального института.  
Протокол № 9 от 18.03.2021 г.

## 1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

| Компетенция | Содержание компетенции  | Индикатор | Содержание индикатора   |
|-------------|---|-----------|---|
| ПК-8        | Способен обеспечивать информационную безопасность уровня баз данных | ПК-8.1    | Разрабатывает политику информационной безопасности на уровне БД |
|             |   | ПК-8.2    | Анализирует запросы к БД  |

## 2. Место дисциплины в структуре образовательной программы

|   |   |
|---|---|
| Дисциплины (практики), предшествующие изучению дисциплины, результаты освоения которых необходимы для освоения данной дисциплины.                 | Базы данных   |
| Дисциплины (практики), для которых результаты освоения данной дисциплины будут необходимы, как входные знания, умения и владения для их изучения. | Выпускная квалификационная работа, Преддипломная практика |

## 3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося

3.1 Для очного обучения: Общий объем дисциплины в з.е. /час: 3 / 108

Форма промежуточной аттестации: Зачет

| Форма обучения | Виды занятий, их трудоемкость (час.) |                     |                      |                        | Объем контактной работы обучающегося с преподавателем (час) |
|----------------|--------------------------------------|---------------------|----------------------|------------------------|---|
|                | Лекции                               | Лабораторные работы | Практические занятия | Самостоятельная работа |   |
| очная          | 16                                   | 16                  | 0                    | 76                     | 43  |

3.2 Для заочного обучения: Общий объем дисциплины в з.е. /час: 3 / 108

Форма промежуточной аттестации: Зачет

| Форма обучения | Виды занятий, их трудоемкость (час.) |                     |                      |                        | Объем контактной работы обучающегося с преподавателем (час) |
|----------------|--------------------------------------|---------------------|----------------------|------------------------|---|
|                | Лекции                               | Лабораторные работы | Практические занятия | Самостоятельная работа |   |
| заочная        | 6                                    | 6                   | 0                    | 96                     | 16  |

#### **4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий**

##### **4.1 Форма обучения: очная**

**Семестр: 8**

Лекционные занятия (16ч.)

1. Общие положения по информационной безопасности баз данных.(2ч.)[2,4,5,8] Понятие, виды и структура автоматизированных информационных систем.  
Общая характеристика составляющих, методов и механизмов обеспечения информационной безопасности баз данных.
2. Методы, модели и механизмы обеспечения конфиденциальности данных(2ч.)[2,4,5] Дискреционные и мандатные модели разграничения доступа к информации базы данных.  
Модели ролевого доступа и технологии рабочих групп пользователей.  
Тематическое разграничение доступа к информации в документальных базах данных
3. Методы, модели и механизмы обеспечения целостности данных(2ч.)[2,4,5] Механизмы обеспечения целостности данных, реализуемые реляционными СУБД. Модели обеспечения целостности данных в процессах коллективного доступа к разделяемым информационным ресурсам.  
Механизмы транзакций и обеспечение целостности данных в клиент-серверных СУБД
4. Методы, механизмы и технологии обеспечения сохранности и правомерной доступности информации в БД(2ч.)[2,4,5] Резервирование, архивирование и журнализация БД.  
Технологии репликации данных в распределенных БД
5. Критерии и стандарты информационной безопасности (защищенности)(2ч.)[2,4,5,8] Критерии оценки безопасности информационных технологий. Профили защиты СУБД. Разработка политики информационной безопасности на уровне БД.
6. Организационное обеспечение информационной безопасности(2ч.)[2,4,5] Администрирование и эксплуатация БД
7. Обеспечивание функционирование БД {лекция с разбором конкретных ситуаций} (2ч.)[2,4,8] Транзакции в СУБД.  
Хранимые процедуры в СУБД.  
Триггеры в СУБД.  
Представления в СУБД.
8. Анализ и оптимизация запросов к БД {лекция с разбором конкретных ситуаций} (2ч.)[1,6,7] Прототипы базы данных. Типы запросов. Оптимизация запросов к БД.

## Анализ запросов к БД.

### Лабораторные работы (16ч.)

1. На примере заданной предметной области разработать структуру таблиц БД. Учесть требования к разграничению доступа к информации, определить правила ввода входных данных(4ч.)[1,3,6,7]
2. Разработка модулей хранимых процедур. Проверка целостности данных(4ч.)[1,3,4,8]
3. Работа с транзакциями(4ч.)[1,3,4,8]
4. Разработка SQL запросов по заданной области автоматизации. Анализ и оптимизация запросов(4ч.)[1,2,6,7,8]

### Самостоятельная работа (76ч.)

1. Изучение литературных источников и информационных ресурсов(28ч.)[2,3,4] Изучения основ информационной безопасности на уровне баз данных. и способов анализирования запросов к базе данных.
2. Подготовка к защите лабораторных работ(16ч.)[1,3,6,7] Самостоятельное изучения материала для формирования навыков составления SQL запросов. Проведения анализа запросов к база данных и их оптимизация. Разработка приложений БД.
3. Выполнение расчетного задания(16ч.)[1,3,6] Разработка приложения к БД заданой предметной области, Организация разграничения доступа к информации. Ограничения и условия целостности данных., Разработка SQL запросов по заданной области автоматизации
4. Подготовка к зачету(16ч.)[2,3,4,5]

## 4.2 Форма обучения: заочная

*Семестр: 10*

### Лекционные занятия (6ч.)

1. Общие положения по информационной безопасности баз данных.(1ч.)[2,4,5,8] Понятие, виды и структура автоматизированных информационных систем. Общая характеристика составляющих, методов и механизмов обеспечения информационной безопасности баз данных.
2. Методы, модели и механизмы обеспечения конфиденциальности данных(1ч.)[2,4,5] Дискреционные и мандатные модели разграничения доступа к информации базы данных. Модели ролевого доступа и технологии рабочих групп пользователей. Тематическое разграничение доступа к информации в документальных базах данных

3. Методы, модели и механизмы обеспечения целостности данных(1ч.)[2,4,5]  
Механизмы обеспечения целостности данных, реализуемые реляционными СУБД.  
Модели обеспечения целостности данных в процессах коллективного доступа к разделяемым информационным ресурсам.

Механизмы транзакций и обеспечение целостности данных в клиент-серверных СУБД

4. Критерии и стандарты информационной безопасности (защищенности)(1ч.)[2,4,5,8] Критерии оценки безопасности информационных технологий. Профили защиты СУБД. Разработка политики информационной безопасности на уровне БД. Методы, механизмы и технологии обеспечения сохранности и правомерной доступности информации в БД

5. Обеспечивание функционирование БД {лекция с разбором конкретных ситуаций} (1ч.)[2,4,8] Транзакции в СУБД.

Хранимые процедуры в СУБД.

Триггеры в СУБД.

Представления в СУБД.

6. Анализ и оптимизация запросов к БД {лекция с разбором конкретных ситуаций} (1ч.)[6,7] Прототипы базы данных. Типы запросов. Оптимизация запросов к БД. Анализ запросов к БД.

Лабораторные работы (6ч.)

1. На примере заданной предметной области разработать структуру таблиц БД. Учесть требования к разграничению доступа к информации, определить правила ввода входных данных(2ч.)[6,7]

2. Разработка модулей хранимых процедур. Проверка целостности данных(1ч.)[4,8]

3. Работа с транзакциями(1ч.)[4,8]

4. Разработка SQL запросов по заданной области автоматизации. Анализ и оптимизация запросов(2ч.)[2,3,6,7,8]

Самостоятельная работа (96ч.)

1. Изучение литературных источников(28ч.)[2,3,4]

2. Подготовка к лабораторным работам(30ч.)[3,6,7] Формирования основ инженерных знаний

3. Выполнение письменной контрольной работа(30ч.)[3,6]

4. Подготовка к зачету(8ч.)[2,3,4,5]

## **5. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине**

Для каждого обучающегося обеспечен индивидуальный неограниченный доступ к электронно-библиотечным системам: Лань, Университетская библиотека

он-лайн, электронной библиотеке АлтГТУ и к электронной информационно-образовательной среде [1-5]:

1. Дудник, Е.А. Программирование. Часть III (лабораторный практикум по программированию в среде Delphi) [текст]: учебно-метод. пособие для студентов направления ИВТ/ Е.А. Дудник, Н.С. Зорина. - Рубцовск: РИИ, 2017. - 63 с. (6 экз.+ЭР)

## **6. Перечень учебной литературы**

### **6.1. Основная литература**

2. Девянин, П. Н. Модели безопасности компьютерных систем. Управление доступом и информационными потоками : учебное пособие / П. Н. Девянин. — 2-е изд., испр. и доп. — Москва : Горячая линия-Телеком, 2017. — 338 с. — ISBN 978-5-9912-0328-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/111049> (дата обращения: 13.01.2021). — Режим доступа: для авториз. пользователей.

3. Маркин А.В. Построение запросов и программирование на SQL : Учеб. пособие: [текст]//М.: Диалог-МИФИ, 2008.-318с. (30экз.)

### **6.2. Дополнительная литература**

4. Прохорова, О.В. Информационная безопасность и защита информации : учебник / О.В. Прохорова ; Самарский государственный архитектурно-строительный университет. – Самара : Самарский государственный архитектурно-строительный университет, 2014. – 113 с. : табл., схем., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=438331> (дата обращения: 25.01.2021). – Библиогр. в кн. – ISBN 978-5-9585-0603-3. – Текст : электронный.

5. Алгоритмы категорирования персональных данных для систем автоматизированного проектирования баз данных информационных систем : монография / А. В. Благодаров, В. С. Зияутдинов, П. А. Корнев, В. Н. Малыш. — Москва : Горячая линия-Телеком, 2015. — 116 с. — ISBN 978-5-9912-0307-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/111019> (дата обращения: 13.01.2021). — Режим доступа: для авториз. пользователей.

## **7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

6. <http://sql-tutorial.ru/> - интерактивный учебник по SQL

7. Дюбуа Поль MySQL [Электронный ресурс] : Учебное пособие/М.: Изд. Дом "Вильямс", 2001. - 816 с

8. Интернет-курс «Безопасность баз данных»  
[http://www.e-biblio.ru/book/bib/01\\_informatika/b\\_baz\\_dan/sg.html](http://www.e-biblio.ru/book/bib/01_informatika/b_baz_dan/sg.html)

## **8. Фонд оценочных материалов для проведения текущего контроля успеваемости и промежуточной аттестации**

Содержание промежуточной аттестации раскрывается в комплекте контролирующих материалов, предназначенных для проверки соответствия уровня подготовки по дисциплине требованиям ФГОС, которые хранятся на кафедре-разработчике РПД в печатном виде и в ЭИОС.

## **9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем**

Для успешного освоения дисциплины используются ресурсы электронной информационно-образовательной среды, образовательные интернет-порталы, глобальная компьютерная сеть Интернет. В процессе изучения дисциплины происходит интерактивное взаимодействие обучающегося с преподавателем через личный кабинет студента.

| <b>№пп</b> | <b>Используемое программное обеспечение</b> |
|------------|---|
| 1          | Dev-C++                                     |
| 2          | Lazarus                                     |
| 3          | LibreOffice                                 |
| 4          | Microsoft Access                            |
| 5          | Microsoft SQL Server Express                |
| 6          | Python                                      |
| 7          | Visual FoxPro Professional 9.0              |
| 8          | Windows                                     |
| 9          | Антивирус Kaspersky                         |
| 10         | Яндекс.Браузер                              |
| 11         | 7-Zip                                       |

| <b>№пп</b> | <b>Используемые профессиональные базы данных и информационные справочные системы</b>   |
|------------|--|
| 1          | Бесплатная электронная библиотека онлайн "Единое окно к образовательным ресурсам" для студентов и преподавателей; каталог ссылок на образовательные интернет-ресурсы ( <a href="http://Window.edu.ru">http://Window.edu.ru</a> )   |
| 2          | Национальная электронная библиотека (НЭБ) — свободный доступ читателей к фондам российских библиотек. Содержит коллекции оцифрованных документов (как открытого доступа, так и ограниченных авторским правом), а также каталог изданий, хранящихся в библиотеках России. ( <a href="http://нэб.рф/">http://нэб.рф/</a> ) |

## **10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

|  |
|--|
| <b>Наименование специальных помещений и помещений для самостоятельной работы</b> |
|--|

|  |
|--|
| <b>Наименование специальных помещений и помещений для самостоятельной работы</b> |
| помещения для самостоятельной работы   |
| учебные аудитории для проведения учебных занятий                                 |

Материально-техническое обеспечение и организация образовательного процесса по дисциплине для инвалидов и лиц с ограниченными возможностями здоровья осуществляется в соответствии с «Положением об обучении инвалидов и лиц с ограниченными возможностями здоровья».

**ПРИЛОЖЕНИЕ А**  
**ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**  
**ПО ДИСЦИПЛИНЕ «Информационная безопасность баз данных»**

**1. Перечень оценочных средств для компетенций, формируемых в результате освоения дисциплины**

| <b>Код контролируемой компетенции</b>                                     | <b>Способ оценивания</b> | <b>Оценочное средство</b>                     |
|---|--------------------------|---|
| ПК-8: Способен обеспечивать информационную безопасность уровня баз данных | Зачет                    | Комплект контролирующих материалов для зачета |

**2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания**

Оцениваемые компетенции представлены в разделе «Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций» рабочей программы дисциплины «Информационная безопасность баз данных».

При оценивании сформированности компетенций по дисциплине «Информационная безопасность баз данных» используется 100-балльная шкала.

| <b>Критерий</b>  | <b>Оценка по 100-балльной шкале</b> | <b>Оценка по традиционной шкале</b> |
|--|-------------------------------------|-------------------------------------|
| Студент освоил изучаемый материал, выполняет задания в соответствии с индикаторами достижения компетенций, может допускать отдельные ошибки.                 | 25-100                              | <i>Зачтено</i>                      |
| Студент не освоил основное содержание изученного материала, задания в соответствии с индикаторами достижения компетенций не выполнены или выполнены неверно. | 0-24                                | <i>Не зачтено</i>                   |

**3. Типовые контрольные задания или иные материалы, необходимые для оценки уровня достижения компетенций в соответствии с индикаторами**

*1. Типовое задания на разработку политики безопасности БД*

| <b>Компетенция</b>   | <b>Индикатор достижения компетенции</b>                                |
|--|--|
| ПК-8 Способен обеспечивать информационную безопасность уровня баз данных | ПК-8.1 Разрабатывает политику информационной безопасности на уровне БД |

1. На основе анализа предметной области разработать политику информационной безопасности баз данных.

а) Предметная область: доставка на склад товаров поставщиками.

*Объект 1. ТОВАРЫ: наименование товара характеристики товара: номер договора поставки; поставщик; единица измерения товара; цена за единицу; количество;*

*Объект 2. ПОСТАВЩИКИ: название фирмы; адрес; телефон(ы); ФИО директора; ИНН.*

б) Описать концептуальную модель в виде ER- диаграммы логическую модель БД в виде схемы БД. Определить ограничения целостности БД для хранимых данных.

в) Спроектировать триггеры по одному на каждое из событий Insert, Update, Delete. Для триггера обязательно указать событие, на которое он срабатывает, время инициирования, дать описание.

д) Спроектировать хранимую процедуру. Определить входные и выходные параметры, их типы. Дать словесное описание действий, выполняемых хранимой процедурой.

## 2. Типовое задание на анализ SQL запроса к БД

| Компетенция  | Индикатор достижения компетенции |
|--|----------------------------------|
| ПК-8 Способен обеспечивать информационную безопасность уровня баз данных | ПК-8.2 Анализирует запросы к БД  |

1. Предметная область: доставка на склад товаров поставщиками.

*Объект 1. ТОВАРЫ: наименование товара характеристики товара: номер договора поставки; поставщик; единица измерения товара; цена за единицу; количество;*

*Объект 2. ПОСТАВЩИКИ: название фирмы; адрес; телефон(ы); ФИО директора; ИНН.*

2. Применяя язык манипулирования данными SQL сформулировать запросы и сделать анализ запросов к БД:

а) Выявить самого активного поставщика, доставившего наибольшее количество товаров за прошедший месяц.

б) Выявить товар и поставщика, средняя цена доставленного товара была наименьшая за прошедший квартал.

## 7. Лабораторный практикум

### ЗАДАНИЕ НА ВЫПОЛНЕНИЕ ЛАБОРАТОРНЫХ РАБОТ:

Лабораторная работа №1. Создание функций на PL/SQL.

Функции предназначены для использования в SQL-предложениях, обращающихся к соответствующим отношениям. Они, по возможности, не должны включать обращения к таблицам БД.

Лабораторная работа №2. Создание процедур на PL/SQL.

Процедуры запускаются из командной строки. Они предназначены для обработки таблиц, созданных в первом семестре. Использование функций из предыдущей лабораторной работы обязательно в тех случаях, когда это оговаривается явно. Процедуры должны включать блоки обработки исключительных ситуаций. Отсутствие данных в обрабатываемой таблице считать исключительной ситуацией.

Лабораторная работа №3. Создание триггеров.

Лабораторная работа №4. Работа со средствами динамического SQL.

#### Вариант 1. БД отдела кадров.

Л.р. №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 значение ФИО в фамилию с инициалами (например, "Иванов Иван Сергеевич" в "Иванов И.С."). При невозможности преобразования функция возвращает строку '#####'.
3. Функция, определяющая количество полных лет и месяцев, прошедших между двумя датами. Если вторая дата не указана, берется текущая дата. Функция вызывается для даты приёма на работу.

Л.р. №2. Создание процедур.

1. Процедура, выводящая список всех сотрудников – юбиляров текущего года (с указанием даты юбилея и возраста). Использовать результаты лаб. работы №1.
2. Процедура, выводящая список всех должностей и сотрудников в виде:

```
отдел1должность1 фамилия1   ставка
фамилия2   ставка
вакантно   количество вакантных ставок
должность2 фамилия1   ставка
фамилия2   ставка
фамилия3   ставка
вакантно   количество вакантных ставок
```

...

```
отдел2должность1 фамилия1   ставка
```

...

3. Процедура, выводящая сведения о превышении установленного количества ставок по номеру отдела. Если таких случаев не обнаружено, процедура должна выводить сообщение об этом.

Л.р. №3. Создание триггеров.

1. Реализация ограничения внешнего ключа.
2. Проверка значений всех полей отношения "Сотрудники", для которых могут быть определены домены (в частности, количество ставок кратно 0.25 и не может превышать 1.5, дата поступления на работу не может быть больше текущей).

3. 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).

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

**Л.р. №4. Работа со средствами динамического SQL.**

Создать процедуру, которая выводит в специальную таблицу информацию о количестве записей и средней длине записи во всех таблицах, принадлежащих пользователю, от имени которого запускается эта процедура. Таблица должна принадлежать пользователю, от имени которого запускается процедура. Данные берутся из представления USER\_TABLES. Если данные по анализу таблицы устарели или отсутствуют, процедура сначала запускает команду ANALYZE.

## **Вариант 2. БД бухгалтерии.**

**Л.р. №1. Создание функций.**

1. Функция расчета стажа работы. Принимает 2 аргумента: стаж на прежних местах работы и дату поступления на данную работу.

2. Функция, возвращающая отчество из строки ФИО или строку '#####', если она не смогла выделить отчество.

3. Функция, определяющая, является ли человек пенсионером по полу и дате рождения. Возвращает строку "пенсионер" или пустую строку.

**Л.р. №2. Создание процедур.**

1. Процедура, повышающая заработную плату сотрудников:

- до размера в 1.5 прожиточного минимума, если оклад меньше прожиточного минимума;
- на 5%, если оклад больше, чем 1.5 прожиточного минимума;
- на 10%, если работник является начальником подразделения.

Входной параметр – прожиточный минимум.

2. Процедура расчета зарплаты в виде:

отдел1 фамилия1 сумма

2. фамилия2 сумма

...

отдел21 фамилия1 сумма

2. фамилия2 сумма

...

Сумма рассчитывается как (оклад-13%). Сотрудникам, чей стаж работы на данном предприятии превышает 10 лет, выплачивается надбавка (от 10 до 15 лет – 10% от оклада, от 15 до 20 лет – 20% от оклада, свыше 20 лет – 30% от оклада). Использовать функцию из предыдущей лабораторной работы, в качестве первого аргумента передавать 0.

3. Процедура начисления премии. Входной параметр – общий размер премиального фонда. Размер премии зависит от оклада и надбавки, надбавка зависит от стажа работы на данном предприятии:

- 5% от оклада, если стаж от 1 до 5 лет;
- 10% от оклада, если стаж от 5 до 10 лет;
- 30% от оклада, если стаж от 10 до 20 лет;
- 50% от оклада, если стаж от 20 до 30 лет;
- 100% от оклада, если стаж свыше 30 лет.

Премиальный фонд распределяется пропорционально полученным значениям (оклад + надбавка). Использовать функцию из предыдущей лабораторной работы.

### Л.р. №3. Создание триггеров.

1. Реализация ограничения внешнего ключа.
2. 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)).
3. Установка значения поля "пол", если оно не установлено. Правила: если отчество оканчивается на '-ВНА', то пол женский, если на '-ВИЧ', то мужской. В противном случае триггер должен генерировать ошибку.
4. Регистрация изменений, вносимых в таблицу "Сотрудники" (дублирование старой записи в специальной таблице с указанием даты изменения и пользователя, который их проводит).

### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая принимает в качестве параметров имя таблицы и имена 4-х полей в этой таблице. Первое поле она интерпретирует как ФИО, разбивает его на составляющие и заполняет три оставшихся поля. Если значение первого поля не может быть правильно проинтерпретировано как ФИО (отсутствует отчество, имя и отчество или в строке встречаются недопустимые символы), она помещает в специальную таблицу это значение и соответствующее значение ключа базы данных (ROWID).

## Вариант 3. БД деканата (Сессия).

### Л.р. №1. Создание функций.

1. Функция, возвращающая по дате строку, в которой указаны время начала и завершения экзамена или консультации. 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, преобразующая две строки формата 'DD.ММ.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 фамилию с инициалами по значению ФИО или исходную строку, если строка ФИО содержит менее трех элементов.

### Л.р. №2. Создание процедур.

1. Процедура, выводящая расписание сессии. Список должен выглядеть так:

```
факультет1
группа1  дисциплин1 препод. консультация (дата, время, ауд.) экз. (дата, время, ауд.)
          дисциплин2 препод. консультация (дата, время, ауд.) экз. (дата, время, ауд.)
          дисциплин3 препод. консультация (дата, время, ауд.) экз. (дата, время, ауд.)
группа2  дисциплин1 препод. консультация (дата, время, ауд.) экз. (дата, время, ауд.)
          дисциплин2 препод. консультация (дата, время, ауд.) экз. (дата, время, ауд.)
...
факультет2
...
```

Использовать функции из первой лабораторной работы.

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 1000 руб., доцент – 800 руб., старший преподаватель – 500 руб.

### Л.р. №3. Создание триггеров.

1. Реализация ограничения внешнего ключа.
2. Проверка значений всех полей отношения "Сессия", для которых могут быть определены домены.
3. 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.
4. Проверка: консультация по каждому предмету у каждой группы назначена раньше, чем экзамен (триггер уровня предложения).

### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая принимает в качестве параметров имя таблицы и имя поля в этой таблице. Поле имеет символьный тип с маской '\_ \_ / \_ \_ / \_ \_ \_ \_'. Процедура исправляет неправильные даты:

- если неверно указано число, она заменяет его на '01';
- если неверно указан месяц, она заменяет его на '01';
- если неверно указан год (больше текущего или меньше 1900), она заменяет его на '1900'.

## Вариант 4. БД спортивного клуба.

### Л.р. №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 строку "подходит по возрасту", если спортсмен может участвовать в соревновании, и null в противном случае. 2 параметра: ограничение по возрасту и дата рождения спортсмена.
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.

### Л.р. №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
```

...

Использовать функцию из лабораторной работы №1.

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.

### Л.р. №3. Создание триггеров.

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

4. Реализация ограничения внешнего ключа.

#### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая принимает в качестве параметра имя таблицы и имя поля в этой таблице. Процедура выводит на экран статистику по этой таблице: количество записей в таблице – имя поля – количество различных значений поля – количество null-значений. Статистика подсчитывается в этой процедуре.

### Вариант 5. БД диссертаций.

#### Л.р. №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 интересуется возраст. Если вторая дата не указана, то возраст на текущую дату. Вызывать функцию для двух полей: дата рождения автора и дата защиты диссертации.

#### Л.р. №2. Создание процедур.

1. Процедура, выдающая отчет по диссертациям, защищенным в определенном году:

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. Использовать функцию из лабораторной работы №1.

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'). Использовать функцию из лабораторной работы №1.

#### Л.р. №3. Создание триггеров.

1. Реализация ограничений внешнего ключа.
2. 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).
3. Замена при добавлении данных сокращенных значений поля "Тип" отношения "Диссертации" на полные ("канд" на "кандидатская" и т.п.).
4. Триггер, переносящий в архив (в специальную таблицу) изменения сведений об авторах.

#### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая принимает в качестве параметра имя таблицы. Процедура выводит на экран информацию обо всех таблицах, связанных с указанной таблицей по внешнему ключу и принадлежащих пользователю, от имени которого запускается эта процедура. Эта информация включает в себя имя таблицы, общее количество записей и количество различных значений внешнего ключа.

#### Вариант 6. БД больницы.

##### Л.р. №1. Создание функций.

1. Функция, возвращающая строку "больше месяца", если со времени поступления пациента прошло более 1 месяца. 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, 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 и дата поступления пациента.

##### Л.р. №2. Создание процедур.

1. Процедура, распределяющая больных временно отсутствующего лечащего врача по другим лечащим врачам в соответствии со специализацией врачей. Процедура не должна допускать перекося 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.

##### Л.р. №3. Создание триггеров.

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

#### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая принимает в качестве параметра имя таблицы. Процедура выводит на экран информацию обо всех таблицах, связанных с указанной таблицей по внешнему ключу и принадлежащих пользователю, от имени которого запускается эта процедура. Эта информация включает в себя имя таблицы, количество различных значений внешнего ключа и среднее количество записей на одно значение внешнего ключа.

#### Вариант 7. БД собственников квартир.

##### Л.р. №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 соответствует 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.").

#### Л.р. №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 время более 10 собственников или квартир, у которых с начала первого владения более 20 собственников, в том числе, бывших.

#### Л.р. №3. Создание триггеров.

1. Реализация ограничения внешнего ключа.
2. Триггер, проверяющий, что дата окончания владения или не определена, или больше даты начала владения.
3. Триггер, переносащий в архив все записи, удаляемые из таблицы "Владение".
4. 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).

#### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая делает копию указанной таблицы, добавляя в неё поле 'идентификатор' с типом number(N). Параметры: имя исходной таблицы, имя результирующей таблицы, имя добавляемого поля, размер N. Процедура копирует структуру и данные исходной таблицы, проставляя в качестве значения поля 'идентификатор' целые числа, начиная с 1.

### **Вариант 8. БД транспортного предприятия.**

#### Л.р. №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.").

#### Л.р. №2. Создание процедур.

1. Процедура, выдающая расписание работы водителя на текущий месяц. Использовать функции из лабораторной работы №1.
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, добавляющая рейсы на указанную во входном параметре дату в таблицу "Рейсы". Поле "Водитель" остается пустым.

#### Л.р. №3. Создание триггеров.

1. Реализация ограничения внешнего ключа.
2. Триггер, проверяющий, что количество проданных на рейс билетов не превышает количество мест в автобусе.
3. Проверка значений всех полей отношения "Маршруты", для которых могут быть определены домены.
4. Триггер, переносящий сведения об удаляемых рейсах в архив (в специальную таблицу).

#### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая выводит в специальную таблицу информацию о количестве записей и средней длине записи во всех таблицах, принадлежащих пользователю, от имени которого запускается эта процедура. Таблица должна принадлежать пользователю, от имени которого запускается процедура. Данные берутся из представления USER\_TABLES. Если данные по анализу таблицы устарели или отсутствуют, процедура сначала запускает команду ANALYZE.

### Вариант 9. БД библиотеки.

#### Л.р. №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 '#####'.

#### Л.р. №2. Создание процедур.

1. Процедура списания книг, изданных более 50 лет назад (перенос в архив). При этом удаляются записи из абонементов читателей.
2. Процедура вывода списка книг по указанной рубрике в виде:  
 Название рубрики:

|                 |           |             |               |                  |
|-----------------|-----------|-------------|---------------|------------------|
| 1. автор1       | название1 | год выпуска | место издания | [старое издание] |
| автор2          |           |             |               |                  |
| 2. (без автора) | название2 | год выпуска | место издания | [старое издание] |
| 3. автор1       | название3 | год выпуска | место издания | [старое издание] |
| автор1          | название4 | год выпуска | место издания | [старое издание] |
| автор2          |           |             |               |                  |
| автор3          |           |             |               |                  |

...

Использовать функции из лаб. работы №1.

3. Процедура добавления в специальную таблицу "Оповещения" сообщений для читателей-должников. Таблица содержит следующие данные: ФИО, e-mail, количество взятых книг, последний срок их возврата и сообщение вида "Вы взяли в библиотеке [название

библиотеки] следующие книги: (перечень книг через запятую с указанием авторов и названия) и не вернули их в срок до <дата>". Срок возврата книг – 30 дней со дня выдачи. Если информация для данного читателя уже есть, изменяется только список взятых книг, для которых прошел срок возврата.

#### Л.р. №3. Создание триггеров.

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

#### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая принимает в качестве параметров имя таблицы и имена двух полей в этой таблице. Первое поле имеет символьный тип с маской 'DD.MM.YYYY' и интерпретируется как дата. Во второе поле эта дата проставляется в формате 'YYYY/MM/DD'. Если дата неправильная, то второму полю присваивается значение null.

### Вариант 10. "Журнальные публикации" из БД библиотеки.

#### Л.р. №1. Создание функций.

1. Функция, возвращающая пустую строку или строку "старая публикация" для статей, опубликованных более 10 лет назад или более пяти лет назад, если они относятся к рубрикам по общественным наукам (код начинается с '30'). Параметры: год издания и шифр рубрики.
2. Функция, возвращающая количество авторов статьи по значению поля "Авторы".
3. Функция, возвращающая количество страниц на основании значения поля "Страницы". Если значение имеет вид 'с.х', возвращается 1. Если обнаружена ошибка, то возвращается -1.

#### Л.р. №2. Создание процедур.

1. Процедура, распечатывающая список статей определенного автора, у которого вышло не менее N статей (возможно, в соавторстве), с указанием названия статьи, названия журнала, рубрики и соавторов:

Список публикаций <Автора>

|                      |        |         |                               |
|----------------------|--------|---------|-------------------------------|
| название публикации1 | журнал | рубрика | количество страниц [соавтор1] |
| [соавтор2]           |        |         |                               |
| название публикации2 | журнал | рубрика | количество страниц [соавтор1] |

...

N по умолчанию равно 1. Использовать функцию из лаб. работы №1.

2. Процедура, распечатывающая список статей по определенной теме в виде:

Тема "название\_темы"

|            |            |            |        |                     |
|------------|------------|------------|--------|---------------------|
| журнал1    | номер1     | 1. статья1 | авторы | [старая публикация] |
| 2. статья2 | авторы     |            |        | [старая публикация] |
| журнал1    | номер1     | 1. статья1 | авторы | [старая публикация] |
| номер2     | 1. статья1 | авторы     |        | [старая публикация] |
| 2. статья2 | авторы     |            |        | [старая публикация] |

...

3. Процедура переноса в основные таблицы БД данных из таблицы "Новые поступления" ("Название журнала", "Год", "Номер выпуска", "Название рубрики", "Авторы", "Название публикации", "Страница (начало публикации)", "Количество страниц", "Примечание").

#### Л.р. №3. Создание триггеров.

1. Реализация ограничения внешнего ключа.
2. Проверка: поле "Страницы" должно содержать значение типа "х-у", причем х должен быть меньше у. Если х равен у, то значение преобразуется к виду 'с.х'.

3. Фиксация всех изменений (по UPDATE и DELETE) отношения "Рубрикатор" в отдельной таблице (с указанием пользователя и даты внесения изменений).

#### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая принимает в качестве параметров имя существующей таблицы, имя новой таблицы и режим. Процедура создает копию существующей таблицы с новым именем. В зависимости от значения последнего параметра процедура либо выдает ошибку в том случае, если существует объект, совпадающий с указанным именем новой таблицы, либо удаляет его и создает копию вместо него.

### Вариант 11. БД гостиницы.

#### Л.р. №1. Создание функций.

1. Функция расчета стоимости оказанных услуг в зависимости от класса номера:

1-й класс – на 5% выше базовой стоимости,

"полулюкс" – на 10%,

"люкс" – на 20%.

Поместить справочные данные о процентах в отдельную таблицу. Параметры – класс номера и стоимость услуги.

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 количество дней, прожитых постояльцем в гостинице на основании двух дат (день приезда – день отъезда считать как один день). Если второй параметр не определен, считать до текущей даты.

#### Л.р. №2. Создание процедур.

1. Процедура подготовки счёта клиенту за предоставленные услуги (с использованием функций из лабораторной работы № 1). Параметры – ФИО и паспортные данные клиента. Если в один день клиенту было оказано несколько одинаковых услуг, выводить их одной строкой, указывая количество.

2. Процедура переноса данных об услугах, оказанных выбывшим клиентам, в архив. Если в один день клиенту было оказано несколько одинаковых услуг, в архивную таблицу добавлять их одной строкой, указывая количество.

3. Процедура, распечатывающая списки постояльцев, проживающих в данный момент в гостинице:

|        |          |               |                |               |
|--------|----------|---------------|----------------|---------------|
| номер1 | фамилия1 | дата вселения | номер паспорта | [организация] |
|--------|----------|---------------|----------------|---------------|

|          |               |                |               |
|----------|---------------|----------------|---------------|
| фамилия2 | дата вселения | номер паспорта | [организация] |
|----------|---------------|----------------|---------------|

|        |          |               |                |               |
|--------|----------|---------------|----------------|---------------|
| номер2 | фамилия1 | дата вселения | номер паспорта | [организация] |
|--------|----------|---------------|----------------|---------------|

|          |               |                |               |
|----------|---------------|----------------|---------------|
| фамилия2 | дата вселения | номер паспорта | [организация] |
|----------|---------------|----------------|---------------|

|          |               |                |               |
|----------|---------------|----------------|---------------|
| фамилия3 | дата вселения | номер паспорта | [организация] |
|----------|---------------|----------------|---------------|

...

#### Л.р. №3. Создание триггеров.

1. Реализация ограничения внешнего ключа.
2. Копирование в архив всех изменений таблицы "Услуги". Указывать в архиве дату внесения изменений и имя пользователя, который вносил изменения.
3. Проверка значений всех полей отношения "Постояльцы", для которых могут быть определены домены.
4. Триггер, устанавливающий значение поля "дата выселения" как текущая дата при изменении поля "Признак" на 'выбыл'.

#### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая принимает в качестве параметра имя таблицы и выводит на экран информацию о количестве записей и средней длине записи в этой таблице. Таблица должна принадлежать пользователю, от имени которого запускается процедура. Данные берутся из

представления USER\_TABLES. Если данные по анализу таблицы устарели или отсутствуют, процедура сначала запускает команду ANALYZE.

## **Вариант 12. БД адвоката.**

### **Л.р. №1. Создание функций.**

1. Функция, возвращающая длительность периода, прошедшего между двумя датами. Длительность указывается в годах, месяцах и днях. Параметры – две даты. Если вторая дата не определена, то считать ее равной текущей дате.
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 "несовершеннолетний", между двумя датами прошло менее 18-и лет, или пустую строку. Параметры – две даты. Если вторая дата не определена, то считать ее равной текущей дате.

### **Л.р. №2. Создание процедур.**

1. Процедура удаления законченного дела в архив. Параметр – номер дела. Процедура должна проверять, что дело закончено.
2. Процедура, распечатывающая список подзащитных по законченным делам за текущий год, в виде:

```
дело1 дата начала – дата окончания (длительность)
фамилия1 год рожд. статьи приговор срок [несовершеннолетний]
дело2 дата начала – год окончания (длительность)
фамилия1 год рожд. статьи приговор срок [несовершеннолетний]
фамилия2 год рожд. статьи приговор срок [несовершеннолетний]
фамилия3 год рожд. статьи приговор срок [несовершеннолетний]
...
```

Использовать функцию из лабораторной работы № 1. Статьи указывать через запятую.

3. Процедура формирования отчета об эффективности защиты по законченным делам за указанный год:

Отчет за \_\_\_\_ год:

```
дело1 фамилия1 статьи мин. срок макс. срок срок по приговору
фамилия2 статьи мин. срок макс. срок срок по приговору
дело2 фамилия1 статьи мин. срок макс. срок срок по приговору
фамилия2 статьи мин. срок макс. срок срок по приговору
...
```

Общее количество подзащитных

Количество оправданных подзащитных

Количество подзащитных, осужденных условно

Параметр – год, за который требуется отчет. Значение "максимальный срок" брать как максимальный срок из всех статей, по которым обвинялся подзащитный, значение "минимальный срок" брать как минимальный срок из всех его статей.

### **Л.р. №3. Создание триггеров.**

1. Реализация ограничения внешнего ключа.
2. Проверка значений всех полей отношения "Клиенты", для которых могут быть определены домены.
3. Если на момент начала ведения дела обвиняемому меньше 18 лет, добавлять в поле "Примечание" 'несовершеннолетний' (если это не указано).
4. Устанавливать для оправданных срок по приговору равным 0.

### **Л.р. №4. Работа со средствами динамического SQL.**

Создать процедуру, которая принимает в качестве параметров имя таблицы и список полей (одним параметром). По этому списку она создает индекс. Проверки: все поля – разные, индекса с такими же лидирующими полями для данной таблицы нет.

### Вариант 13. БД складского предприятия.

#### Л.р. №1. Создание функций.

1. Функция, возвращающая пустую строку или строку "остатки", если товара осталось меньше 10 единиц.
2. Функция, возвращающая количество дней, прошедших между двумя датами. Если первая дата больше второй, функция возвращает -1.
3. Функция, приводящая строку к формату телефонного номера: '\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-'. Исходная строка может содержать от 0 до 3 дефисов, например: 1234567, 123-4567 или 123-45-67. Если строка содержит 7 цифр, то функция должна разделить их двумя дефисами; если в строке 10 цифр, то еще добавить в начало '8-'; если 11 цифр – привести к указанному формату. Если строка не может быть корректно приведена к указанному формату, то функция должна возвращать NULL.

#### Л.р. №2. Создание процедур.

1. Процедура формирования по заказу бланка с расположением товара:

Заказ № <номер\_заказа>

Товар склад1            линия1            количество\_товара

склад2            линия1            количество\_товара

линия2            количество\_товара

Итого: общее\_количество\_товара

Общий\_вес\_товара

В ситуации, когда на складе в одном месте лежит ровно столько товара, сколько заказано, бланк должен содержать одну строку (плюс заголовок и итоги).

2. Процедура, создающая отчет о движении товаров за последний период:

Отчет за период с <...> по <...>

1. товар1            поступление (ед.изм.)            количество реализованного

2. товар2            поступление (ед.изм.)            количество реализованного

...

Параметр: длительность периода (в днях).

3. Процедура оформления накладных на поставку товаров, запасы которых подходят к концу:

Накладная номер <номер1>

Поставщик1, адрес поставщика, телефон

Товар1, количество, текущая дата

Накладная номер <номер2>

Поставщик2, адрес поставщика, телефон

Товар2, количество, текущая дата

Если поставщиков у данного товара несколько, в конце накладной приводится список альтернативных поставщиков. Номер накладной берется из последовательности (sequence).

В накладную включаются товары:

- суммарное количество которых меньше, чем реализованное за последнюю неделю;
- которых не хватает для выполнения текущих заказов.

Объем поставки определяется в первом случае как количество товара, реализованного за последнюю неделю; во втором случае как количество заказанного товара плюс количество товара, реализованного за последнюю неделю.

#### Л.р. №3. Создание триггеров.

1. Реализация ограничения внешнего ключа.
2. Если при вводе данных дата поставки не указана, устанавливать текущую дату.
3. Перенос в архив сведений о поставках товаров, которые уже реализованы (остаток равен 0).

4. При изменении данных о товарах – копирование старых значений в специальную таблицу. Сохранять в этой таблице дату изменения.

#### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая выводит на экран информацию о количестве записей во всех таблицах, принадлежащих пользователю, от имени которого запускается эта процедура.

### Вариант 14. БД магазина.

#### Л.р. №1. Создание функций.

1. Функция, возвращающая пустую строку или строку "закончился срок реализации", если дата поставки плюс срок реализации больше текущей даты и времени. Параметры: дата и срок реализации.
2. Функция, возвращающая количество часов, оставшихся до завершения срока реализации товара. Параметры: дата поставки и срок реализации.
3. Функция, возвращающая новую цену товара, если до завершения срока реализации осталось менее 3-х часов для скоропортящегося товара и менее одного дня для всех остальных. Цена уменьшается на 20%, но не более чем на 50 рублей. Использовать предыдущую функцию. Параметры: цена товара, дата поставки и срок реализации.

#### Л.р. №2. Создание процедур.

1. Процедура, формирующая заказ на товары, которых осталось меньше необходимого запаса.
2. Процедура, распечатывающая список товаров с их общим количеством, старой и новой ценой (для скоропортящихся товаров) по категориям:

|            |           |            |             |              |
|------------|-----------|------------|-------------|--------------|
| категория1 | 1. товар1 | количество | старая_цена | [новая_цена] |
| 2. товар2  |           | количество | старая_цена | [новая_цена] |
| категория2 | 1. товар1 | количество | старая_цена | [новая_цена] |
| 2. товар2  |           | количество | старая_цена | [новая_цена] |

...

3. Процедура, переносящая в архив информацию о поставках товаров, которые уже реализованы. Архив должен содержать не значения ключей, а реальные названия товаров, поставщиков и категорий.

#### Л.р. №3. Создание триггеров.

1. Реализация ограничений внешнего ключа.
2. Если при вводе данных дата поставки не указана, устанавливать текущую дату.
3. Проверка значений всех полей отношения "Поставки", для которых могут быть определены домены.
4. При изменении данных о товарах – копирование старых значений в специальную таблицу.

#### Л.р. №4. Работа со средствами динамического SQL.

Создать функцию, которая принимает в качестве параметра имя таблицы и имя поля в этой таблице и возвращает среднее арифметическое по этому полю. В том случае, если тип поля не позволяет посчитать среднее арифметическое, функция должна возвращать null.

### Вариант 15. БД деканата.

#### Л.р. №1. Создание функций.

1. Функция, возвращающая строку "не имеет", если значение поля "Ученое звание" – null.
2. Функция, преобразующая значение ФИО в фамилию с инициалами (например, "Иванов Иван Сергеевич" в "Иванов И.С.").
3. Функция, возвращающая по параметрам "время" и "тип занятия" время окончания занятия, например:
  - для лекции в '09.00' – '10.35', т.е. полтора часа занятие и 5 минут перерыв;
  - для семинара – то же самое;
  - для лабораторной работы в '09.00' – '10.30', т.е. полтора часа без перерыва.

Время возвращается как строка формата 'hh.mi', hh – часы, mi – минуты.

#### Л.р. №2. Создание процедур.

1. Процедура, формирующая на основании таблицы "Преподаватели" первые 4 поля таблицы "Научные кадры" (Id, "ФИО", "Ученая степень", "Кафедра", "Специальность", "Дата присвоения степени", "Шифр ученого совета"). Она добавляет преподавателей, получивших новые ученые звания, и изменяет ученое звание сотрудника с кандидата наук на доктора наук, если степени получены по одной и той же специальности.
2. Процедура, принимающая в качестве параметра название кафедры и выдающая для всех преподавателей этой кафедры расписание в следующем виде:

преподаватель1      дисциплина1 тип занятия   группа1    день недели    время    ауд.  
группа2

преподаватель1      дисциплина2 тип занятия   группа1    день недели    время    ауд.  
группа2

группа3

преподаватель2      дисциплина1 тип занятия   группа1    день недели    время    ауд.

...

Время вывести интервалом с использованием функции из лабораторной работы №1.

3. Процедура, проверяющая корректность расписания для преподавателей. Расписание считается некорректным, если у одного преподавателя более 3-х занятий в день, а также если время разных занятий у одного преподавателя совпадает или пересекается.

#### Л.р. №3. Создание триггеров.

1. Реализация ограничений внешнего ключа.
2. Проверка значений всех полей отношения "Расписание", для которых могут быть определены домены. (В частности, занятия не могут начинаться раньше 9.00 и заканчиваться позднее, чем в 18.00).
3. Триггер, проверяющий соответствие "должности" и "ученой степени": для должности 'профессор' необходима степень 'доктор наук', для должности 'доцент' – 'кандидат наук'. (Проверяется правильность назначения на должность).
4. Триггер, копирующий в архив все данные об изменениях в таблице "Преподаватели" с указанием даты изменения и пользователя, который эти изменения внес.

#### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая принимает в качестве параметра имя таблицы. Процедура выводит на экран информацию обо всех индексах, связанных с указанной таблицей, в виде:    имя\_индекса (имя\_поля1,..., имя\_поляN).

### Вариант 16. БД кинофильмов.

#### Л.р. №1. Создание функций.

1. Функция расчета примерного возраста актера на момент съемки в фильме. Параметры – дата рождения актера и год выхода фильма.
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 продолжительность фильма в виде строки "X ч. Y м." по значению числового поля "Продолжительность".

#### Л.р. №2. Создание процедур.

1. Процедура, выводящая список фильмов, в которых режиссер является одновременно исполнителем одной из главных ролей, с указанием фамилии режиссера и роли (ролей), которую он сыграл.

2. Процедура, принимающая в качестве параметра название фильма и выводящая список других фильмов, в которых режиссер указанного фильма является исполнителем одной из главных ролей или сценаристом.
3. Процедура, распечатывающая список участников указанного фильма в виде:

Название фильма (год выхода)

Жанр

Режиссер: ФИО режиссера

Сценарист: ФИО сценариста

Оператор: ФИО оператора

Композитор: ФИО композитора

Актеры:

ФИО актера1 (роль)

ФИО актера2 (роль)

...

Если режиссеров, операторов и проч. несколько, выводить их через запятую.

### Л.р. №3. Создание триггеров.

1. Реализация ограничений внешнего ключа.
2. Если при вводе данных год выхода фильма не указан, устанавливать текущий год. Если указан год больше текущего – генерировать ошибку.
3. Преобразование содержимого поля "Страна" следующим образом:

РФ, Российская Федерация – в "Россия";

Великобритания – в "Англия";

Голландия – в "Нидерланды";

КНР – в "Китай".

4. При удалении данных о фильмах записывать их в архив (специальную таблицу).

### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая принимает в качестве параметра имя таблицы и имена двух полей в этой таблице и добавляет содержание первого поля к содержанию второго через запятую. Например, если значения поле1 = 'Иванов', поле2 = 'Петров', то после запуска этой процедуры значение поля2 = 'Петров, Иванов'. Если поле2 пустое, то просто копировать поле1 в поле2.

## Вариант 17.

### Л.р. №1. Создание функций.

- Функция расчета возраста в годах и месяцах. Параметры – дата рождения собаки и дата, на которую нужен возраст. Если вторая дата не указана, считать до текущей даты.
- Функция, преобразующая значение ФИО в фамилию с инициалами (например, "Иванов Иван Сергеевич" в "Иванов И.С."). При невозможности преобразования функция возвращает строку '#####'.
- Функция, приводящая строку к формату телефонного номера: '\_-\_-\_-\_-\_-\_-\_-\_-'. Исходная строка может содержать от 0 до 3 дефисов, например: 1234567, 123-4567 или 123-45-67. Если строка содержит 7 цифр, то функция должна разделить их двумя дефисами; если в строке 10 цифр, то еще добавить в начало '8-'; если 11 цифр – привести к указанному формату. Если строка не может быть корректно приведена к указанному формату, то функция должна возвращать NULL.

### Л.р. №2. Создание процедур.

1. Процедура, распечатывающая список собак определенной породы:

Название породы

Дата отчета

Кличка1 пол возраст [ награда1 дата получения ]



1. Реализация ограничений внешнего ключа.
2. Если при вводе данных не указано количество порций, устанавливать 1.
3. Проверка значений всех полей отношения "Блюда", для которых могут быть определены домены.
4. Автоматизация переноса в архив данных о блюде при удалении данных из основной таблицы.

#### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая принимает в качестве параметра имя таблицы и выводит на экран информацию о количестве записей и средней длине записи в этой таблице. Таблица должна принадлежать пользователю, от имени которого запускается процедура. Данные берутся из представления USER\_TABLES. Если данные по анализу таблицы устарели или отсутствуют, процедура сначала запускает команду ANALYZE.

### Вариант 19.

#### Л.р. №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.

#### Л.р. №2. Создание процедур.

1. Процедура, распечатывающая расписание занятий по номеру зала:

| Зал №       | Дата распечатки   |
|-------------|---|
| Понедельник | Начало занятия1 – Окончание занятия1 – Группа<br>Начало занятия2 – Окончание занятия2 – Группа<br>Начало занятия3 – Окончание занятия3 – Группа |
| Вторник     | Начало занятия1 – Окончание занятия1 – Группа<br>Начало занятия2 – Окончание занятия2 – Группа  |
| ...         |   |

Процедура должна проверять, что время занятий у разных групп 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 групп в день, или время его занятий с одной группой совпадает со временем занятий в другой группе, 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".

#### Л.р. №3. Создание триггеров.

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

#### Л.р. №4. Работа со средствами динамического SQL.

Создать процедуру, которая принимает в качестве параметра имя таблицы и имя поля в этой таблице. Выводит на экран информацию о количестве записей в таблице, количестве разных значений указанного поля и количестве неопределенных значений этого поля.

## 8. Методические рекомендации студентам по изучению дисциплины

Планирование и организация времени, отведенного на изучение дисциплины.

В начале семестра студент знакомится с содержанием и структурой дисциплины. Студент самостоятельно планирует свое время, опираясь на календарный график, приведенный в рабочей программе дисциплины. Все виды работ можно разделить на две группы – контактная работа и самостоятельная работа. Контактная работа может быть аудиторной, внеаудиторной, а также проводиться в электронной информационно–образовательной среде.

Контактная работа при проведении учебных занятий по дисциплине (модулю) включает в себя лекции, лабораторные работы, консультации по выполнению лабораторных работ. Консультации могут быть групповыми или индивидуальными. Контактная работа студентов по дисциплине также может содержать элементы самостоятельной работы. В этом случае она выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию. Объем времени на контактную работу студентов регламентируется расписанием занятий.

Самостоятельная работа студентов – планируемая учебная, научно-исследовательская работа студентов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия и не регламентируется расписанием занятий.

Самостоятельная работа, которую студент организует по своему усмотрению, без непосредственного контроля со стороны преподавателя - подготовка к лекциям, практическим занятиям, контрольным работам, экзамену. В этой связи стоит подчеркнуть, что очень важно умение оптимизировать процесс сочетания этих двух частей, необходимо равномерно распределять силы по всей дистанции семестра.

Для успешного освоения материала и качественного выполнения лабораторной работы необходимо после лекции и перед лабораторной работой, повторить материал (15 – 30 минут).

Перед контрольной работой необходимо не только повторить материал по конспекту лекций, но и изучить рекомендуемую литературу по соответствующим темам.

Сценарий изучения дисциплины (последовательность действий):

1. Посещение лекций (регламентируется расписанием занятий).
  2. Выполнение лабораторных работ и выполнение индивидуальных заданий (регламентируется расписанием занятий).
  - 3 Самостоятельная внеаудиторная работа с конспектом лекций и рекомендуемой литературой, решение задач.
  4. Выполнение контрольных работ.
  5. Подготовка к зачету.
- Рекомендации по работе с литературой.

Работа с литературой является основным методом самостоятельного овладения знаниями. Это сложный процесс, требующий выработки определенных навыков, поэтому студенту нужно обязательно научиться работать с книгой.

Осмысление литературы требует системного подхода к освоению материала. В работе с литературой системный подход предусматривает не только тщательное (при необходимости – многократное) чтение текста и изучение специальной литературы, но и обращение к дополнительным источникам – справочникам, энциклопедиям, словарям. Эти источники – важное подспорье в самостоятельной работе студента, поскольку глубокое изучение именно их материалов позволит студенту уверенно «распознавать», а затем самостоятельно оперировать теоретическими категориями и понятиями, следовательно – освоить новейшую научную терминологию. Такого рода работа с литературой обеспечивает решение студентом поставленной перед ним задачи (подготовка к практическому занятию, выполнение контрольной работы и т.д.).

Выбор литературы для изучения делается обычно по предварительному списку литературы, который выдал преподаватель, либо путем самостоятельного отбора материалов. После этого непосредственно начинается изучение материала, изложенного в книге.

Прежде чем приступить к чтению, необходимо запомнить или записать выходные данные издания: автор, название, издательство, год издания, название интересующих глав. Предисловие или введение книги поможет установить, на кого рассчитана данная публикация, какие задачи ставил перед собой автор. Это помогает составить представление о степени достоверности или научности данной книги. Содержание (оглавление) дает представление о системе изложения ключевых положений всей публикации и помогает найти нужные сведения. Если в книге есть главы или отдельные параграфы, которые соответствуют исследуемой теме дисциплины, то после этого необходимо ознакомиться с введением.

Во введении или предисловии разъясняются цели издания, его значение, содержится краткая информация о содержании глав работы. Иногда полезно после этого посмотреть послесловие или заключение. Особенно это важно, если это не учебник, а монография, потому что в заключении объясняется то, что может оказаться непонятным при изучении материала. В целом, это поможет правильно структурировать полученные знания.

После просмотра книги целиком или отдельной главы, которая была необходима для изучения определенной темы курса, нужно сделать записи в виде краткого резюме источника. В таком резюме следует отразить основную мысль изученного материала, приведенные в ее подтверждение автором аргументы, ценность данных аргументов и т.п. Данные аргументы помогут сформировать собственную оценку изучаемого вопроса.

Во время изучения литературы необходимо конспектировать и составлять рабочие записи прочитанного. Такие записи удлиняют процесс проработки, изучения книги, но способствуют ее лучшему осмыслению и усвоению, выработке навыков кратко и точно излагать материал. В идеале каждая подобная запись должна быть сделана в виде самостоятельных ответов на вопросы, которые задаются в конце

параграфов и глав изучаемой книги. Однако такие записи могут быть сделаны и в виде простого и развернутого плана, цитирования, тезисов, резюме, аннотации, конспекта.

При изучении литературы особое внимание следует обращать на новые термины и понятия. Понимание сущности и значения терминов способствует формированию способности логического мышления, приучает мыслить абстракциями, что важно при усвоении дисциплины. Поэтому при изучении темы курса студенту следует активно использовать универсальные и специализированные энциклопедии, словари, иную справочную литературу.

Вся рекомендуемая для изучения курса литература подразделяется на основную и дополнительную. К основной литературе относятся источники, необходимые для полного и твердого усвоения учебного материала (учебники и учебные пособия). Необходимость изучения дополнительной литературы диктуется прежде всего тем, что в учебной литературе (учебниках) зачастую остаются неосвещенными современные проблемы, а также не находят отражение новые документы, события, явления, научные открытия последних лет. Поэтому дополнительная литература рекомендуется для более углубленного изучения программного материала.

### **Список литературы**

11. Маркин А.В. Построение запросов и программирование на SQL : Учеб. пособие: [текст]//М.: Диалог-МИФИ, 2008.-318с. (30экз.)

## **6. Перечень учебной литературы**

### **6.1. Основная литература**

2. Девянин, П. Н. Модели безопасности компьютерных систем. Управление доступом и информационными потоками : учебное пособие / П. Н. Девянин. — 2-е изд., испр. и доп. — Москва : Горячая линия-Телеком, 2017. — 338 с. — ISBN 978-5-9912-0328-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/111049> (дата обращения: 13.01.2021). — Режим доступа: для авториз. пользователей.

3. Дудник, Е.А. Программирование. Часть III (лабораторный практикум по программированию в среде Delphi) [текст]: учебно-метод. пособие для студентов направления ИВТ/ Е.А. Дудник, Н.С. Зорина. - Рубцовск: РИИ, 2017. - 63 с. (6 экз.+ЭР)

### **6.2. Дополнительная литература**

4. Прохорова, О.В. Информационная безопасность и защита информации : учебник / О.В. Прохорова ; Самарский государственный архитектурно-строительный университет. – Самара : Самарский государственный архитектурно-строительный университет, 2014. – 113 с. : табл., схем., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=438331> (дата обращения: 25.01.2021). – Библиогр. в кн. – ISBN 978-5-9585-0603-3. – Текст : электронный.

5. Алгоритмы категорирования персональных данных для систем автоматизированного проектирования баз данных информационных систем : монография / А. В. Благодаров, В. С. Зияутдинов, П. А. Корнев, В. Н. Малыш. — Москва : Горячая линия-Телеком, 2015. — 116 с. — ISBN 978-5-9912-0307-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/111019> (дата обращения: 13.01.2021). — Режим доступа: для авториз. пользователей.

**Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

6. Goodfellow I., Bengio Y., Courville A. Deep Learning / MIT Press, 2016. – 775 с. ISBN-13: 978-0262035613; То же [Электронный ресурс]. - URL: <https://www.deeplearningbook.org/>

7. Библиотека машинного обучения <https://www.pytorch.org/>

8. Google CoLab: платформа для облачных вычислений <https://colab.research.google.com/>

9. Язык программирования Python 3: <https://learnpythonthehardway.org/python3/>

10. Open Machine Learning Course <https://mlcourse.ai/>

11. Anaconda: Сборка пакетов для научных и технических расчетов на Python <https://www.anaconda.com/>

12. <https://developers.google.com/machine-learning/crash-course>  
Курс лекций и лабораторных работ по машинному обучению