

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

**Т.В.Зудилова, С.Е. Иванов, С. Э. Хоружников**

**SQL и PL/SQL для разработчиков СУБД  
ORACLE**

**практикум**



**Санкт-Петербург**

**2012**

УДК 004.655, 004.657, 004.62

Т.В.Зудилова, С.Е. Иванов, С. Э. Хоружников

SQL и PL/SQL для разработчиков СУБД ORACLE - СПб: НИУ ИТМО,  
2012. – 74 с.

В пособии излагаются методические рекомендации к выполнению лабораторных работ по дисциплине “Создание клиент-серверных приложений”.

Предназначено для студентов всех инженерных специальностей, изучающих курс “Создание клиент-серверных приложений”

Рекомендовано к печати Ученым советом факультета инфокоммуникационных технологий. Протокол №4 от 13 декабря 2011 г.



В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

© Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2012

© Т.В.Зудилова, С.Е. Иванов, С. Э. Хоружников, 2012.

## Оглавление

Введение .....	4
Лабораторная работа 1. Выборка данных с помощью оператора SQL SELECT .....	13
Лабораторная работа 2. Ограничение и сортировка данных .....	17
Лабораторная работа 3. Применение однострочных функций.....	23
Лабораторная работа 4. Создания отчетов с помощью групповых функции. ....	31
Лабораторная работа 5. Выборка данных из нескольких таблиц.....	34
Лабораторная работа 6. Применение подзапросов.....	39
Лабораторная работа 7. Применение операторов соединения.....	42
Лабораторная работа 8. Управление данными.....	45
Лабораторная работа 9. Создание и управление таблицами с помощью операторов DDL.....	47
Лабораторная работа 10. Создание объектов схемы.....	49
Лабораторная работа 11. Управление объектами с помощью словаря базы данных.....	52
Лабораторная работа 12. Управление доступом пользователей.....	55
Лабораторная работа 13. Введение в PL /SQL.....	57
Лабораторная работа 14. Объявление переменных PL/SQL .....	57
Лабораторная работа 15. Исполняемые операторы.....	59
Лабораторная работа 16. Взаимодействие с Oracle Server .....	62
Лабораторная работа 17. Управляющие структуры.....	64
Лабораторная работа 18. Создание хранимых процедур и функций.....	65
Литература .....	66
Приложение 1. Таблицы схемы STAFF (персонал) .....	67

## Введение

В результате курса, проводимого под руководством преподавателя, слушатели познакомятся с:

- технологиями и концепцией реляционных баз данных;
- языком запросов SQL для обработки данных, метаданных, создания объектов базы данных;
- возможностями языка SQL для создания сложных подзапросов;
- языком PL/SQL;
- созданием блоков PL/SQL кода;
- использованием инструментов SQL\*Plus и SQL Developer.

## Цель курса

По окончании данного курса слушатели смогут:

- понимать основные свойства базы данных Oracle 11g;
- разбираться в компонентах, теоретических аспектах, терминологии;
- работать с SQL-командой SELECT;
- осуществлять ограничение строк и сортировку данных;
- использовать однострочные функции;
- группировать данные, используя групповые функции;
- делать выборки данных из нескольких таблиц;
- формулировать подзапросы;
- использовать операторы вертикального соединения (SET- операторов);
- манипулировать данными;
- создавать таблицы;
- создавать и использовать другие объекты базы данных: представления, последовательности, индексы;
- работать с объектами при помощи словаря данных.
- управлять пользовательским доступом;
- управлять объектами схемы;
- формировать отчеты, используя различные группировки данных;
- работать с данными в разных временных зонах;
- создавать сложные подзапросы;
- описывать язык PL/SQL, понимать его возможности и преимущества использования;
- определять и использовать PL/SQL-переменные;
- писать исполнимые команды;
- взаимодействовать с Oracle Server;
- использовать управляющие структуры;
- создавать хранимые процедуры и функции.

SQL (Structure Querying Language)— язык структурированных запросов — стандартный язык управления реляционными базами данных. Прототип языка структурированных запросов был разработан фирмой IBM на основе предложений доктора Кодда в статье "Реляционная модель данных для больших банков данных общего пользования". В 1979 году, появился первый продукт SQL под названием ORACLE, который был выпущен компанией Relational Software, Incorporated (впоследствии Oracle Corporation). Сегодня компания Oracle является одним из лидеров в области реализации технологий реляционных баз данных. SQL был утвержден стандартным языком в области управления базами данных в 1986 году. В 1987 году стандарт ANSI SQL был принят в качестве международного стандарта Международной организацией стандартов (ISO). Этот стандарт был изменен в 1992 году и получил название SQL/92. Самый новый на сегодня стандарт называется SQL3 и был принят в 1999 году SQL/99.

Реляционная база данных — это база данных, разделенная на логически цельные сегменты, называемые таблицами, которые внутри базы данных связаны между собой. Реляционная база данных позволяет разделить данные на логичные более мелкие и более управляемые сегменты, что обеспечивает оптимальное представление данных и возможность организации нескольких уровней доступа к данным.

Основные виды команд, реализующих в SQL выполнение различных функций:

- DDL (Data Definition Language — язык определения данных);
- DML (Data Manipulation Language — язык манипуляций данными);
- DQL (Data Query Language — язык запросов к данным);
- DCL (Data Control Language — язык управления данными);
- команды администрирования данных;
- команды управления транзакциями.

Язык определения данных (DDL) предоставляет пользователю возможность создавать различные объекты базы данных и переопределять их структуру. Основные команды DDL: CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, ALTER INDEX, DROP INDEX

Язык манипуляций данными (DML) предоставляет пользователю возможность манипулировать данными внутри объектов реляционной базы данных. Основные команды DML: INSERT, UPDATE, DELETE

Выборка данных (DQL)

Для пользователя реляционной базы данных язык запросов к данным (DQL) является самой важной частью SQL. Команда SELECT, имеющая множество опций и необязательных параметров, используется для построения запросов к реляционным базам данных. С ее помощью можно конструировать запросы любой сложности.

Команды управления данными в SQL позволяют осуществлять контроль над возможностью доступа к данным внутри базы данных.

Команды DCL обычно используются для создания объектов, относящихся к управлению доступом пользователей к базе данных, а также для назначения пользователям уровней привилегий доступа. Команд управления данными:

ALTER PASSWORD, GRANT, REVOKE

Команды администрирования данных дают пользователю возможность выполнять аудит и анализ операций внутри базы данных. Эти команды могут также помочь при анализе производительности системы данных.

Команды управления транзакциями:

Команда COMMIT используется для сохранения транзакции.

Команда ROLLBACK используется для отмены транзакции.

Команда SAVEPOINT создает точки внутри транзакций, к которым отсылает команда ROLLBACK.

PL/SQL - это процедурно-ориентированный язык, созданный для облегчения обработки команд SQL в СУБД Oracle. Его можно использовать для написания процедур хранения, функций, исполняемых в этой СУБД. Кроме того, этот язык можно применять и в клиентских приложениях и приложениях среднего уровня.

Oracle SQL Developer - графический инструментарий, который повышает производительность разработки приложений и упрощает задачи при разработке баз данных. Созданное специально для разработчиков баз в среде Oracle, данное решение упрощает цикл разработки и сокращает потребность в покупке стороннего ПО для разработки и отладки кода в SQL и PL/SQL.

СУБД Oracle Database 11g - это комплексное решение для управления данными, которое поддерживает все популярные среды разработки приложений, имеющиеся на сегодняшний день. При использовании Oracle SQL Developer повышается производительность разработки приложения Oracle Database 11g и сокращается цикл разработки.

СУБД Oracle Database 11g - это единое интегрированное решение, для управления информацией, доступное на различных открытых платформах и поддерживаемое множеством сторонних приложений. В базе данных хранятся не только реляционные данные, но и документы, XML, мультимедиа информация о размещении данных и ГИС информация. Производительность системы оптимизирована для перечисленных видов данных, например файлы, хранящиеся в базе с использованием функции "Безопасность файлов" (Secure Files), могут быть отправлены на хранение и извлечены обратно быстрее, чем в других системах. Для всех видов данных в базе осуществляется индексирование и предоставляется SQL-доступ, позволяя легко внедрять новые виды данных в существующие

приложения. Также предоставляется обычный порядок доступа по протоколам, характерным для конкретных видов данных.

СУБД Oracle Database 11g - база данных, разработанная специально для работы в сетях распределенных вычислений Grid, предназначенная для эффективного развертывания на базе различных типов оборудования, от небольших серверов до мощных симметричных многопроцессорных серверных систем, от отдельных кластеров до корпоративных распределенных вычислительных систем. СУБД предоставляет возможность автоматической настройки и управления, что делает ее использование простым и экономически выгодным.

СУБД Oracle Database 11g поставляется в четырех различных редакциях, ориентированных на различные сценарии разработки и развертывания приложений. Кроме того, корпорация Oracle предлагает несколько дополнительных программных продуктов, расширяющих возможности Oracle Database 11g для работы с конкретными прикладными пакетами. Ниже перечислены существующие редакции СУБД Oracle Database 11g:

СУБД Oracle Database 11g Standard Edition One характеризуется простотой эксплуатации, мощностью и выгодным соотношением цены и производительности для приложений масштаба рабочих групп, отдельных подразделений или приложений, работающих в среде интернет/интранет. Работая в различных средах, начиная от односерверных конфигураций для малого бизнеса и заканчивая распределенными средами крупных филиалов, Oracle Database 11g Standard Edition One обладает всеми функциональными возможностями для обеспечения работы критических для бизнеса приложений. Редакция Standard Edition One лицензируется только для серверов, имеющих не более двух процессоров.

СУБД Oracle Database Standard Edition 11g (SE) обеспечивает простоту эксплуатации, мощь и производительность, что и редакция Standard Edition One, поддерживая работу более мощных вычислительных систем с использованием технологии кластеризации сервисов Real Application Clusters. Эта редакция лицензируется для использования на одном сервере с числом процессоров, не превышающим четырех, или на серверном кластере, поддерживающем не более четырех процессоров.

СУБД Oracle Database 11g Enterprise Edition (EE) обеспечивает эффективное, надежное и безопасное управление данными таких критически важных для бизнеса приложений, как онлайн-среды, выполняющие масштабную обработку транзакций (OLTP), хранилища данных с высокой интенсивностью потока запросов, а также ресурсоемкие интернет-приложения. Редакция Oracle Database Enterprise Edition предоставляет инструментальные средства и функции, обеспечивающие соответствие требованиям современных корпоративных приложений в области доступности и масштабируемости. Эта редакция содержит все компоненты Oracle Database, а также допускает расширение посредством

приобретения дополнительных модулей и приложений, описанных далее в этой статье.

СУБД Oracle Database 11g Personal Edition поддерживает однопользовательскую разработку и развертывание приложений, полностью совместимых с редакциями Oracle Database Standard Edition One, Oracle Database Standard Edition и Oracle Database Enterprise Edition. Предоставив отдельным пользователям мощную функциональность пакета Oracle Database 11g, корпорация Oracle создала базу данных, сочетающую мощь популярнейшей в мире СУБД и простоту эксплуатации, которую вы вправе ожидать от приложения для настольного ПК.

Используя СУБД Oracle Database 11g, организации смогут управлять всей корпоративной информацией, а также оперативно и уверенно адаптироваться ко все более изменчивой конкурентной среде. Чтобы предоставить такие возможности, в новом выпуске расширены уникальные механизмы Oracle, обеспечивающие кластеризацию баз данных, автоматизацию центров обработки данных и управление рабочими нагрузками. Используя защищенные, масштабируемые grid-инфраструктуры на базе недорогих серверов и систем хранения, обеспечивающие высокие уровни готовности, клиенты Oracle могут создавать OLTP приложения, хранилища данных и системы управления контентом с самыми высокими требованиями.

СУБД Oracle Database 11g предлагает важные новые возможности в области секционирования и сжатия данных, обеспечивая экономически эффективное управление жизненным циклом информации и ее хранением. Oracle Database 11g позволяет автоматизировать множество ручных операций секционирования данных, дополняя уже существующие методы секционирования (по диапазону, по списку и хэш-секционирование) методами секционирования по интервалу, ссылке и виртуальному столбцу. Oracle Database 11g, обеспечивает управление хранением в соответствии с бизнес-правилами. Oracle Database 11g обеспечивает эффективное сжатие как структурированных, так и неструктурированных (LOB) данных в OLTP приложениях, хранилищах данных и системах управления контентом. Реализованные в Oracle Database 11g новые передовые технологии обеспечивают сжатие в 2-3 раза и более для любых данных.

В СУБД Oracle Database 11g появились новые функции обеспечения высокой надежности, такие как: Oracle Flashback Transaction упрощает откат транзакции, сделанной по ошибке, а также любых связанных транзакций, Parallel Backup and Restore повышает скорость резервного копирования и восстановления баз данных и Hot Patching повышает уровень доступности систем благодаря применению пакетов обновлений без остановки СУБД. Новый компонент Data Recovery Advisor позволяет администраторам значительно ускорить процедуры восстановления, автоматизируя процесс исследования проблем, предлагая план восстановления и учитывая сбои, предшествующие данному сбою.



Oracle Fast Files - это технология нового поколения для хранения в базе данных больших объектов (LOB), таких как изображения и большие текстовые объекты, а также XML-данные, медицинские снимки и трехмерные объекты. Oracle Fast Files позволяет приложениями для баз данных использовать весь спектр функциональных возможностей файловых систем. Возможность хранить корпоративную информацию различных типов, а также просто и быстро извлекать ее, помогает предприятиям лучше понимать свой бизнес и оперативно адаптироваться к изменениям.

СУБД Oracle Database 11g предлагает значительно усовершенствованную технологию XML DB, позволяющую клиентам хранить XML-данные в своем собственном формате и умело управлять ими. Реализованная в этой версии поддержка бинарного XML позволяет клиентам выбирать варианты хранения XML в соответствии со специфическими потребностями. Кроме того, XML DB обеспечивает управление XML-данными с использованием стандартных для отрасли интерфейсов, благодаря поддержке стандартов XQuery, Java Specification Requests и SQL/XML.

В СУБД Oracle Database 11g реализованы еще более совершенные, чем прежде, технологии обеспечения безопасности. В новой версии средства Oracle Transparent Data Encryption выходят за пределы шифрования на уровне столбцов. Oracle Database 11g предлагает возможности шифрования табличного пространства, которые могут применяться для шифрования всех таблиц, индексов или других объектов, хранимых в данном табличном пространстве. Шифрование обеспечивается и для хранящихся в базе данных объектов LOB.

В СУБД Oracle Database 11g разработаны функции, которые помогают организациям поддерживать высокопроизводительную, масштабируемую инфраструктуру для предоставления пользователям высочайшего качества обслуживания. Oracle Database 11g предлагает такие новые функции, как Query Result Caches для повышения производительности и масштабируемости приложений благодаря кэшированию и повторному использованию результатов часто выполняемых запросов к базе данных и функций на уровнях базы данных и приложений и Database Resident Connection Pooling повышает масштабируемость web-систем, обеспечивая создание пулов соединений для приложений, не являющихся многопоточными.

СУБД Oracle Database 11g предлагает различные инструменты разработки, а также простой процесс создания приложений, максимально эффективно реализующий ключевые функциональные возможности Oracle Database 11g. Среди новых функций - Client Side Caching (кэширование на стороне клиента), Binary XML для ускорения работы приложений, обрабатывающих, хранящих и извлекающих XML файлы. Кроме того, Oracle Database 11g предлагает новый Just in time Компилятор Java, обеспечивая ускоренное исполнение Java-процедур базы данных без

необходимости использовать компилятор стороннего поставщика; встроенную интеграцию с Visual Studio для разработки приложений .NET в среде Oracle; поддержку экспорта данных из Microsoft Access в Oracle Application Express; и функцию простого построения запросов в SQL Developer для быстрой разработки SQL- и PL/SQL-процедур.

В СУБД Oracle Database 11g функции обеспечения управляемости помогают организациями легко управлять корпоративными grid-инфраструктурами и отвечать ожиданиям пользователей, связанным с уровнями обслуживания. Oracle Database 11g предлагает расширенные функции самоуправления и автоматизации, которые помогут клиентам сокращать затраты на управление системами, повышая производительность, масштабируемость, доступность и безопасности приложений для баз данных. Новые средства обеспечения управляемости в Oracle Database 11g включают автоматическую настройку SQL и области памяти, новый компонент Partitioning Advisor, предлагающий администраторам рекомендации по секционированию таблиц и индексов для повышения производительности, а также улучшенные средства диагностики производительности для кластерных баз данных. Кроме того, Oracle Database 11g включает новый компонент Support Workbench, который предлагает простой в использовании интерфейс, представляющий администраторам сведения о проблемах, возникающих при работе СУБД, вместе с информацией о том, как можно быстро решить эти проблемы.

В СУБД Oracle Database 11g предусмотрена функция автоматизированного управления дисками (Automated Storage Management, ASM), которая автоматически производит разделение баз данных и их зеркалирование без необходимости покупать стороннее ПО для управления томами. С ростом объема данных можно добавлять дополнительные диски, и ASM автоматически снова разделит и перераспределит данные по доступным дискам, чтобы достичь оптимальной производительности. Сходным образом диски, на которых обнаружались ошибки, можно извлечь, и ASM снова перераспределит данные. В Oracle Database 11g Release 2 функция автоматизированного управления дисками значительно усовершенствована. Новые интеллектуальные средства помещают редко используемые данные на внутренних секторах физических дисков, а часто используемые - на внешних секторах, тем самым оптимизируя производительность.

Кластерная файловая система ASM (ASM Cluster File System, ACFS) означает, что теперь возможно применять ASM для баз данных Oracle, а также для файловых систем общего назначения, т. е. предлагается единое хранилище как для файлов баз данных Oracle и бинарных файлов приложений Oracle, так и для файлов приложений сторонних производителей. Также поддерживаются снимки дисков в режиме "только для чтения" - до 64 копий данных о файловой системе одновременно. С увеличением объема баз данных усложняется управление ими. Методы, хорошо зарекомендовавшие себя в работе с несколькими гигабайтами

информации, редко дают столь же хороший результат с терабайтами. Вот уже десять лет Oracle расширяет возможности создания разделов на дисках для решения этих вопросов.

Oracle лидирует на рынке хранилищ данных и продолжает оптимизировать механизм своей СУБД. Оптимизация включает использование передовых технологий, таких как параллельная обработка, индексация по растровому отображению, именованные выводимые таблицы с хранимым результатом, общее управление, встроенные алгоритмы ETL, OLAP и Data Mining. В СУБД Oracle Database 11g Release 2 добавлены новые функции, в том числе автоматическое определение оптимальной степени распараллеливания запроса исходя из имеющихся ресурсов. Предусмотрено также автоматическое формирование параллельных очередей запросов, когда система на основании имеющихся ресурсов определяет, будет ли более эффективно поставить запрос в очередь, пока требуемые ресурсы не освободятся.

СУБД Oracle Database 11g Release 2 предоставляет преимущества от наращивания памяти в вычислительной сети, состоящей из недорогих серверов, а также возможностей передового сжатия. Oracle Database 11g Release 2 автоматически распределяет сжатую таблицу с большим количеством данных (или несжатую таблицу меньших размеров) в доступную память во всех серверах вычислительной сети, а затем сводит обработку параллельных запросов к данным в памяти отдельных узлов. Это существенно ускоряет обработку запросов, особенно когда большие таблицы полностью сжаты в доступной памяти с помощью функции сжатия. Для защиты от непредвиденных отключений необходимо, чтобы в архитектуру были встроены компоненты резервирования. Для создания копий данных нужны дополнительные диски, для восстановления сервера после отказа требуется дополнительное аппаратное обеспечение, а для повышения отказоустойчивости необходимы дополнительные центры обработки данных. Резервное оборудование достаточно дорого обходится и оправдывает себя обычно только в случае реального отказа компонента: это напоминает дорогостоящую страховку. Кроме того, для интеграции таких компонентов, как правило, от разных поставщиков, требуется разнообразное ПО. В результате система усложняется и растет риск ошибок персонала.

Кластеры Oracle Real Application Clusters защищают данные в случае отказа сервера, и обеспечивают дополнительную масштабируемость приложений. В случае аварийного восстановления используется функция защиты данных Data Guard. Она обеспечивает эффективную синхронизацию между рабочими базами данных и базами, помещенными в хранилище. Любой компонент аппаратного обеспечения вычислительной сети Oracle можно удалить или добавить в любое время. Можно добавлять и удалять диски, использующие ASM, и данные будут автоматически перераспределены в рамках новой инфраструктуры. Можно легко добавлять и удалять серверы в кластере RAC, а пользователи,

подключенные к этим узлам, будут автоматически перераспределены по системе. Эта функция - перераспределение пользователей с одного сервера на другой в пределах кластера RAC - также позволяет выполнять обновление ПО баз данных. Если необходимо внести изменения в ПО, можно удалить сервер из кластера, изменить ПО и затем вернуть сервер обратно в кластер. Аналогичным образом пользователей можно перевести с рабочих баз данных на резервные с помощью Oracle Data Guard. Поддерживаются различия в версиях баз данных и операционных систем между рабочими и резервными станциями. Это означает, что резервную базу данных можно обновить до последней версии, протестировать новую систему, а затем перевести пользователей прежней версии на новую, без каких-либо простоев.

СУБД Oracle Database 11g Release 2 обеспечивает для всех данных снижение затрат, безопасность и высокую надежность, упрощается объединение, управление и согласованность различных данных, достигается высокая доступность информации и простота управления средой.

## Лабораторная работа 1. Выборка данных с помощью оператора SQL SELECT

В этой лабораторной работе Вы научитесь работать в среде программирования SQL Developer, соединяться с базой данных и создавать выборки данных с помощью языка запросов (DQL) и рассмотрите синтаксис оператора select.

### Создание соединения с базой данных в SQL Developer

1. Запустите SQL Developer.
2. Чтобы создать новое соединение с базой данных, в Навигаторе Соединений, нажмите кнопку Connections. Выберите Новое Соединение из меню.
3. Создайте соединение с базой данных, используя следующую информацию:
  - a. Имя подключения: ORCL.
  - b. Имя пользователя: oracle
  - c. Пароль: oracle
  - d. Имя узла: Введите имя хоста машины, где работает сервер базы данных.
  - e. Порт: 1521
  - f. SID: ORCL
  - g. Установите флажок Save Password.

### Тестирование соединения с базой данных Oracle

5. Протестируйте новое соединение.
6. Если состояние - Успех, подключитесь к базе данных.

### Просмотр Таблиц в Навигаторе Соединений

7. В Навигаторе Соединений, просмотрите объекты, доступные Вам в Табличном узле. Проверьте, какие таблицы присутствуют:
8. Просмотрите структуру таблицы WORKERS.
9. Просмотрите данные таблицы SECTIONS.

### Запустите SQL Developer

1. Ваша задача состоит в том, чтобы определить структуру таблицы SECTIONS и ее содержание

Ниже приведен результат выполнения

Name	Null?	Type
SECTION_ID	NOT NULL	NUMBER(4)
SECTION_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER(6)
AREA_ID		NUMBER(4)

SECTION_ID	SECTION_NAME	MANAGER_ID	AREA_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700
100	Finance	108	1700
110	Accounting	205	1700

Определите структуру таблицы WORKERS

Ниже приведен результат выполнения

Name	Null?	Type
WORKER_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
WORK_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
SECTION_ID		NUMBER(4)

Выведите на экран фамилию, код работы, дату приема, и идентификатор сотрудника для каждого сотрудника. Укажите псевдоним STARTDATE для столбца HIRE\_DATE. Сохраните свой SQL-оператор в script\_01\_07.sql. Протестируйте свой запрос в script\_01\_07.sql.

Ниже приведен результат выполнения

WORKER_ID	LAST_NAME	WORK_ID	STARTDATE
198	OConnell	SH_CLERK	21-JUN-99
199	Grant	SH_CLERK	13-JAN-00
200	Whalen	AD_ASST	17-SEP-87
201	Hartstein	MK_MAN	17-FEB-96
202	Fay	MK_REP	17-AUG-97
203	Mavris	STAFF_REP	07-JUN-94
204	Baer	PR_REP	07-JUN-94

205 Higgins	AC_MGR	07-JUN-94
206 Gietz	AC_ACCOUNT	07-JUN-94

9. Выведите на экран все уникальные коды работ из таблицы WORKERS

Ниже приведен результат выполнения.

```

WORK_ID
-----
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
FI_ACCOUNT
STAFF_REP
IT_PROG
MK_MAN
PR_REP

```

10. Определите заголовки столбцов для своего отчета. Скопируйте script\_01\_07.sql в SQL\*Plus . Назовите заголовки столбцов Emp #, WORKER, WORK, Hire Date соответственно.

Ниже приведен результат выполнения.

Emp #	WORKER	WORK	Hire Date
198	OConnell	SH_CLERK	21-JUN-99
199	Grant	SH_CLERK	13-JAN-00
200	Whalen	AD_ASST	17-SEP-87
201	Hartstein	MK_MAN	17-FEB-96
202	Fay	MK_REP	17-AUG-97
203	Mavris	STAFF_REP	07-JUN-94
204	Baer	PR_REP	07-JUN-94
205	Higgins	AC_MGR	07-JUN-94
206	Gietz	AC_ACCOUNT	07-JUN-94

11. Создайте отчет обо всех сотрудниках и их ID работы. Выведите на экран фамилию, и ID работы (разделенный запятой), и назовите столбец WORKER and Title.

Ниже приведен результат выполнения.

```

WORKER and Title
-----
OConnell, SH_CLERK

```

Grant, SH\_CLERK  
Whalen, AD\_ASST  
Hartstein, MK\_MAN  
Fay, MK\_REP  
Mavris, STAFF\_REP  
Baer, PR\_REP  
Higgins, AC\_MGR  
Gietz, AC\_ACCOUNT  
King, AD\_PRES  
Kochhar, AD\_VP

12. Создайте запрос, чтобы вывести на экран все данные из таблицы WORKERS. Разделите каждый столбец, запятой. Назовите заголовок столбца THE\_OUTPUT.

Ниже приведен результат выполнения.

THE\_OUTPUT

```
-----  
198,Donald,OConnell,DOCONNEL,650.507.9833,SH_CLERK,124,21-JUN-  
99,2600,,50  
199,Douglas,Grant,DGRANT,650.507.9844,SH_CLERK,124,13-JAN-  
00,2600,,50  
200,Jennifer,Whalen,JWHALEN,515.123.4444,AD_ASST,101,17-SEP-  
87,4400,,10  
201,Michael,Hartstein,MHARTSTE,515.123.5555,MK_MAN,100,17-FEB-  
96,13000,,20  
202,Pat,Fay,PFAY,603.123.6666,MK_REP,201,17-AUG-97,6000,,20  
203,Susan,Mavris,SMAVRIS,515.123.7777,STAFF_REP,101,07-JUN-  
94,6500,,40  
204,Hermann,Baer,HBAER,515.123.8888,PR_REP,101,07-JUN-94,10000,,70  
205,Shelley,Higgins,SHIGGINS,515.123.8080,AC_MGR,101,07-JUN-  
94,12000,,110  
206,William,Gietz,WGIETZ,515.123.8181,AC_ACCOUNT,205,07-JUN-  
94,8300,,110  
100,Steven,King,SKING,515.123.4567,AD_PRES,,17-JUN-87,24000,,90  
101,Neena,Kochhar,NKOCHHAR,515.123.4568,AD_VP,100,21-SEP-  
89,17000,,90
```



## Лабораторная работа 2. Ограничение и сортировка данных

В этой лабораторной работе Вы научитесь работать с SQL-командой SELECT и осуществлять ограничение строк и сортировку данных по различным полям.

1. Создайте отчет, который выводит на экран фамилию и зарплату сотрудников, которые зарабатывают больше чем 12 000\$. Сохраните SQL в текстовый файл, script\_02\_01.sql. Выполните свой запрос.

Ниже приведен результат выполнения.

LAST_NAME	SALARY
Hartstein	13000
King	24000
Kochhar	17000
De Haan	17000
Russell	14000
Partners	13500

2. Создайте отчет, который выводит на экран фамилию и идентификатор отдела для сотрудника номер 176.

Ниже приведен результат выполнения.

LAST_NAME	SECTION_ID
Taylor	80

3. Найдите сотрудников с высокой зарплатой и низкой. Измените script\_02\_01.sql, чтобы вывести на экран фамилию и зарплату для любого сотрудника, зарплата которого не находится в диапазоне от 5 000\$ до 12 000\$. Сохраните SQL в текстовый файл, script\_02\_03.sql.

Ниже приведен результат выполнения.

LAST_NAME	SALARY
OConnell	2600
Grant	2600
Whalen	4400
Hartstein	13000
King	24000
Kochhar	17000

De Haan	17000
Austin	4800
Pataballa	4800
Lorentz	4200
Khoo	3100

4. Создайте отчет, чтобы вывести на экран фамилию, ID работы, и дату найма сотрудников Мэтоса и Тэйлора. Упорядочите запрос в порядке возрастания даты найма.

Ниже приведен результат выполнения.

LAST_NAME	WORK_ID	HIRE_DATE
Taylor	SH_CLERK	24-JAN-98
Matos	ST_CLERK	15-MAR-98
Taylor	SA_REP	24-MAR-98

5. Выведите на экран фамилию и идентификатор отдела всех сотрудников в отделах 20 или 50 в возрастающем алфавитном порядке по имени.

Ниже приведен результат выполнения.

LAST_NAME	SECTION_ID
Atkinson	50
Bell	50
Bissot	50
Bull	50
Cabrio	50
Chung	50
Davies	50
Dellinger	50
Dilly	50
Everett	50
Fay	20

6. Измените script\_02\_03.sql, чтобы вывести на экран фамилию и зарплату сотрудников, которые зарабатывают между 5 000\$ и 12 000\$ и находятся в отделе 20 или 50. Назовите столбцы WORKER и Monthly Salary, соответственно. Сохраните script\_02\_03.sql как script\_02\_06.sql. Выполните script\_02\_06.sql.

Ниже приведен результат выполнения.

WORKER                      Monthly Salary

```
-----  
Fay                              6000  
Weiss                            8000  
Fripp                            8200  
Kaufling                        7900  
Vollman                        6500  
Mourgos                        5800
```

7.     Создайте отчет, который выводит на экран фамилию и, дату найма всех сотрудников, которые были наняты в 1994.

Ниже приведен результат выполнения.

LAST\_NAME                      HIRE\_DATE

```
-----  
Mavris                        07-JUN-94  
Baer                            07-JUN-94  
Higgins                        07-JUN-94  
Gietz                          07-JUN-94  
Greenberg                      17-AUG-94  
Faviet                        16-AUG-94  
Raphaely                       07-DEC-94
```

8.     Создайте отчет, чтобы вывести на экран фамилию и название должности всех сотрудников, у которых нет менеджера.

Ниже приведен результат выполнения.

LAST\_NAME                      WORK\_ID

```
-----  
King                            AD_PRES
```

9.     Создайте отчет, чтобы вывести на экран фамилию, зарплату, и комиссию всех сотрудников, которые зарабатывают комиссии. Данные выведите в порядке убывания зарплаты и комиссий.

Ниже приведен результат выполнения.

LAST\_NAME                      SALARY COMMISSION\_PCT

```
-----  
Russell                        14000            .4  
Partners                       13500            .3
```

Errazuriz	12000	.3
Ozer	11500	.25
Cambrault	11000	.3
Abel	11000	.3
Vishney	10500	.25
Zlotkey	10500	.2
King	10000	.35
Tucker	10000	.3
Bloom	10000	.2

10. Создайте отчет, который выводит на экран фамилию и зарплату сотрудников, которые зарабатывают больше чем количество, которое пользователь вводит после подсказки. Сохраните этот запрос в script\_02\_10.sql. Если Вы вводите 6800, отчет выводит на экран следующие результаты:

Ниже приведен результат выполнения.

LAST_NAME	SALARY
-----	-----
Hartstein	13000
Baer	10000
Higgins	12000
Gietz	8300
Popp	6900
Raphaely	11000
Weiss	8000
Fripp	8200
Kaufling	7900
Russell	14000
Partners	13500
Errazuriz	12000
Cambrault	11000
Zlotkey	10500
Tucker	10000
Bernstein	9500
Hall	9000
Olsen	8000
Cambrault	7500
Tuvault	7000
King	10000
Sully	9500
McEwen	9000
Smith	8000
Doran	7500

Sewall	7000
Vishney	10500
Greene	9500
Marvins	7200
Ozer	11500
Bloom	10000
Fox	9600
Smith	7400
Bates	7300
Abel	11000
Hutton	8800
Taylor	8600
Livingston	8400
Grant	7000

48 rows selected.

11. Создайте запрос, который запрашивает пользователя ввести ID менеджера и выводит ID сотрудника, фамилию, зарплату, и отдел для сотрудников этого менеджера. Отсортируйте отчет относительно введенного столбца.

12. Выведите на экран все фамилии сотрудника, в которых третья буква имени - *a*.

Ниже приведен результат выполнения.

LAST\_NAME

-----

Grant  
Grant  
Whalen

13. Выведите на экран фамилии всех сотрудников, у которых есть и буква *a* и *e* в фамилии.

Ниже приведен результат выполнения.

LAST\_NAME

-----

Baer  
Bates  
Colmenares  
Davies  
De Haan  
Faviet  
Fleaur

Gates  
Hartstein  
Markle  
Nayer  
Partners  
Patel  
Philtanker  
Raphaely  
Sewall  
Whalen

14. Выведите на экран фамилию, работу, и зарплату для всех сотрудников, работа которых - или SA\_REP или ST\_CLERK и их зарплаты не равны 2 500\$, 3 500\$, или 7 000\$.

Ниже приведен результат выполнения.

LAST_NAME	WORK_ID	SALARY
-----	-----	-----
Nayer	ST_CLERK	3200
Mikkilineni	ST_CLERK	2700
Landry	ST_CLERK	2400
Markle	ST_CLERK	2200
Bissot	ST_CLERK	3300
Atkinson	ST_CLERK	2800
Olson	ST_CLERK	2100
Mallin	ST_CLERK	3300
Rogers	ST_CLERK	2900
Gee	ST_CLERK	2400
Philtanker	ST_CLERK	2200

15. Измените script\_02\_06.sql, чтобы вывести на экран фамилию, зарплату, и комиссию для всех сотрудников, количество комиссии которых составляет 20 %. Сохраните script\_02\_06.sql как script\_02\_15.sql. Запустите script\_02\_15.sql.

Ниже приведен результат выполнения.

WORKER	Monthly Salary	COMMISSION_PCT
-----	-----	-----
Zlotkey	10500	.2
Olsen	8000	.2
Cambrault	7500	.2
Bloom	10000	.2
Fox	9600	.2
Taylor	8600	.2
Livingston	8400	.2

### Лабораторная работа 3. Применение однострочных функций.

В этой лабораторной работе Вы научитесь использовать однострочные функции для применения их в логических и математических операциях выборки данных.

1. Напишите запрос, чтобы вывести на экран текущую дату. Назовите столбец Date.

Ниже приведен результат выполнения.

Date

-----

11-OCT-11

2. Выведите на экран идентификатор сотрудника, фамилию, зарплату, и зарплату, увеличенную на 15.5 % (выраженную в целом число) для каждого сотрудника. Назовите столбец New Salary. Сохраните SQL в текстовый файл, script\_03\_02.sql.
3. Выполните свой запрос в файле script\_03\_02.sql.

Ниже приведен результат выполнения.

WORKER_ID	LAST_NAME	SALARY	New Salary
198	OConnell	2600	3003
199	Grant	2600	3003
200	Whalen	4400	5082
201	Hartstein	13000	15015
202	Fay	6000	6930
203	Mavris	6500	7508
204	Baer	10000	11550
205	Higgins	12000	13860
206	Gietz	8300	9587
100	King	24000	27720
101	Kochhar	17000	19635
102	De Haan	17000	19635
103	Hunold	9000	10395
104	Ernst	6000	6930
105	Austin	4800	5544
106	Pataballa	4800	5544
107	Lorentz	4200	4851
108	Greenberg	12000	13860
109	Faviet	9000	10395

110 Chen	8200	9471
111 Sciarra	7700	8894
112 Urman	7800	9009

4. Измените свой запрос `script_03_02.sql`, чтобы добавить столбец, который вычитает старую зарплату из новой зарплаты. Назовите столбец `Increase`. Сохраните `script_03_04.sql`. Выполните запрос.

Ниже приведен результат выполнения.

WORKER_ID	LAST_NAME	SALARY	New Salary	Increase
198	OConnell	2600	3003	403
199	Grant	2600	3003	403
200	Whalen	4400	5082	682
201	Hartstein	13000	15015	2015
202	Fay	6000	6930	930
203	Mavris	6500	7508	1008
204	Baer	10000	11550	1550
205	Higgins	12000	13860	1860
206	Gietz	8300	9587	1287
100	King	24000	27720	3720
101	Kochhar	17000	19635	2635
102	De Haan	17000	19635	2635
103	Hunold	9000	10395	1395
104	Ernst	6000	6930	930
105	Austin	4800	5544	744
106	Pataballa	4800	5544	744
107	Lorentz	4200	4851	651
108	Greenberg	12000	13860	1860
109	Faviet	9000	10395	1395
110	Chen	8200	9471	1271
111	Sciarra	7700	8894	1194
112	Urman	7800	9009	1209

5. Запишите запрос, который выводит на экран фамилию (с верхним регистром первой буквы) и длину фамилии для всех сотрудников, имя которых начинается с букв *J*, *A*, или *M*. Отсортируйте результаты по фамилиям сотрудников.

Ниже приведен результат выполнения.

Name	Length
Abel	4
Ande	4



Atkinson	8
Austin	6
Johnson	7
Jones	5
Mallin	6
Markle	6
Marlow	6
Marvins	7
Matos	5
Mavris	6
Mccain	6
Mcewen	6
Mikkilineni	11
Mourgos	7

Перепишите запрос так, чтобы пользователь мог ввести букву, с которой начинается фамилия. Например, если пользователь вводит М.

Name                      Length

```
-----
```

Mallin	6
Markle	6
Marlow	6
Marvins	7
Matos	5
Mavris	6
Mccain	6
Mcewen	6
Mikkilineni	11
Mourgos	7

10 rows selected.

6. Найдите продолжительность работы для каждого сотрудника. Для каждого сотрудника, выведите на экран фамилию и вычислите число месяцев между текущей датой и датой найма сотрудника. Назовите столбец MONTHS\_WORKED. Упорядочите свои результаты по числу месяцев. Округлите число месяцев до самого близкого целого числа.

**Примечание:** Ваши результаты будут отличаться.

Ниже приведен результат выполнения.

LAST_NAME	MONTHS_WORKED
Banda	138
Kumar	138
Ande	139

Markle	139
Philtanker	140
Zlotkey	140
Lee	140
Geoni	140
Grant	141
Marvins	141
Johnson	141
Gee	142
Popp	142
Perkins	142
Tuvault	143
Mourgos	143
Cambrault	144
Colmenares	146
OConnell	148
Sullivan	148
Grant	149
Olson	150

7. Создайте отчет, который выводит для каждого сотрудника:  
 <фамилия сотрудника> зарабатывает <зарплата> ежемесячно, но хочет  
 <зарплату x 3>. Назовите столбец Dream Salaries.  
 Ниже приведен результат выполнения.

#### Dream Salaries

```
-----
De Haan earns $17,000.00 monthly but wants $51,000.00.
Hunold earns $9,000.00 monthly but wants $27,000.00.
Ernst earns $6,000.00 monthly but wants $18,000.00.
Austin earns $4,800.00 monthly but wants $14,400.00.
Pataballa earns $4,800.00 monthly but wants $14,400.00.
Lorentz earns $4,200.00 monthly but wants $12,600.00.
Greenberg earns $12,000.00 monthly but wants $36,000.00.
Faviet earns $9,000.00 monthly but wants $27,000.00.
Chen earns $8,200.00 monthly but wants $24,600.00.
Sciarra earns $7,700.00 monthly but wants $23,100.00.
Urman earns $7,800.00 monthly but wants $23,400.00.
```

8. Создайте запрос, чтобы вывести на экран фамилию и зарплату для  
 всех сотрудников. Отформатируйте зарплату, чтобы было 15 символов, с  
 лево дополните символом \$. Назовите столбец SALARY.  
 Ниже приведен результат выполнения.

LAST_NAME	SALARY
De Haan	\$\$\$\$\$\$\$\$\$17000
Hunold	\$\$\$\$\$\$\$\$\$9000
Ernst	\$\$\$\$\$\$\$\$\$6000
Austin	\$\$\$\$\$\$\$\$\$4800
Pataballa	\$\$\$\$\$\$\$\$\$4800
Lorentz	\$\$\$\$\$\$\$\$\$4200
Greenberg	\$\$\$\$\$\$\$\$\$12000
Faviet	\$\$\$\$\$\$\$\$\$9000
Chen	\$\$\$\$\$\$\$\$\$8200
Sciarra	\$\$\$\$\$\$\$\$\$7700
Urman	\$\$\$\$\$\$\$\$\$7800

9. Выведите на экран фамилию каждого сотрудника, дату найма, и дату анализа зарплаты, которая является первым понедельником после шести месяцев работы. Назовите столбец REVIEW. Отформатируйте дату в формате “понедельник, тридцать первого июля 2011.”

Ниже приведен результат выполнения.

LAST_NAME	HIRE_DATE	REVIEW
Urman	07-MAR-98	Monday, the Fourteenth of September, 1998
Popp	07-DEC-99	Monday, the Twelfth of June, 2000
Raphaely	07-DEC-94	Monday, the Twelfth of June, 1995

10. Выведите на экран фамилию, дату найма, и день недели, в которую начал работать сотрудник. Назовите столбец DAY. Упорядочьте результаты по дням недели, начиная с понедельника.

Ниже приведен результат выполнения.

LAST_NAME	HIRE_DATE	DAY
Bloom	23-MAR-98	MONDAY
Smith	10-MAR-97	MONDAY
Errazuriz	10-MAR-97	MONDAY
Bernstein	24-MAR-97	MONDAY

Olsen	30-MAR-98 MONDAY
Mikkilineni	28-SEP-98 MONDAY
Sully	04-MAR-96 MONDAY
Everett	03-MAR-97 MONDAY
Kaufling	01-MAY-95 MONDAY
Grant	24-MAY-99 MONDAY
Marvins	24-JAN-00 MONDAY
OConnell	21-JUN-99 MONDAY
Sullivan	21-JUN-99 MONDAY
Fleaur	23-FEB-98 MONDAY
Doran	15-DEC-97 MONDAY
Patel	06-APR-98 MONDAY
Rajs	17-OCT-95 TUESDAY
Mourgos	16-NOV-99 TUESDAY
Colmenares	10-AUG-99 TUESDAY
Popp	07-DEC-99 TUESDAY
Perkins	19-DEC-99 SUNDAY
Partners	05-JAN-97 SUNDAY
Matos	15-MAR-98 SUNDAY
Stiles	26-OCT-97 SUNDAY
Cabrio	07-FEB-99 SUNDAY
Philtanker	06-FEB-00 SUNDAY
Gee	12-DEC-99 SUNDAY

11. Создайте запрос, который выводит на экран фамилии сотрудников и количество комиссии. Если сотрудник не зарабатывает комиссию, покажите “No Commisson” Назовите столбец COMM

Ниже приведен результат выполнения.

LAST_NAME	COMM
OConnell	No Commission
Grant	No Commission
Whalen	No Commission
Hartstein	No Commission
Fay	No Commission
Ernst	No Commission
Khoo	No Commission
Baida	No Commission
Tobias	No Commission
Nayer	No Commission
Vargas	No Commission
Russell	.4
Partners	.3

Errazuriz	.3
Cambrault	.3
Zlotkey	.2
Tucker	.3
Bernstein	.25
Hall	.25

12. Создайте запрос, который выводит на экран первые восемь символов фамилий сотрудников и показывает количество их зарплаты звездочками. Каждая звездочка показывает тысячу долларов. Отсортируйте данные в порядке убывания зарплаты. Назовите столбец WORKERS\_AND\_THEIR\_SALARIES. Ниже приведен результат выполнения.

WORKERS\_AND\_THEIR\_SALARIES

```

-----
Kochhar *****
De Haan *****
Russell *****
Partners *****
Hartstei *****
Higgins *****
Greenber *****
Errazuri *****
Ozer *****
Cambraul *****
Abel *****
Raphaely *****
Vishney *****
Zlotkey *****
Bloom *****
King *****
Tucker *****
Baer *****
Fox *****

```

13. Используя функцию DECODE, напишите запрос, который выводит на экран класс всех сотрудников, основанный на значении столбца WORK\_ID, используя следующие данные:

<i>Задание</i>	<i>Класс</i>
AD_PRES	A
ST_MAN	B
IT_PROG	C
SA_REP	D
ST_CLERK	E

Ни один из вышеупомянутых 0

Ниже приведен результат выполнения.

```
WORK_ID  G
----- -
AC_ACCOUNT 0
AC_MGR    0
AD_ASST   0
AD_PRES   A
AD_VP     0
FI_ACCOUNT 0
FI_ACCOUNT 0
FI_MGR    0
STAFF_REP 0
IT_PROG   C
SA_MAN    0
SA_MAN    0
SA_REP    D
SA_REP    D
SH_CLERK  0
SH_CLERK  0
ST_CLERK  E
ST_CLERK  E
ST_MAN    B
ST_MAN    B
```

14. Перепишите преведущий оператор, используя синтаксис CASE

Ниже приведен результат выполнения.

```
WORK_ID  G
----- -
AC_ACCOUNT 0
AC_MGR    0
AD_ASST   0
AD_PRES   A
AD_VP     0
AD_VP     0
FI_ACCOUNT 0
FI_ACCOUNT 0
FI_MGR    0
STAFF_REP 0
IT_PROG   C
IT_PROG   C
IT_PROG   C
```

#### Лабораторная работа 4. Создания отчетов с помощью групповых функций.

В этой лабораторной работе Вы научитесь группировать данные, используя групповые функции в различных операциях выборки данных, изучите в каких выражениях возможно применение групповых функций.

1. Найдите самую высокую, самую низкую зарплату, сумму, и среднюю зарплату всех сотрудников. Назовите столбец Максимум, Минимум, Сумма, и Среднее число, соответственно. Округлите результаты к близкому целому числу. Сохраните SQL- в текстовый файл, script\_04\_04.sql.

Ниже приведен результат выполнения.

Maximum	Minimum	Sum	Average
24000	2100	691400	6462

2. Измените запрос в script\_04\_04.sql, чтобы вывести на экран минимум, максимум, сумму, и среднюю зарплату для каждого типа работы. Сохраните script\_04\_04.sql как script\_04\_05.sql. Выполните оператор в script\_04\_05.sql.

Ниже приведен результат выполнения.

WORK_ID	Maximum	Minimum	Sum	Average
AC_MGR	12000	12000	12000	12000
AC_ACCOUNT	8300	8300	8300	8300
IT_PROG	9000	4200	28800	5760
ST_MAN	8200	5800	36400	7280
AD_ASST	4400	4400	4400	4400
PU_MAN	11000	11000	11000	11000
SH_CLERK	4200	2500	64300	3215
AD_VP	17000	17000	34000	17000
FI_ACCOUNT	9000	6900	39600	7920
MK_MAN	3000	13000	13000	13000
PR_REP	10000	10000	10000	10000
FI_MGR	12000	12000	12000	12000
PU_CLERK	3100	2500	13900	2780
SA_MAN	4000	10500	61000	12200
MK_REP	6000	6000	6000	6000
AD_PRES	24000	24000	24000	24000
SA_REP	11500	6100	250500	8350
STAFF_REP	6500	6500	6500	6500
ST_CLERK	3600	2100	55700	2785

19 rows selected.

3. Напишите запрос, чтобы вывести на экран число людей с одинаковой работой

Ниже приведен результат выполнения.

```
ORK_ID    COUNT(*)
```

```
-----  
AC_ACCOUNT 1  
AC_MGR      1  
AD_ASST     1  
AD_PRES     1  
AD_VP       2  
FI_ACCOUNT  5  
FI_MGR      1  
STAFF_REP   1  
IT_PROG     5  
MK_MAN      1  
MK_REP      1  
PR_REP      1  
PU_CLERK    5  
PU_MAN      1  
SA_MAN      5  
SA_REP     30  
SH_CLERK    20  
ST_CLERK    20  
ST_MAN      5
```

Обобщите запрос так, чтобы пользователь вводил название должности.  
Сохраните script\_04\_06.sql.

Ниже приведен результат выполнения.

```
Enter value for WORK_title: SA_REP  
old 3: WHERE WORK_id = '&WORK_title'  
new 3: WHERE WORK_id = 'SA_REP'
```

```
WORK_ID    COUNT(*)
```

```
-----  
SA_REP     30
```

4. Определите число менеджеров, не перечисляя их.

Используйте столбец MANAGER\_ID, чтобы определить число менеджеров.

Ниже приведен результат выполнения.

Number of Managers

```
-----  
18
```



5. Найдите разницу между самыми высокими и самыми низкими зарплатами. Назовите столбец DIFFERENCE.

Ниже приведен результат выполнения.

DIFFERENCE

```
-----  
21900
```

6. Создайте отчет, чтобы вывести на экран идентификатор менеджера и зарплату самого низкооплачиваемого сотрудника для этого менеджера. Исключите любого, менеджер которого не известен. Исключите любые группы, где минимальная зарплата составляет 6 000\$ или меньше. Отсортируйте вывод в порядке убывания зарплаты.

Ниже приведен результат выполнения.

MANAGER\_ID MIN(SALARY)

```
-----  
102      9000  
205      8300  
146      7000  
145      7000  
108      6900  
147      6200  
149      6200  
148      6100
```

8 rows selected.

7. Создайте запрос, чтобы вывести на экран общее количество сотрудников и численность персонала, нанятая в 1995, 1996, 1997, и 1998. Создайте соответствующие заголовки столбцов.

Ниже приведен результат выполнения.

```
TOTAL      1995      1996      1997      1998  
-----  
107         4        10        28        23
```

## Лабораторная работа 5. Выборка данных из нескольких таблиц.

В этой лабораторной работе Вы научитесь выполнять выборку данных из нескольких таблиц, логически связанных посредством первичных и внешних ключей.

1. Напишите запрос, чтобы вывести адреса всех отделов. Используйте таблицы LANDS и AREAS. Покажите ID расположения, адрес, город, область, и страну. Используйте Natural Join

Ниже приведен результат выполнения.

REA\_ID STREET\_ADDRESS CITY STATE\_PROVINCE LAND\_NAME

-----  
2200 12-98 Victoria Street Sydney New South Wales Australia

2800 Rua Frei Caneca 1360 Sao Paulo Sao Paulo Brazil

2. Напишите запрос, чтобы вывести на экран фамилию, номер отдела, и имя отдела для всех сотрудников

Ниже приведен результат выполнения.

LAST\_NAME SECTION\_ID SECTION\_NAME

-----  
De Haan 90 Executive  
Hunold 60 IT  
Ernst 60 IT  
Austin 60 IT  
Pataballa 60 IT  
Lorentz 60 IT  
Greenberg 100 Finance  
Faviet 100 Finance  
Chen 100 Finance  
Sciarra 100 Finance  
Urman 100 Finance  
Popp 100 Finance  
Raphaely 30 Purchasing  
Khoo 30 Purchasing  
Baida 30 Purchasing  
Tobias 30 Purchasing  
Himuro 30 Purchasing  
Colmenares 30 Purchasing  
Weiss 50 Shipping  
Fripp 50 Shipping  
Kaufling 50 Shipping  
Vollman 50 Shipping  
Mourgos 50 Shipping

Nayer	50 Shipping
Mikkilineni	50 Shipping
Landry	50 Shipping
Markle	50 Shipping
Bissot	50 Shipping
Atkinson	50 Shipping

3. Создайте отчет о сотрудниках в Торонто. Выведите на экран фамилию, работу, номер отдела, и имя отдела для всех сотрудников, которые работают в Торонто.

Ниже приведен результат выполнения.

LAST_NAME	WORK_ID	SECTION_ID	SECTION_NAME
Hartstein	MK_MAN	20	Marketing
Fay	MK_REP	20	Marketing

4. Создайте отчет, чтобы вывести на экран фамилию сотрудников и идентификатор сотрудника вместе с фамилией их менеджера и идентификатором менеджера. Назовите столбцы WORKER, EMP#, Manager, и Mgr#, соответственно. Сохраните SQL в текстовый файл, script\_05\_04.sql.

Ниже приведен результат выполнения.

WORKER	EMP# Manager	Mgr#
Hunold	103 De Haan	102
Ernst	104 Hunold	103
Austin	105 Hunold	103
Pataballa	106 Hunold	103
Lorentz	107 Hunold	103
Greenberg	108 Kochhar	101
Faviet	109 Greenberg	108
Chen	110 Greenberg	108
Sciarra	111 Greenberg	108
Urman	112 Greenberg	108
Popp	113 Greenberg	108
Raphaely	114 King	100
Khoo	115 Raphaely	114
Baida	116 Raphaely	114
Tobias	117 Raphaely	114
Himuro	118 Raphaely	114
Colmenares	119 Raphaely	114
Weiss	120 King	100
Frapp	121 King	100

Kaufling	122 King	100
Vollman	123 King	100
Mourgos	124 King	100
Nayer	125 Weiss	120
Mikkilineni	126 Weiss	120
Landry	127 Weiss	120
Markle	128 Weiss	120
Bissot	129 Fripp	121
Atkinson	130 Fripp	121
Marlow	131 Fripp	121
Olson	132 Fripp	121
Mallin	133 Kaufling	122
Rogers	134 Kaufling	122
Gee	135 Kaufling	122
Philtanker	136 Kaufling	122
Ladwig	137 Vollman	123
Stiles	138 Vollman	123
Seo	139 Vollman	123
Patel	140 Vollman	123
Rajs	141 Mourgos	124
Davies	142 Mourgos	124

5. Измените `script_05_04.sql`, чтобы вывести на экран всех сотрудников, включая King, у которого нет никакого менеджера. Упорядочьте результаты по идентификатору сотрудника.. Сохраните SQL в текстовый файл `script_05_05.sql`. Выполните запрос в `script_05_05.sql`.

6. Создайте отчет для вывода фамилии сотрудника, номера отдела, и всех сотрудников, которые работают в том же самом отделе с данным сотрудником. Сохраните SQL в текстовый файл `script_05_06.sql`.

Ниже приведен результат выполнения.

SECTION WORKER COLLEAGUE

```
-----
```

20 Fay	Hartstein
20 Hartstein	Fay
30 Baida	Colmenares
30 Baida	Himuro
30 Baida	Khoo
30 Baida	Raphaely
30 Baida	Tobias
30 Colmenares	Baida
30 Colmenares	Himuro
30 Colmenares	Khoo
30 Colmenares	Raphaely
30 Colmenares	Tobias

30 Himuro	Baida
30 Himuro	Colmenares
30 Himuro	Khoo
30 Himuro	Raphaely
30 Himuro	Tobias
30 Khoo	Baida

7. Создайте отчет относительно тарифных разрядов и зарплат. Сначала выведите структуру таблицы WORK\_GRADES. Затем создайте запрос, который выводит на экран имя, работу, имя отдела, зарплату, и разряд для всех сотрудников.

8. Определите имена всех сотрудников, которые были наняты после Дэвиса. Создайте запрос, чтобы вывести на экран имя и дату найма любого сотрудника, нанятого после Дэвиса.

Ниже приведен результат выполнения.

LAST\_NAME                      HIRE\_DATE

```
-----
```

Bates	24-MAR-99
Kumar	21-APR-00
Hutton	19-MAR-97
Taylor	24-MAR-98
Livingston	23-APR-98
Grant	24-MAY-99
Johnson	04-JAN-00
Taylor	24-JAN-98
Fleaur	23-FEB-98
Sullivan	21-JUN-99
Geoni	03-FEB-00
Bull	20-FEB-97
Dellinger	24-JUN-98
Cabrio	07-FEB-99
Chung	14-JUN-97
Dilly	13-AUG-97
Gates	11-JUL-98
Perkins	19-DEC-99

9. Найдите имена и даты найма всех сотрудников, которые были наняты перед их менеджерами, наряду с именами их менеджеров и датами найма. Сохраните lab5\_09.sql.

Ниже приведен результат выполнения.

LAST_NAME	HIRE_DATE	LAST_NAME	HIRE_DATE
OConnell	21-JUN-99	Mourgos	16-NOV-99
Whalen	17-SEP-87	Kochhar	21-SEP-89
Hunold	03-JAN-90	De Haan	13-JAN-93
Faviet	16-AUG-94	Greenberg	17-AUG-94
Marlow	16-FEB-97	Fripp	10-APR-97
Ladwig	14-JUL-95	Vollman	10-OCT-97
Rajs	17-OCT-95	Mourgos	16-NOV-99
Davies	29-JAN-97	Mourgos	16-NOV-99
Matos	15-MAR-98	Mourgos	16-NOV-99
Vargas	09-JUL-98	Mourgos	16-NOV-99
King	30-JAN-96	Partners	05-JAN-97
Sully	04-MAR-96	Partners	05-JAN-97
McEwen	01-AUG-96	Partners	05-JAN-97
Ozer	11-MAR-97	Cambraut	15-OCT-99
Bloom	23-MAR-98	Cambraut	15-OCT-99
Fox	24-JAN-98	Cambraut	15-OCT-99
Smith	23-FEB-99	Cambraut	15-OCT-99

## Лабораторная работа 6. Применение подзапросов.

В этой лабораторной работе Вы научитесь записывать подзапросы для последующего применения их в сложных запросах данных.

1. Создайте запрос, который запрашивает фамилию сотрудника. Запрос выводит на экран фамилию и дату найма любого сотрудника в том же самом отделе, где и сотрудник, имя которого введено (исключая этого сотрудника). Например, если пользователь вводит Zlotkey, найдите всех сотрудников, которые работают с Zlotkey (исключая Zlotkey).

Ниже приведен результат выполнения.

Enter value for enter\_name: Zlotkey

LAST_NAME	HIRE_DATE
Russell	01-OCT-96
Partners	05-JAN-97
Errazuriz	10-MAR-97
Cambault	15-OCT-99
Tucker	30-JAN-97
Bernstein	24-MAR-97
Hall	20-AUG-97
Olsen	30-MAR-98

2. Создайте отчет, который выводит на экран идентификатор сотрудника, фамилию, и зарплату всех сотрудников, которые зарабатывают больше чем средняя зарплата. Отсортируйте результаты в порядке возрастающей зарплаты.

Ниже приведен результат выполнения.

WORKER_ID	LAST_NAME	SALARY
203	Mavris	6500
123	Vollman	6500
165	Lee	6800
113	Popp	6900
155	Tuvault	7000
178	Grant	7000
161	Sewall	7000
164	Marvins	7200
172	Bates	7300
171	Smith	7400
154	Cambault	7500

3. Напишите запрос, который выводит на экран номер сотрудника и фамилию всех сотрудников, которые работают в отделе с любым сотрудником, фамилия которого содержит букву *u*. Сохраните SQL в текстовый файл script\_06\_03.sql. Выполните запрос.

Ниже приведен результат выполнения.

WORKER\_ID LAST\_NAME

```
-----
107 Lorentz
106 Pataballa
105 Austin
104 Ernst
103 Hunold
119 Colmenares
118 Himuro
117 Tobias
116 Baida
115 Khoo
114 Raphaely
197 Feeney
196 Walsh
195 Jones
194 McCain
193 Everett
192 Bell
191 Perkins
190 Gates
```

4. Создайте отчет, который выводит на экран фамилию, номер отдела, и ID работы всех сотрудников, ID расположения отдела которых - 1700.

Ниже приведен результат выполнения.

LAST\_NAME SECTION\_ID WORK\_ID

```
-----
Whalen          10 AD_ASST
Higgins         110 AC_MGR
Gietz          110 AC_ACCOUNT
King           90 AD_PRES
Kochhar        90 AD_VP
De Haan        90 AD_VP
Greenberg      100 FI_MGR
Faviet         100 FI_ACCOUNT
Chen           100 FI_ACCOUNT
Sciarra        100 FI_ACCOUNT
Urman          100 FI_ACCOUNT
Popp           100 FI_ACCOUNT
Raphaely       30 PU_MAN
```



Khoo	30 PU_CLERK
Baida	30 PU_CLERK
Tobias	30 PU_CLERK
Himuro	30 PU_CLERK
Colmenares	30 PU_CLERK

18 rows selected.

Измените запрос так, чтобы пользователь вводил ID расположения. Сохраните script\_06\_04.sql.

5. Создайте отчет, который выводит на экран фамилию и зарплату каждого сотрудника, у которого менеджер King.

6. Создайте отчет, который выводит на экран номер отдела, фамилию, и ID работы для каждого сотрудника в отделе Executive.

Ниже приведен результат выполнения.

SECTION_ID	LAST_NAME	WORK_ID
90	King	AD_PRES
90	Kochhar	AD_VP
90	De Haan	AD_VP

7. Измените запрос в script\_06\_03.sql, чтобы вывести на экран номер сотрудника, фамилию, и зарплату всех сотрудников, которые зарабатывают больше чем средняя зарплата и кто работает в отделе с любым сотрудником, фамилия которого содержит букву *и*. Сохраните script\_06\_03.sql как script\_06\_07.sql. Выполните script\_06\_07.sql.

Ниже приведен результат выполнения.

WORKER_ID	LAST_NAME	SALARY
103	Hunold	9000
114	Raphaely	11000
123	Vollman	6500
122	Kaufling	7900
121	Fripp	8200
120	Weiss	8000
177	Livingston	8400
176	Taylor	8600
175	Hutton	8800
174	Abel	11000
172	Bates	7300

## Лабораторная работа 7. Применение операторов соединения.

В этой лабораторной работе Вы научитесь использовать операторы соединения, изучите виды и область применения операторов соединения.

1. Создайте список ID отдела для отделов, которые не содержат ID работы ST\_CLERK. Используйте операторы соединения, чтобы создать этот отчет.

Ниже приведен результат выполнения.

SECTION\_ID

-----

10  
20  
30  
40  
60  
70  
240  
250  
260  
270

26 rows selected.

2. Создайте список стран, у которых нет никаких отделов, расположенных в них. Выведите на экран ID страны и имя стран. Используйте операторы соединения, чтобы создать этот отчет.

Ниже приведен результат выполнения.

CO LAND\_NAME

-----

AR Argentina  
BE Belgium  
DK Denmark  
EG Egypt  
FR France  
HK HongKong  
IL Israel  
KW Kuwait  
NG Nigeria  
ZM Zambia  
ZW Zimbabwe

11 rows selected.

3. Создайте список работ для отделов 10, 50, и 20, в этом порядке. Выведите ID работы и ID отдела, используя операторы соединения.

Ниже приведен результат выполнения.

```
WORK_ID  SECTION_ID
-----
AD_ASST      10
SH_CLERK     50
ST_CLERK     50
ST_MAN       50
MK_MAN       20
MK_REP       20
```

6 rows selected.

4. Создайте отчет, который выводит ID сотрудника и ID работы для тех сотрудников, у которых в настоящий момент должность, совпадает с первоначальной должностью (то есть, они сменили работу, но теперь вернулись к выполнению их первоначальной работе).

Ниже приведен результат выполнения.

```
WORKER_ID WORK_ID
-----
      176 SA_REP
      200 AD_ASST
```

5. Создайте отчет со следующими данными:

- Фамилия и ID отдела всех сотрудников из таблицы WORKERS, независимо от того, принадлежат ли они отделу
- ID отдела и имя отдела всех отделов из таблицы SECTIONS, независимо от того, есть ли у них сотрудники, работающие в них

Напишите составной запрос.

Ниже приведен результат выполнения.

```
LAST_NAME          SECTION_ID TO_CHAR(NULL)
-----
Smith              80
Stiles             50
Sullivan           50
Sully              80
Taylor             80
```

Tobias	30
Vargas	50
Vishney	80
Vollman	50
Weiss	50
Whalen	10
Zlotkey	80
	10 Administration
	20 Marketing
	30 Purchasing
	40 Human Resources
	50 Shipping
	60 IT
	70 Public Relations
	80 Sales
	90 Executive
	100 Finance
	110 Accounting
	120 Treasury
	130 Corporate Tax

## Лабораторная работа 8. Управление данными.

В этой лабораторной работе Вы научитесь применять язык манипуляций данными (DML), выполнять вставку, обновление и удаления данных из различных таблиц, изучите влияние ограничений для столбцов на операции DML.

Напишите SQL, чтобы вставить, обновить, и удалить данные сотрудника. Как прототип, Вы используете таблицу MY\_WORKER.

Вставьте данные в таблицу MY\_WORKER.

1. Выполните оператор в script\_08\_01.sql сценарии, чтобы создать таблицу MY\_WORKER.
2. Выведите структуру таблицы MY\_WORKER, чтобы определить имена столбцов.

Name	Null?	Type
-----		
ID	NOT NULL	NUMBER(4)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
USERID		VARCHAR2(8)
SALARY		NUMBER(9,2)

3. Создайте INSERT, чтобы добавить *первую строку* данных к таблице MY\_WORKER со следующими данными. Не перечисляйте столбцы в INSERT. *Не вводите все строки.*

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750
5	Ropeburn	Audrey	aropebur	1550

4. Заполните таблицу MY\_WORKER для второй строки из предыдущего списка. Перечислите столбцы явно в INSERT.
5. Проверьте дополнение к таблице (select).
6. Запишите оператор вставки для повторного использования в файле сценария, чтобы загрузить строки в таблицу MY\_WORKER. Сохраните script\_08\_06.sql.
7. Заполните таблицу следующими двумя строками из предыдущего списка, выполняя оператор вставки в сценарии, который Вы создали.
8. Проверьте дополнение к таблице.

9. Сделайте добавление данных постоянными.  
Обновите и удалите данные в таблице MY\_WORKER.
  10. Измените фамилию сотрудника номер 3 на Drexler.
  11. Измените зарплату на 1 000\$ для всех сотрудников, у которых зарплата меньше чем 900\$.
  12. Проверьте изменения в таблице.
  13. Удалите Бетти Дэнкс из таблицы MY\_WORKER.
  14. Проверьте изменения в таблице.
  15. Фиксируйте все изменения.
  16. Заполните таблицу последней строкой данных в списке, приведенном на шаге 3 при использовании сценария, который Вы создали на шаге 6. Выполните операторы в сценарии.
  17. Проверьте дополнение к таблице.
  18. Отметьте промежуточную точку в транзакции.
  19. Очистите всю таблицу.
  20. Проверьте, что таблица пуста.
  21. Отмените delete, не отменяя insert.
  22. Проверьте, что введенная строка присутствует.
- Ниже приведен результат выполнения.

```
SELECT * FROM my_WORKER;
```

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
1	Grant	Douglas	dgrant	2600
2	Victor	Danal	dvictor	5500
4	Victor	Rafael	rvictor	4500

## Лабораторная работа 9. Создание и управление таблицами с помощью операторов DDL.

В этой лабораторной работе Вы научитесь применять язык определения данных (DDL), создавать различные таблицы в базе данных, переопределять их структуру, изучите виды ограничений, определяемых на столбцы таблиц, рассмотрите создание первичных и внешних ключей для столбцов.

1. Создайте таблицу DEPT, исходя из следующей таблицы. Сохраните script\_09\_01.sql. Проверьте, что таблица создана.

<b>Column Name</b>	ID	NAME
<b>Key Type</b>	Primary key	
<b>Nulls/Unique</b>		
<b>FK Table</b>		
<b>FK Column</b>		
<b>Data type</b>	NUMBER	VARCHAR2
<b>Length</b>	7	25

2. Заполните таблицу DEPT данными из таблицы SECTIONS. Включите только необходимые столбцы.

3. Создайте таблицу EMP, исходя из следующей таблицы. Сохраните script\_09\_03.sql, Проверьте, что таблица создана.

<b>Column Name</b>	ID	LAST_NAME	FIRST_NAME	DEPT_ID
<b>Key Type</b>				
<b>Nulls/Unique</b>				
<b>FK Table</b>				DEPT
<b>FK Column</b>				ID
<b>Data type</b>	NUMBER	VARCHAR2	VARCHAR2	NUMBER
<b>Length</b>	7	25	25	7

NAME	NULL	TYPE
<b>ID</b>		NUMBER(7)
<b>LAST_NAME</b>		VARCHAR(25)
<b>FIRST_NAME</b>		VARCHAR(25)
<b>DEPT_ID</b>		NUMBER(7)

4. Создайте таблицу WORKERS2, основанную на структуре таблицы WORKERS. Включите только WORKER\_ID, FIRST\_NAME, LAST\_NAME, SALARY, и SECTION\_ID. Назовите столбцы в новой таблице ID, FIRST\_NAME, LAST\_NAME, SALARY, и DEPT\_ID, соответственно.

5. Удалите таблицу EMP



## Лабораторная работа 10. Создание объектов схемы.

В этой лабораторной работе Вы научитесь применять операторы DDL - язык определения данных для создания представлений в базе данных, переопределять их структуру, выполнять изменение и удаление представлений.

1. Необходимо скрыть некоторые из данных в таблице WORKERS. Создайте представление под названием WORKERS\_VU, основанное на номере сотрудника, имени сотрудника, и номере отдела из таблицы WORKERS.

2. Выведите на экран содержание из представления WORKERS\_VU. Ниже приведен результат выполнения.

WORKER\_ID WORKER SECTION\_ID

---

198	OConnell	50
199	Grant	50
200	Whalen	10
201	Hartstein	20
202	Fay	20
203	Mavris	40
204	Baer	70
205	Higgins	110
206	Gietz	110
100	King	90
101	Kochhar	90
102	De Haan	90
103	Hunold	60
104	Ernst	60
105	Austin	60
106	Pataballa	60
107	Lorentz	60
108	Greenberg	100
127	Landry	50
128	Markle	50
129	Bissot	50
130	Atkinson	50
141	Rajs	50
142	Davies	50
143	Matos	50
149	Zlotkey	80
150	Tucker	80
151	Bernstein	80
152	Hall	80

3. Используя представление WORKERS\_VU, напишите запрос, чтобы вывести на экран все имена сотрудника и номер отдела. Ниже приведен результат выполнения.

WORKER	SECTION_ID
OConnell	50
Grant	50
Whalen	10
Hartstein	20
Fay	20
Mavris	40
Baer	70
Higgins	110
Gietz	110
King	90
Kochhar	90
De Haan	90
Hunold	60
Ernst	60
Austin	60
Pataballa	60
Lorentz	60
Greenberg	100
Faviet	100
Chen	100

4. Отделу 50 необходим доступ к своим данным. Создайте представление под названием DEPT50, которое содержит номер сотрудника, фамилию сотрудника, и номер отдела для всех сотрудников в отделе 50.

5. Выведите на экран структуру и содержание представления DEPT50.

Ниже приведен результат выполнения.

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(6)
WORKER	NOT NULL	VARCHAR2(25)
DEPTNO		NUMBER(4)

Содержание представления DEPT50:

EMPNO WORKER	DEPTNO
120 Weiss	50
121 Fripp	50
122 Kaufling	50
123 Vollman	50
124 Mourgos	50
125 Nayer	50
126 Mikkilineni	50
127 Landry	50
129 Bissot	50
130 Atkinson	50
131 Marlow	50
132 Olson	50
133 Mallin	50
134 Rogers	50
135 Gee	50
136 Philtanker	50
137 Ladwig	50
138 Stiles	50
140 Patel	50
141 Rajs	50
183 Geoni	50
184 Sarchand	50
185 Bull	50
187 Cabrio	50
188 Chung	50
189 Dilly	50
190 Gates	50
191 Perkins	50
192 Bell	50
193 Everett	50
194 McCain	50
195 Jones	50

7. Необходимо создать последовательность, которая может использоваться для столбца первичного ключа таблицы DEPT. Последовательность должна начинаться с 200 и иметь максимальное значение 1 000. Инкремент последовательности 10. Назовите последовательность DEPT\_ID\_SEQ.

8. Напишите сценарий, чтобы вставить две строки в таблицу DEPT. Назовите сценарий script\_10\_08.sql. Добавьте два отдела: Education и Administration. Проверьте изменения. Выполните сценарий.

9. Создайте групповой индекс на столбце NAME в таблице ОТДЕЛА.

10. Создайте синоним для таблицы WORKERS.

## Лабораторная работа 11. Управление объектами с помощью словаря базы данных.

В этой лабораторной работе Вы научитесь использовать представления словаря базы данных для получения информации об объектах базы данных, их структуре.

1. Для указанной таблицы, создайте сценарий, который сообщает об именах столбцов, типах данных, длине типов данных, и разрешен ли NULL. Запросите пользователя ввести имя таблицы. Дайте соответствующие псевдонимы столбцам DATA\_PRECISION и DATA\_SCALE. Сохраните этот сценарий в файле, script\_11\_01.sql.

Например, если пользователь вводит WORKS, следующие результаты:

```
COLUMN_NAME DATA_TYPE DATA_LENGTH PRECISION SCALE N
-----
WORK_ID      VARCHAR2      10              N
WORK_TITLE   VARCHAR2      35              N
MIN_SALARY   NUMBER         22      6      0 Y
MAX_SALARY   NUMBER         22      6      0 Y
```

2. Создайте сценарий, который сообщает об имени столбца, имени ограничения, типе ограничения, условии поиска, и состоянии для указанной таблицы. Следует использовать USER\_CONSTRAINTS и USER\_CONS\_COLUMNS, чтобы получить эту информацию. Запросите пользователя ввести имя таблицы. Сохраните сценарий в файле, script\_11\_02.sql.

Например, если пользователь вводит SECTIONS, следующие результаты:

```
COLUMN_NAME CONSTRAINT_NAME C SEARCH_CONDITION
STATUS
-----
SECTION_NAME DEPT_NAME_NN      C "SECTION_NAME" IS NOT
NULL ENABLED

SECTION_ID DEPT_ID_PK      P ENABLED

MANAGER_ID DEPT_MGR_FK    R ENABLED

AREA_ID DEPT_LOC_FK      R ENABLED
```

3. Добавьте комментарий к таблице SECTIONS. Затем запросите представление USER\_TAB\_COMMENTS, чтобы проверить, что комментарий присутствует.

COMMENTS

---

Company information.

4. Найдите имена всех синонимов, которые находятся в Вашей схеме. Ниже приведен результат выполнения.

SYNONYM\_NAME TABLE\_OWNER TABLE\_NAME DB\_LINK

---

EMP STAFF WORKERS

5. Вы должны определить имена и определения всех представлений в Вашей схеме. Создайте отчет, который получает информацию о представлении: имя представления и текст из представления словаря данных USER\_VIEWS.

Ниже приведен результат выполнения.

VIEW\_NAME TEXT

---

EMP\_DETAILS\_VIEW

SELECT

e.WORKER\_id,  
e.WORK\_id,  
e.manager\_id,  
e.SECTION\_id,  
d.AREA\_id,  
l.LAND\_id,  
e.first\_name,  
e.last\_name,  
e.salary,  
e.commission\_pct,  
d.SECTION\_name,  
j.WORK\_title,  
l.city,  
l.state\_province,  
c.LAND\_name,  
r.PLACE\_name

FROM

WORKERs e,  
SECTIONs d,  
WORKs j,

```

AREAs l,
LANDS c,
PLACES r
WHERE e.SECTION_id = d.SECTION_id
      AND d.AREA_id = l.AREA_id
      AND l.LAND_id = c.LAND_id
      AND c.PLACE_id = r.PLACE_id
      AND j.WORK_id = e.WORK_id
WITH READ ONLY

```

```

DEPT50
SELECT WORKER_id empno, last_name WORKER,
SECTION_id deptno
FROM WORKERs
WHERE SECTION_id = 50
WITH CHECK OPTION

```

```

WORKERS_VU
SELECT WORKER_id, last_name WORKER, SECTION_id
FROM WORKERs

```

6. Найдите имена последовательностей. Напишите запрос, чтобы вывести на экран следующую информацию о последовательностях: имя последовательности, максимальное значение, приращение, и последнее число. Назовите сценарий script\_11\_06.sql.

Ниже приведен результат выполнения.

SEQUENCE_NAME	MAX_VALUE	INCREMENT	BY	LAST_NUMBER
AREAS_SEQ	9900	100		3300
SECTIONS_SEQ	9990	10		280
WORKERS_SEQ	1.0000E+27	1		207
DEPT_ID_SEQ	1000	10		400

## Лабораторная работа 12. Управление доступом пользователей.

В этой лабораторной работе Вы научитесь использовать операции DCL для предоставления или отзыва доступа пользователей, изучите виды привилегий и способы управления доступом на основе ролей.

1. Какое полномочие пользователю нужно дать, чтобы войти в систему Oracle?
2. Какое полномочие пользователю нужно дать, чтобы создавать таблицы?
3. Если Вы создаете таблицу, кто может предоставить полномочия другим пользователям для Вашей таблицы?
4. Вы - DBA. Вы создаете много пользователей, которые требуют тех же самых системных полномочий. Что следует использовать?
5. Какую команду Вы используете, чтобы изменить Ваш пароль?
6. Предоставьте пользователю ORA1 доступ к своей таблице SECTIONS.
7. Запросите все строки в таблице SECTIONS.

SECTION_ID	SECTION_NAME	MANAGER_ID	AREA_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700
100	Finance	108	1700
110	Accounting	205	1700

8. Добавьте новую строку к своей таблице SECTIONS. Добавьте Education как отдел номер 500.
9. Создайте синоним для таблицы SECTIONS.
10. Запросите все строки в таблице SECTIONS, используя синоним.
11. Запросите словарь данных USER\_TABLES, чтобы получить информацию о доступных таблицах,.

TABLE\_NAME

-----  
PLACES  
AREAS  
SECTIONS  
WORKS

DEPT  
 MY\_WORKER  
 WORKERS2  
 LANDS  
 WORKERS  
 WORK\_STORY  
 10 rows selected.

12. Запросите представление словаря данных ALL\_TABLES, чтобы видеть информацию обо всех таблицах, к которым можно получить доступ. Исключите таблицы, которыми Вы владеете.

TABLE_NAME	OWNER
-----	-----
DUAL	SYS
SYSTEM_PRIVILEGE_MAP	SYS
TABLE_PRIVILEGE_MAP	SYS
STMT_AUDIT_OPTION_MAP	SYS
AUDIT_ACTIONS	SYS
DEF\$_TEMP\$LOB	SYS TEM
WM\$WORKSPACES_TABLE	WMSYS
WM\$VERSION_TABLE	WMSYS
WM\$NEXTVER_TABLE	WMSYS
WM\$VERSION_HIERARCHY_TABLE	WMSYS
HELP	SYS TEM
DR\$OBJECT_ATTRIBUTE	CTXSYS
DR\$POLICY_TAB	CTXSYS
DR\$NUMBER_SEQUENCE	CTXSYS
OGIS_SPATIAL_REFERENCE_SYSTEMS	MDSYS
OGIS_GEOMETRY_COLUMNS	MDSYS
SDO_UNITS_OF_MEASURE	MDSYS
SDO_PRIME_MERIDIANS	MDSYS
SDO_COORD_OP_PARAMS	MDSYS
SDO_COORD_OP_PARAM_USE	MDSYS
SDO_XML_SCHEMAS	MDSYS
AW\$AWMD	SYS
AW\$EXPRESS	SYS
AW\$AWCREATE	SYS
AW\$AWCREATE10G	SYS
AW\$AWXML	SYS
AW\$AWREPORT	SYS
MVIEW\$_ADV_INDEX	SYS TEM
MVIEW\$_ADV_PARTITION	SYS TEM
OLAPI_STORY	SYS
OLAPI_SESSION_STORY	SYS
OLAPI_IFACE_OBJECT_STORY	SYS



### Лабораторная работа 13. Введение в PL /SQL.

В этой лабораторной работе Вы научитесь описывать язык PL/SQL, понимать его возможности и преимущества использования, определять и использовать PL/SQL-переменные.

1. Какие из следующих PL/ SQL блоков выполняются успешно?
  - a. BEGIN  
END;
  - b. DECLARE  
amount INTEGER(10);  
END;
  - c. DECLARE  
BEGIN  
END;
  - d. DECLARE  
amount INTEGER(10);  
BEGIN  
DBMS\_OUTPUT.PUT\_LINE(amount);  
END;
  
2. Создайте и выполните анонимный блок, который выводит “Hello Word” Выполните и сохраните этот сценарий как script\_01\_02\_soln.sql.

### Лабораторная работа 14. Объявление переменных PL/SQL

В этой лабораторной работе Вы научитесь определять и создавать PL/SQL-переменные, изучите возможности применения PL/SQL переменных.

Рекомендуется использовать *iSQL\*Plus* для этой практики.

1. Определите допустимые и недопустимые имена идентификатора:
  - a. today
  - b. last\_name
  - c. today's\_date
  - d. Number\_of\_days\_in\_February\_this\_year
  - e. Isleap\$year
  - f. #number
  - g. NUMBER#
  - h. number1to7

2. Определите допустимые и недопустимые определения переменных и инициализации:

- a.                    number\_of\_copies    PLS\_INTEGER;
- b.                    printer\_name            constant VARCHAR2(10);
- c.                    deliver\_to             VARCHAR2(10):=Johnson;
- d.                    by\_when                DATE:= SYSDATE+1;

3. Исследуйте следующий анонимный блок и выберите соответствующий пункт.

```
SET SERVEROUTPUT ON
DECLARE
  fname VARCHAR2(20);
  lname VARCHAR2(15) DEFAULT 'fernandez';
BEGIN
  DBMS_OUTPUT.PUT_LINE( FNAME || ' ' ||lname);
END;
```

- a.    Блок выполнится успешно и напечатает 'fernandez'
- b.    Блок выдаст ошибку, потому что fname переменная используется без инициализации.
- c.    Блок выполнится успешно и напечатает 'null fernandez'
- d.    Блок выдаст ошибку, потому что невозможно использовать ключевое слово DEFAULT, чтобы инициализировать переменную типа VARCHAR2.
- e.    Блок выдаст ошибку, потому что переменный FNAME не объявлена.

4.    Создайте анонимный блок. В *iSQL\*Plus*, загрузите сценарий `script_01_02_soln.sql`.

a.    Добавьте декларативный раздел к этому PL / SQL блоку. В декларативном разделе, объявите следующие переменные:

- 1. Переменная `today` типа `DATE`. Инициализируйте `today` `SYSDATE`.
- 2. Переменная `tomorrow` типа `today`. Используйте атрибут `%TYPE`.

б.    В исполняемом разделе инициализируйте переменную `tomorrow` выражением, которое вычисляет завтрашнюю дату (добавьте к значению `today` 1). Напечатайте значение `today` и `tomorrow` после печати 'Hello World' Выполните и сохраните этот сценарий как `script_02_04_soln.sql`.

5.    Отредактируйте `script_02_04_soln.sql` сценарий.

a.    Добавьте код, чтобы создать две связанные переменные.

b.    Создайте связанные переменные `basic_percent` и `pf_percent` типа `NUMBER`.

c.    В исполнимом разделе PL / SQL блока присвойте значения 45 и 12 `basic_percent` и `pf_percent` соответственно.

d.    Завершите PL / SQL блок "/" и выведите на экран значение связанных переменных при использовании команды `PRINT`.

e.    Выполните и сохраните файл сценария как `script_02_05_soln.sql`.

Нажмите кнопку Next Page.

## Лабораторная работа 15. Исполняемые операторы.

В этой лабораторной работе Вы научитесь применять язык PL/SQL, понимать его возможности и отличия от SQL, записывать исполняемые операторы PL/SQL.

Рекомендуется использовать *iSQL\*Plus* для этой практики.

### PL / SQL Блок

```
DECLARE
weight  NUMBER(3) := 600;
message VARCHAR2(255) := 'Product 10012';
BEGIN
  DECLARE
    weight  NUMBER(3) := 1;
    message VARCHAR2(255) := 'Product 11001';
    new_locn VARCHAR2(50) := 'Europe';
  BEGIN
    weight := weight + 1;           позиция 1
    new_locn := 'Western ' || new_locn;
  END;
  weight := weight + 1;           позиция 2
  message := message || ' is in stock';
  new_locn := 'Western ' || new_locn;
END;
/
```

1. В PL / SQL блоке выше определите тип данных и значение каждой из следующих переменных согласно правилам определения области.

- a. Значение `weight` в позиции 1:
- b. Значение `new_locn` в позиции 1:
- c. Значение `weight` в позиции 2:
- d. Значение `message` в позиции 2:
- e. Значение `new_locn` в позиции 2:

### Пример

```
DECLARE
customer  VARCHAR2(50) := 'Womansport';
credit_rating  VARCHAR2(50) := 'EXCELLENT';
BEGIN
  DECLARE
```

```

customer    NUMBER(7) := 201;
name        VARCHAR2(25) := 'Unisports';
BEGIN
    credit_rating := 'GOOD';
    ...
END;

...
END;
/

```

2. В PL / SQL блоке, показанном выше, определите значения и типы данных для каждого из следующих случаев.

- a. Значение `customer` во вложенном блоке:
- b. Значение `name` во вложенном блоке:
- c. Значение `credit_rating` во вложенном блоке:
- d. Значение `customer` в основном блоке:
- e. Значение `name` в основном блоке:
- f. Значение `credit_rating` в основном блоке:

3. Используйте тот же самый сеанс, что в уроке 2. Если Вы открыли новый сеанс, то выполните `script_02_05_soln.sql`. Отредактируйте `script_02_05_soln.sql`.

- a. Используйте однострочный синтаксис комментариев, чтобы прокомментировать строки, которые создают связанные переменные.
- b. Используйте многострочные комментарии в исполнимом разделе, чтобы прокомментировать строки, которые присваивают значения связанным переменным.
- c. Объявите две переменные: `fname` типа `VARCHAR2` и размером 15, и `emp_sal` типа `NUMBER` и размера 10.
- d. Включайте следующий SQL-оператор в исполнимый раздел:  

```

SELECT first_name, salary
INTO fname, emp_sal FROM WORKERS
WHERE WORKER_id=110;

```
- e. Измените строку, которая печатает 'Hello World', чтобы напечатать 'Hello' и имя. Прокомментируйте строки, которые выводят на экран даты и печатают связанные переменные.
- f. Вычислите премиальные сотрудника из предусмотренного фонда (PF).
- g. PF составляет 12 % основного оклада, и основной оклад составляет 45 % зарплаты. Используйте связанные переменные для вычисления. Выведите зарплату сотрудника и его премиальные PF.
- h. Выполните и сохраните свой сценарий как `script_03_03_soln.sql`.

4. Измените сценарий, который создали в упражнении 3, чтобы выполнить ввод данных пользователем.
  - a. Загрузите файл сценария `script_03_04.sql`.
  - b. Включайте команду `PROMPT`, чтобы запросить пользователя: 'Пожалуйста, введите номер сотрудника.'
  - d. Измените объявление `empno` переменной, чтобы выполнить ввод данных пользователем.
  - e. Измените оператор выбора, чтобы включать переменную `empno`.
  - f. Выполните и сохраните свой сценарий как `script_03_04_soln.sql`.

Введите 100 и нажмите кнопку Continue.

## Лабораторная работа 16. Взаимодействие с Oracle Server

В этой лабораторной работе Вы научитесь осуществлять взаимодействие с Oracle Server, применять язык PL/SQL, записывать исполняемые анонимные блоки PL/SQL.

Рекомендуется использовать *iSQL\*Plus* для этой практики.

1. Создайте PL / SQL блок, который выбирает максимальный ID отдела в таблице SECTIONS и хранит это в max\_deptno переменной. Выведите на экран максимальный ID отдела.

- a. Объявите переменную max\_deptno типа NUMBER в декларативном разделе.
- b. Начните исполнимый раздел с ключевым словом BEGIN и включите оператор SELECT, чтобы получить максимальный SECTION\_id из таблицы отделов.
- c. Выведите на экран max\_deptno и закончите исполнимый блок.
- d. Выполните и сохраните свой сценарий как script\_04\_01\_soln.sql.

2. Измените PL / SQL блок, который Вы создали в упражнении 1, чтобы ввести новый отдел в таблицу SECTIONS.

- a. Загрузите сценарий script\_04\_01\_soln.sql. Объявите две переменные:
- b. dept\_name типа SECTIONS.SECTION\_name.
- c. Связанную переменную dept\_id типа NUMBER.
- d. Присвойте ' Education ' к dept\_name в декларативном разделе.
- e. Получите текущее максимальное число отдела из таблицы SECTIONS. Добавьте 10 к этому и присвойте результат dept\_id.
- f. Включите оператор INSERT, чтобы вставить данные в SECTION\_name, SECTION\_id, и AREA\_id столбцы таблицы SECTIONS.
- g. Используйте значения в dept\_name, dept\_id для SECTION\_name, SECTION\_id и используйте null для AREA\_id.
- h. Используйте атрибут SQL%ROWCOUNT, чтобы вывести на экран число измененных строк.
- i. Выполните оператор выбора, чтобы проверить, введен ли новый отдел. Можно завершить PL / SQL блок с “/” и включить оператор SELECT в сценарий.
- j. Выполните и сохраните свой сценарий как script\_04\_02\_soln.sql.

3. Создайте PL / SQL блок, который изменяет AREA\_id на 3000 для нового отдела. Используйте связанную переменную dept\_id, чтобы обновить строку.

- a. Если Вы запустили новый SQL\*Plus сеанс, удалите отдел, который Вы добавили в таблице SECTIONS и выполняете сценарий script\_04\_02\_soln.sql.
- b. Начните исполняемый блок ключевым словом BEGIN. Включите оператор UPDATE, чтобы установить AREA\_id в 3000 для нового отдела. Используйте связанную переменную dept\_id в операторе UPDATE.
- c. Закончите исполнимый блок ключевым словом END. Завершите PL/SQL блок “/” и включите оператор SELECT, чтобы вывести на экран отдел, который Вы обновили.
- d. Включите Оператор DELETE, чтобы удалить отдел, который Вы добавили.
- e. Выполните и сохраните свой сценарий как script\_04\_03\_soln.sql.

## Лабораторная работа 17. Управляющие структуры.

В этой лабораторной работе Вы научитесь применять управляющие структуры в исполняемых анонимных блоках PL/SQL, осуществлять различную логику в блоке.

Рекомендуется использовать *iSQL\*Plus* для этой практики.

1. Выполните команду в файле `script_05_01.sql`, чтобы создать таблицу `messages`. Запишите PL / SQL блок, чтобы вставить числа в таблицу `messages`.
  - a. Вставьте номера 1 - 10, исключая 6 и 8.
  - b. Фиксируйте вставку перед концом блока.
  - c. Выполните оператор `SELECT`, чтобы проверить, что Ваш PL / SQL блок работает.
  
2. Выполните сценарий `script_05_02.sql`. Этот сценарий создает `emp` таблицу, которая является копией таблицы `WORKERS`. Измените `emp` таблицу, чтобы добавить новый столбец `stars`, типа данных `VARCHAR2` и размера 50. Создайте PL / SQL блок, который вставляет звездочку в столбец `stars` за каждые 1000\$ зарплаты сотрудника. Сохраните свой сценарий как `script_05_02_soln.sql`.
  - a. Используйте команду `DEFINE`, чтобы определить переменную, `empno` и инициализируйте переменную значением 176.
  - b. Начните декларативный раздел блока и передайте значение `empno` к PL / SQL блоку через переменную подстановки *iSQL\*Plus*. Объявите переменную `asterisk` типа `emp.stars` и инициализируйте его к `NULL`. Создайте переменную `sal` типа `emp.salary`.
  - c. В исполнимом разделе, запишите логику, чтобы добавить звездочку (\*) к строке за каждые 1000\$ зарплаты. Например, если сотрудник зарабатывает 8000\$, строка звездочек должна содержать восемь звездочек. Если сотрудник зарабатывает 12500\$, строка звездочек должна содержать 13 звездочек.
  - d. Обновите столбец `stars` для сотрудника со строкой `asterisks`. Выполните фиксацию перед концом блока.
  - e. Выведите на экран строку из `emp` таблицы, чтобы проверить, выполняется ли Ваш PL / SQL блок успешно.
  - f. Выполните и сохраните свой сценарий как `script_05_02_soln.sql`.



## Лабораторная работа 18. Создание хранимых процедур и функций.

В этой лабораторной работе Вы научитесь создавать хранимы в базе данных PL/SQL конструкции, создавать и управлять процедурами и функциями, изучите структуру и необходимые секции для процедур и функций.

1. В *iSQL\*Plus*, загрузите сценарий `script_02_04_soln.sql`
  - a. Измените сценарий, чтобы преобразовать анонимный блок в процедуру, `greet`.
  - b. Выполните сценарий, чтобы создать процедуру.
  - c. Сохраните свой сценарий как `script_06_01_soln.sql`.
  - d. Нажмите кнопку `Clear`, чтобы очистить рабочее пространство.
  - e. Создайте и выполните анонимный блок, чтобы вызвать процедуру `greet`.
2. Загрузите сценарий `script_06_01_soln.sql`.

Удалите процедуру `greet`: `DROP PROCEDURE greet`

- a. Измените процедуру, чтобы принимать параметр `name` типа `VARCHAR2`.
- b. Выведите `Hello< name >` вместо, `Hello World`.
- c. Сохраните свой сценарий как `script_06_02_soln.sql`.
- d. Выполните сценарий, чтобы создать процедуру.
- e. Создайте и выполните анонимный блок, чтобы вызвать процедуру `greet` с параметром.

## Литература

1. Андон Ф., Резниченко В. Язык запросов SQL. Учебный курс. - СПб.: Питер; Киев: Издательская группа BHV, 2006. - 416 с.
2. Кристофер Аллен. Oracle PL/SQL Как писать мощные и гибкие программы на PL/SQL.- М.:Изд-во Лори,2005.-369с.
3. Том Кайт. Oracle для профессионалов. Пер. с англ./ТомКайт- СПб.: ООО «ДиаСофтЮП», 2003. - 672 с.
4. С.Фейерштейн, Б.Прибыл.Oracle PL/SQL для профессионалов.3-е издание.- СПб.: Питер; 2004. - 941 с.
5. Oracle Database Installation Guide, 11g for Linux 2007, Oracle.- 246 с.
6. Том Кайт. Oracle для профессионалов. Архитектура, методики программирования и особенности версий 9i, 10g и 11g. 2011- 842 стр.
7. Сэм Р. Алапати. Oracle Database 11g: руководство администратора баз данных. Диалектика.2010 - 1440 стр.
8. Алекс Кригель, Борис Трухнов. SQL. Библия пользователя. Язык запросов SQL 2-е издание. Диалектика.2009 - 752 стр.
9. Oracle 11g. Основы 4-е издание. Символ-Плюс.2009 - 1016 стр.
10. Том Кайт. Oracle для профессионалов. Пер. с англ./ТомКайт- СПб.: ООО «ДиаСофтЮП», 2003. - 672 с.

## Приложение 1. Таблицы схемы STAFF (персонал)

Описание таблицы SECTIONS.

Имя столбца	Тип данных
SECTION_ID	NUMBER(4)
SECTION_NAME	VARCHAR2(30)
MANAGER_ID	NUMBER(6)
AREA_ID	NUMBER(4)

Описание таблицы WORKERS;

Имя столбца	Тип данных
WORKER_ID	NUMBER(6)
FIRST_NAME	VARCHAR2(20)
LAST_NAME	VARCHAR2(25)
EMAIL	VARCHAR2(25)
PHONE_NUMBER	VARCHAR2(20)
HIRE_DATE	DATE
WORK_ID	VARCHAR2(10)
SALARY	NUMBER(8,2)
COMMISSION_PCT	NUMBER(2,2)
MANAGER_ID	NUMBER(6)
SECTION_ID	NUMBER(4)

Описание таблицы WORKS;

Имя столбца	Тип данных
WORK_ID	VARCHAR2(10)
WORK_TITLE	VARCHAR2(35)
MIN_SALARY	NUMBER(6)
MAX_SALARY	NUMBER(6)

Описание таблицы AREAS;

Имя столбца	Тип данных
AREA_ID	NUMBER(4)
STREET_ADDRESS	VARCHAR2(40)
POSTAL_CODE	VARCHAR2(12)
CITY	VARCHAR2(30)
STATE_PROVINCE	VARCHAR2(25)

LAND\_ID CHAR(2)

Описание таблицы PLACES;

Имя столбца	Тип данных
PLACE_ID	NUMBER
PLACE_NAME	VARCHAR2(25)

Описание таблицы WORK\_STORY;

Имя столбца	Тип данных
WORKER_ID	NUMBER(6)
START_DATE	DATE
END_DATE	DATE
WORK_ID	VARCHAR2(10)
SECTION_ID	NUMBER(4)

Описание таблицы LANDS ;

Имя столбца	Тип данных
LAND_ID	CHAR(2)
LAND_NAME	VARCHAR2(40)
PLACE_ID	NUMBER



В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Кафедра **Программных систем** входит в состав нового факультета **Инфокоммуникационные технологии**, созданного решением Ученого совета университета 17 декабря 2010 г. по предложению инициативной группы сотрудников, имеющих большой опыт в реализации инфокоммуникационных проектов федерального и регионального значения.

На кафедре ведется подготовка бакалавров и магистров по направлению **210700 «Инфокоммуникационные технологии и системы связи»:**

**210700.62.10 – ИНТЕЛЛЕКТУАЛЬНЫЕ  
ИНФОКОММУНИКАЦИОННЫЕ СИСТЕМЫ (Бакалавр)**

**210700.68.10 – ИНТЕЛЛЕКТУАЛЬНЫЕ  
ИНФОКОММУНИКАЦИОННЫЕ СИСТЕМЫ (Магистр)**

Выпускники кафедры получают фундаментальную подготовку по: математике, физике, электронике, моделированию и проектированию инфокоммуникационных систем (ИКС), информатике и программированию, теории связи и теории информации.

В рамках профессионального цикла изучаются дисциплины: архитектура ИКС, технологии программирования, ИКС в Интернете, сетевые технологии, администрирование сетей Windows и UNIX, создание программного обеспечения ИКС, Web программирование, создание клиент-серверных приложений.

**Область профессиональной деятельности бакалавров и магистров включает:**

- сервисно-эксплуатационная в сфере современных ИКС;
- расчетно-проектная при создании и поддержке сетевых услуг и сервисов;

- экспериментально-исследовательская;
- организационно-управленческая – в сфере информационного менеджмента ИКС.

#### **Знания выпускников востребованы:**

- в технических и программных системах;
- в системах и устройствах звукового вещания, электроакустики, речевой, и мультимедийной информатики;
- в средствах и методах защиты информации;
- в методах проектирования и моделирования сложных систем;
- в вопросах передачи и распределения информации в телекоммуникационных системах и сетях;
- в методах управления телекоммуникационными сетями и системами;
- в вопросах создания программного обеспечения ИКС.

#### **Выпускники кафедры Программных систем обладают компетенциями:**

- проектировщика и разработчика структур ИКС;
- специалиста по моделированию процессов сложных систем;
- разработчика алгоритмов решения задач ИКС;
- специалиста по безопасности жизнедеятельности ИКС;
- разработчика сетевых услуг и сервисов в ИКС;
- администратора сетей: UNIX и Windows;
- разработчика клиентских и клиент-серверных приложений;
- разработчика Web – приложений;
- специалиста по информационному менеджменту;
- менеджера проектов планирования развития ИКС.

#### **Трудоустройство выпускников:**

1. ОАО «Петербургская телефонная сеть»;
2. АО «ЛЕНГИПРОТРАНС»;
3. Акционерный коммерческий Сберегательный банк Российской Федерации;
4. ОАО «РИВЦ-Пулково»;
5. СПб ГУП «Петербургский метрополитен»;
6. ООО «СоюзБалтКомплект»;
7. ООО «ОТИС Лифт»;
8. ОАО «Новые Информационные Технологии в Авиации»;
9. ООО «Т-Системс СиАйЭс» и др.

**Кафедра** сегодня имеет в своем составе высококвалифицированный преподавательский состав, в том числе:

- 5 кандидатов технических наук, имеющих ученые звания профессора и доцента;
- 4 старших преподавателя;

- 6 штатных совместителей, в том числе кандидатов наук, профессиональных IT - специалистов;
- 15 Сертифицированных тренеров, имеющих Западные Сертификаты фирм: Microsoft, Oracle, Cisco, Novell.

Современная техническая база; лицензионное программное обеспечение; специализированные лаборатории, оснащенные необходимым оборудованием и ПО; качественная методическая поддержка образовательных программ; широкие Партнерские связи существенно влияют на конкурентные преимущества подготовки специалистов.

Авторитет специализаций кафедры в области компьютерных технологий подтверждается Сертификатами на право проведения обучения по методикам ведущих Западных фирм - поставщиков аппаратного и программного обеспечения.

Заслуженной популярностью пользуются специализации кафедры ПС по подготовке и переподготовке профессиональных компьютерных специалистов с выдачей **Государственного Диплома** о профессиональной переподготовке по направлениям: **"Информационные технологии (инженер-программист)"** и **"Системный инженер"**, а также Диплома о дополнительном (к высшему) образовании с присвоением квалификации: **"Разработчик профессионально-ориентированных компьютерных технологий "**. В рамках этих специализаций высокопрофессиональные преподаватели готовят компетентных компьютерных специалистов по современным в России и за рубежом операционным системам, базам данных и языкам программирования ведущих фирм: Microsoft, Cisco, IBM, Intel, Oracle, Novell и др.

Профессионализм, компетентность, опыт, и качество программ подготовки и переподготовки IT- специалистов на кафедре ПС неоднократно были удостоены **высокими наградами «Компьютерная Элита» в номинации лучший учебный центр России.**

#### **Партнеры:**

1. **Microsoft** Certified Learning Solutions;
2. **Novell** Authorized Education Center;
3. **Cisco** Networking Academy;
4. **Oracle** Academy;
5. **Sun Java** Academy и др;
6. **Prometric**;
7. **VUE**.

**Мы готовим квалифицированных инженеров в области инфокоммуникационных технологий с новыми знаниями, образом мышления и способностями быстрой адаптации к современным условиям труда.**

Татьяна Викторовна Зудилова  
Сергей Евгеньевич Иванов  
Сергей Эдуардович Хоружников

**SQL и PL/SQL для разработчиков СУБД ORACLE**  
**ПРАКТИКУМ**

В авторской редакции  
Редакционно-издательский отдел НИУ ИТМО  
Зав. РИО  
Лицензия ИД № 00408 от 05.11.99  
Подписано к печати  
Заказ №  
Тираж 100 экз.  
Отпечатано на ризографе

Н.Ф. Гусарова



**Редакционно-издательский отдел**  
Санкт-Петербургского национального  
исследовательского университета  
информационных технологий, механики

и оптики  
197101, Санкт-Петербург, Кронверкский пр., 49

