

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ярославский государственный университет им. П.Г. Демидова

Кафедра теоретической информатики

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

« 24 » _____ мая _____ 2022 г.

Рабочая программа дисциплины
«Система управления базами данных Oracle»

Направление подготовки
02.03.02 Фундаментальная информатика и информационные технологии

Профиль
«Информатика и компьютерные науки»

Квалификация выпускника
Бакалавр

Форма обучения
очная

Программа рассмотрена
на заседании кафедры
от 15 марта 2022 г.,
протокол № 8

Программа одобрена НМК
факультета ИВТ
протокол № 6 от
18 апреля 2022 г.

Ярославль

1. Цели освоения дисциплины

Целями дисциплины «Система управления базами данных Oracle» являются освоение основ проектирования и разработки реляционных баз данных в СУБД Oracle. Данный курс вырабатывает у студентов умение применять аппарат реляционных баз данных и языки SQL и PL/SQL для эффективного решения прикладных задач.

2. Место дисциплины в структуре ОП бакалавриата

Дисциплина «Система управления базами данных Oracle» относится к вариативной части (дисциплина по выбору) ОП бакалавриата.

Для освоения данной дисциплиной студенты должны обладать знаниями по дискретной математике, математической логике и основам программирования. В процессе обучения требуется проявлять организованность, настойчивость в достижении цели, целеустремленность и инициативу.

Полученные в рамках дисциплины «Информатика и программирование» знания необходимы для развития алгоритмического мышления, развития навыков разработки баз данных и построения информационных систем, изучения профильных курсов по программированию.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП бакалавриата

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Профессиональные компетенции		
ПК-2 Способен к разработке требований и проектированию программного обеспечения	ПК-2.2 Использует современные языки и инструменты программирования для решения задач профессиональной деятельности	Знать: <ul style="list-style-type: none">- основы планирования;- особенности системы управления базами данных Oracle и ее месте среди других реляционных технологий создания баз данных;- основы архитектуры системы управления базами данных Oracle;- объекты базы данных Oracle;- язык SQL для системы Oracle;- язык PL/SQL. Уметь: <ul style="list-style-type: none">- строить план работ и следовать ему;- проектировать схему реляционной базы данных в 3-й нормальной форме;- разрабатывать SQL-запросы для решения

		<p>широкого круга задач;</p> <ul style="list-style-type: none"> - использовать представления; - разрабатывать пакеты процедур и функций для работы с базой данных; - использовать курсоры. <p>Владеть навыками:</p> <ul style="list-style-type: none"> - самостоятельного выполнения программных проектов; - использования программных средств проектирования и разработки баз данных от компании Oracle; - создания и отладки SQL-запросов; - создания и отладки подпрограмм на языке PL/SQL.
--	--	--

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 2 зач. ед., 72 акад. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Се м е ст р	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости
			Контактная работа						Форма промежуточной аттестации (по семестрам)
			лек ции	пра кти чес кие	лаб ора тор ны е	кон сул ьта ции	ат те ст ац ио нн ые ис пы та ния	сам осто ятел ьна я раб ота	
1.	Основы архитектуры системы управления базами данных Oracle	8	2						
2.	Язык SQL Oracle	8	8		8			8	Задания для самостоятельной работы.

								Задания для домашней работы
3.	Язык PL/SQL	8	6		6		10	Контрольная работа. Задания для самостоятельной работы. Задания для домашней работы
4.	Разработка базы данных	8	0		2		18	Задания для самостоятельной работы. Задания для домашней работы
					4			Зачет
	Всего за 8 семестр		16		16	4	36	
	Всего		16		16	4	36	

Содержание разделов дисциплины:

1. Основы архитектуры системы управления базами данных Oracle.

1.1. Введение. Основы теории реляционных баз данных. Классификации баз данных. Архитектуры обработки данных.

1.2. Архитектура СУБД Oracle. Процессы. Структура памяти. Файлы.

1.3. Конфигурирование.

1.4. Информация о результатах операции.

1.5. Поддержка мультиязычности в СУБД Oracle.

2. Язык SQL Oracle.

2.1. DDL. Типы данных. Создание и удаление таблиц. Средства определения и уничтожения представлений. Ограничения целостности. Последовательности.

2.2. DML. Основные операторы. Разработка запросов.

3. Язык PL/SQL.

3.1. Структура программы. Лексические основы. Выражения и операции. Управляющие структуры.

3.2. Записи.

3.3. Динамический SQL.

3.4. Курсоры.

3.5. Процедуры, функции, пакеты.

3.6. Исключительные ситуации.

3.7. Сборные конструкции.

3.8. Триггеры.

4. Разработка базы данных.

4.1. Выполнение практических работ.

4.2. Выполнение практических работ из программы Oracle Academy Advanced Computer Science: Oracle Database SQL Fundamentals.

4.3. Выполнение и проверка самостоятельных домашних работ.

5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция (или лекция общего курса) – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Требования к академической лекции: современный научный уровень и насыщенная информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами. Основной целью является формирование у студентов понимания теоретического материала, изложенного на лекциях, и освоение необходимых умений и навыков, через решение практических заданий. Здесь преподавание строится на сочетании коллективной работы группы с самостоятельной индивидуальной работой студентов. Задание выдается группе для решения. В некотором порядке студенты вызываются для решения и дальнейшего обсуждения решения всей группой. Для демонстрации результатов используется проектор. Результат выполнения фиксируется и учитывается в дальнейшем при сдаче зачета.

Консультации – групповые занятия, являющиеся одной из форм контроля самостоятельной работы студентов. На консультациях по просьбе студентов рассматриваются наиболее сложные моменты в решении задач, которые возникают у них в процессе самостоятельной работы, обсуждаются результаты решения заданий, выполненных студентами самостоятельно.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости)

В процессе осуществления образовательного процесса используются:

- для формирования текстов материалов для промежуточной и текущей аттестации – программа Microsoft Word;
- для наглядной демонстрации теоретических и практических материалов – презентации для программ Microsoft Power Point и Adobe Acrobat Reader;
- для выполнения практических работ – Oracle Database 11g XE;
- для взаимодействия с СУБД – Oracle SQL Developer;
- для построения концептуальных моделей и схем баз данных – Oracle SQL Developer Data Modeler.

7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

а) основная:

1. Пржиялковский, В. В., Введение в Oracle SQL : учеб. пособие для вузов / В. В. Пржиялковский, М., БИНОМ. Лаборатория знаний, 2012, 319с
2. Смирнов, А. В., Основы Oracle PL/SQL : учебно-методическое пособие / А. В. Смирнов ; Яросл. гос. ун-т, Ярославль, ЯрГУ, 2020, 98с
3. Смирнов, А. В., Основы Oracle PL/SQL [Электронный ресурс] : учебно-методическое пособие / А. В. Смирнов ; Яросл. гос. ун-т, Ярославль, ЯрГУ, 2020, 98с

б) дополнительная:

1. Смирнов С. Н., Задворьев И.С. Работаем с Oracle. М.: Гелиос АРВ, 2002. - 495 с.
2. Смирнов, С. Н., Обработка документов средствами Oracle : практикум по XML и JDBC : учеб. пособие для вузов / С. Н. Смирнов, М., Гелиос АРВ, 2004, 187с

в) ресурсы сети «Интернет»

1. <http://www.oracle.com>. Официальная документация Oracle.

8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

- учебные аудитории для проведения занятий лекционного типа;
- учебные аудитории для проведения лабораторных работ;
- учебные аудитории для проведения групповых и индивидуальных консультаций;
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы;
- помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, хранящиеся на электронных носителях и обеспечивающие тематические иллюстрации, соответствующие рабочей программе дисциплины.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для лабораторных работ – списочному составу группы обучающихся.

Автор:

Старший преподаватель кафедры
теоретической информатики, к.ф.-м.н. _____ О. Е. Горбунов

**Приложение №1 к рабочей программе дисциплины
«Система управления базами данных Oracle»
Фонд оценочных средств
для проведения текущей и промежуточной аттестации студентов
по дисциплине**

1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

1.1. Контрольные задания и иные материалы, используемые в процессе текущей аттестации

Задания для самостоятельной работы

Задания по теме 2 «Язык SQL Oracle».

Задания основаны на материалах программы Oracle Academy Advanced Computer Science “Oracle SQL Fundamentals”. Пример заданий по теме «Агрегирующие функции» (используется база данных HR):

1. Вывести максимальный, минимальный оклады, сумму всех окладов сотрудников и средний оклад сотрудников. Столбцы назвать Maximum, Minimum, Sum и Average. Округлить результаты до ближайшего целого.

Решение.

```
SELECT ROUND(MAX(salary),0) AS "Maximum",  
       ROUND(MIN(salary),0) AS "Minimum",  
       ROUND(SUM(salary),0) AS "Sum",  
       ROUND(AVG(salary),0) AS "Average"  
FROM employees;
```

2. Аналогично 1, но вывести дополнительно поле job_id и результаты по каждому коду работы.

Решение.

```
SELECT job_id, ROUND(MAX(salary),0) "Maximum",  
       ROUND(MIN(salary),0) "Minimum",  
       ROUND(SUM(salary),0) "Sum",  
       ROUND(AVG(salary),0) "Average"  
FROM employees  
GROUP BY job_id;
```

3. Вывести job_id и количество сотрудников с такой работой. Модифицировать запрос, чтобы результат был для job_id по запросу (например, IT_PROG).

Решение.

```
SELECT job_id, COUNT(*)  
FROM employees  
GROUP BY job_id;
```

```
SELECT job_id, COUNT(*)  
FROM employees  
WHERE job_id = '&job_title'  
GROUP BY job_id;
```

4. Вывести общее количество менеджеров без их перечисления.

Решение.

```
SELECT COUNT(DISTINCT manager_id) AS "Number of Managers"
```


FROM employees;

5. Вывести разницу между максимальным и минимальным окладами.

Решение.

```
SELECT MAX(salary) - MIN(salary) AS "Difference"
```

```
FROM employees;
```

6. Вывести ID менеджера, оклад самого низкооплачиваемого сотрудника этого менеджера. Исключить сотрудников с неопределенным менеджером. Исключить группы с минимальным окладом 6000 или менее. Сортировать по убыванию минимального оклада.

Решение.

```
SELECT manager_id, MIN(salary)
```

```
FROM employees
```

```
WHERE manager_id IS NOT NULL
```

```
GROUP BY manager_id
```

```
HAVING MIN(salary) > 6000
```

```
ORDER BY MIN(salary) DESC;
```

7. Вывести строку с: общим количеством сотрудников, а также с детализацией количества сотрудников, работающих с 1995, 1996, 1997 и 1998 гг. Обозначить столбцы соответственно.

Решение.

```
SELECT COUNT(*) total,
```

```
      SUM(DECODE(TO_CHAR(hire_date, 'YYYY'), 1995, 1, 0)) AS "1995",
```

```
      SUM(DECODE(TO_CHAR(hire_date, 'YYYY'), 1996, 1, 0)) AS "1996",
```

```
      SUM(DECODE(TO_CHAR(hire_date, 'YYYY'), 1997, 1, 0)) AS "1997",
```

```
      SUM(DECODE(TO_CHAR(hire_date, 'YYYY'), 1998, 1, 0)) AS "1998"
```

```
FROM employees;
```

8. Вывести job_id и сумму окладов для этой работы с детализацией по отделам: 20 50 80 90 и всего.

Обозначить столбцы Dept 20, Dept 50, Dept 80, Dept 90, Total.

Решение.

```
SELECT job_id AS "Job",
```

```
      SUM(DECODE(department_id, 20, salary)) AS "Dept 20",
```

```
      SUM(DECODE(department_id, 50, salary)) AS "Dept 50",
```

```
      SUM(DECODE(department_id, 80, salary)) AS "Dept 80",
```

```
      SUM(DECODE(department_id, 90, salary)) AS "Dept 90",
```

```
      SUM(salary) AS "Total"
```

```
FROM employees
```

```
GROUP BY job_id;
```

Задания по теме 3 «Язык PL/SQL».

Примерные практические задания (основаны на задаче, которая разбирается в методических указаниях по лабораторному практикуму Рублева В.С. (№2 списка основной литературы в разделе 7 Программы)). Необходимо разработать пакет для работы с базой данных.

1. Процедуру добавления нового порта. Использовать последовательность для идентификатора порта.

Решение.

```
PROCEDURE AddPort(v_portName varchar2) IS
```

```
BEGIN
```

```
  INSERT INTO Port(ID, Name)
```

```
  VALUES(SQ_Port.nextval, v_portName);
```

```
END;
```

2. Функцию получения идентификатора порта по имени. В имени могут присутствовать подстановочные символы.

Решение.

```
FUNCTION GetPortIDByName(v_portName varchar2) RETURN number IS
    v_result number;
BEGIN
    --Команда вывода сообщения в консоль
    DBMS_OUTPUT.PUT_LINE('v_portName = ' || v_portName);
    v_result := -1;
    SELECT ID
    INTO v_result
    FROM Port
    WHERE Name LIKE v_portName;
    RETURN (v_result);
END;
```

3. Процедуру, обновляющую данные о корабле по заданному имени корабля.

Решение.

```
PROCEDURE UpdateShip(v_Name varchar2, v_newName varchar2, v_newTonnage
number) IS
BEGIN
    UPDATE Ship
    SET
        Name = v_newName,
        Tonnage = v_newTonnage
    WHERE Name LIKE v_Name;
END;
```

4. При реализации атрибута STATUS в виде столбца таблицы требуется ввести три правила:

4.1. Ограничение для столбца должно быть задано так, чтобы он содержал лишь значения, указанные как допустимые статусы (реализуется в виде ограничения CHECK на столбце STATUS).

Решение.

```
ALTER TABLE SHIP
ADD CONSTRAINT "Check_Ship_State"
CHECK(State IN (1,2,3,4));
```

4.2. Исходное значение столбца в новой строке должно быть допустимым исходным значением (при помощи запускающегося перед вставкой триггера уровня строки в таблице либо при помощи значения по умолчанию).

Решение.

Результаты решения задания обсуждаются на консультациях по просьбе студентов.

4.3. При обновлении столбца необходимо обеспечить допустимое изменение состояния (при помощи запускающегося перед обновлением триггера уровня строки).

Решение.

Результаты решения задания обсуждаются на консультациях по просьбе студентов.

Задания по теме 4 «Разработка базы данных».

Примерные практические задания (основаны на задаче, которая разбирается в методических указаниях по лабораторному практикуму Рублева В.С. (№2 списка основной литературы в разделе 7 Программы)).

1. Сколько кораблей находится в порту (статусы 1-3)?

Решение.

Результаты решения задания обсуждаются на консультациях по просьбе студентов.

2. Сколько товара А находится на складах порта перегрузки?
(А можно задать как параметр или закодировать в запросе).
(2 способа: через вложенный запрос и через объединение таблиц).

Решение.

Результаты решения задания обсуждаются на консультациях по просьбе студентов.

3. Сколько товара А находится на кораблях порта перегрузки?
(А можно задать как параметр или закодировать в запросе).
(2 способа: через вложенный запрос и через объединение таблиц).

Решение (через объединение таблиц).

```
SELECT SUM(Quantity) FROM Cargoes, Ship_Cargoes
JOIN Ships ON (Ships.Id = Ship_Cargoes.Id_Ship)
JOIN Cargoes ON (Cargoes.Id = Ship_Cargoes.Id_Cargo)
WHERE (Ships.State between 1 AND 3) AND
      (Cargoes.Name = &Var);
```

4. Сколько товара А (на кораблях и складах порта перегрузки) направляется в порт В?

Решение.

```
SELECT (SELECT SUM(Ship_Cargoes.Quantity)
        FROM Ship_Cargoes
        JOIN Ships
        ON (Ship_Cargoes.Id_Ship = Ships.Id)
        JOIN Ports
        ON (Ship_Cargoes.Id_Port = Ports.Id)
        JOIN Cargoes
        ON (Ship_Cargoes.Id_Cargo = Cargoes.Id)
        WHERE Cargoes.Name = &Cargo AND
              Ports.Name = &Port AND
              Ships.State BETWEEN 1 AND 3) +
        (SELECT SUM(Port_Cargoes.Quantity)
        FROM Port_Cargoes
        JOIN Ports
        ON (Port_Cargoes.Id_Port = Ports.ID)
        JOIN Cargoes
        ON (Port_Cargoes.Id_Cargo = Cargoes.Id)
        WHERE Cargoes.Name = &Cargo AND
              Ports.Name = &Port AND)
```

FROM Dual;

5. Сколько товара А будет выгружено на склады порта?

Решение.

```
SELECT SUM(Ship_Cargoes.Quantity)
FROM Ship_Cargoes
JOIN Ships
ON Ship_Cargoes.Id_Ship = Ships.Id
JOIN Cargoes
ON Ship_Cargoes.Id = Cargoes.Id
WHERE (Cargoes.Name = '&Cargo') AND
      (Ships.State BETWEEN 1 AND 2) AND
      (Ship_Cargoes.Id_Port_Dest = Ships.Id_Port_Dest);
```

Типовые задания для домашней работы

Домашние индивидуальные задания приведены в методических указаниях по лабораторному практикуму Рублева В.С. (№2 списка дополнительной литературы в разделе 7 Программы).

Общее задание:

1. Разработать схему реляционной базы данных (в 3 нормальной форме) заданной предметной области и пакет процедур и функций работы с этой базой данных, при помощи которого можно динамически изменять и показывать информацию по любому объекту. Предусмотреть реакцию на недопустимые данные. В пакете процедур и функций реализовать возможность поиска по различным полям.

2. Создать соответствующую б.д. в СУБД Oracle.

3. Разработать пользовательский интерфейс в любой среде разработки, позволяющий протестировать работу базы данных. Выбор среды разработки не ограничен.

4. Разработать набор тестов (входные данные и ожидаемые результаты).

5. Составить отчет, содержащий:

- постановку задачи (общего и индивидуального задания);
- логическую схему (ER-диаграмму);
- схему реляционной БД;
- SQL написать запросы для создания всех необходимых таблиц;
- пакет функций и процедур с подробными комментариями к каждой процедуре и функции в спецификации пакета (назначение процедуры или функции, описание её параметров и возвращаемого функцией значения).

Пример решения задания приведен в методических указаниях по лабораторному практикуму Рублева В.С. (№2 списка основной литературы в разделе 7 Программы). Решения обсуждаются на консультациях по просьбе студентов.

Типовой вариант контрольной работы

На контрольной работе студентам предлагается следующие типовые задания.

Имеется следующая предметная область:

Товары (item) характеризуются кодом (натуральное число, не может быть пустым), артикулом (строка из 20 символов, не может быть пустой), названием (строка из максимум 50 символов, не может быть пустой) и товарной категорией. Код товара однозначно определяет товар. Товарные категории (item category) определяют классификацию товаров по некоторому признаку. Товарная категория имеет название (строка из максимум 20 символов, не может быть пустой) и может включать несколько товаров (название каждой товарной категории уникально). Склады (warehouse) определяются кодом (натуральное число, не может быть пустым), названием (строка из максимум 20 символов, не может быть пустой) и адресом (строка из максимум 80 символов). Код склада однозначно определяет склад. Некоторое количество товара может находиться на складе или на нескольких складах. С другой стороны, на одном складе могут храниться различные товары. Количество товара на складе не может быть отрицательным.

1. Нарисовать ER-диаграмму, схему таблиц и SQL-операторы создания таблиц для указанной выше предметной области. Предусмотреть поддержку ограничений целостности: Not null, Unique, Primary key, Foreign key, Check, каскадное удаление в случае удаления записей, соответствующих Товару, Складу и Товарной категории.

Решение.

ER-диаграмма:

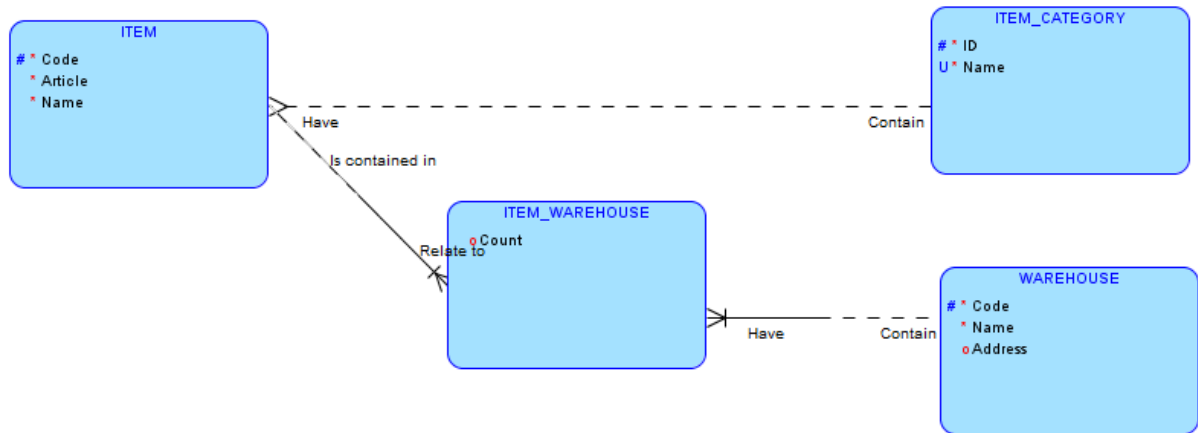
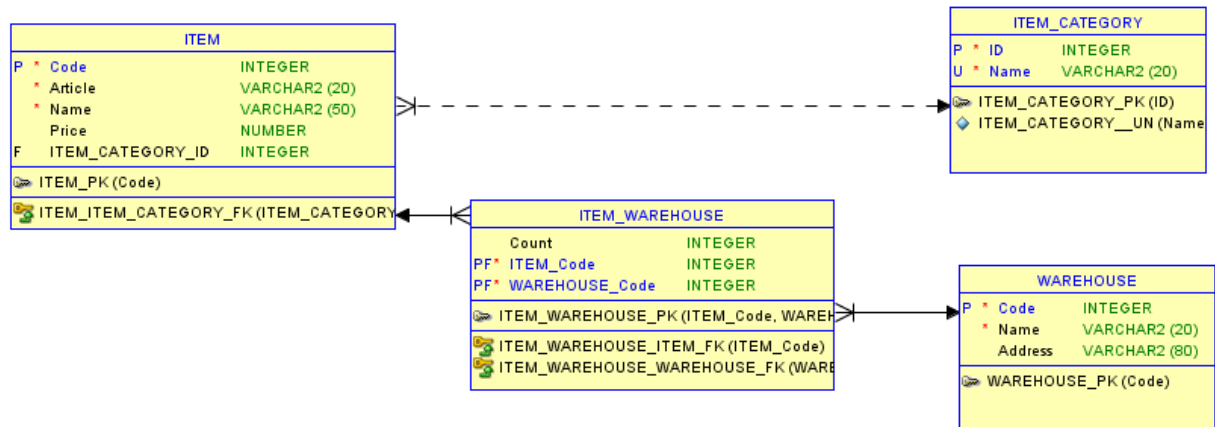


Схема таблиц:



SQL-операторы:

```
CREATE TABLE Item(
  Code integer NOT NULL ENABLE,
  Name varchar2(50) NOT NULL ENABLE,
  Article varchar2(20) NOT NULL ENABLE,
  Item_category_id Integer,
  CONSTRAINT "pk_item" PRIMARY KEY(Code) ENABLE,
  CONSTRAINT "fk_item_category" FOREIGN KEY(Item_category_code) REFERENCES
  Item_category(Id));
```

```
CREATE TABLE Item_category(
  Id Integer,
  Name varchar2(20) NOT NULL ENABLE,
  CONSTRAINT "pk_item_category" PRIMARY KEY(Id) ENABLE,
  CONSTRAINT "u_item_category" UNIQUE(Name) ENABLE
);
```

```
CREATE TABLE Warehouse(
  Code Integer NOT NULL ENABLE,
  Name varchar2(20) NOT NULL ENABLE,
  Address varchar2(80),
```

```

CONSTRAINT "pk_warehouse" PRIMARY KEY(Code) ENABLE
);

CREATE TABLE Item_warehouse(
    Item_code Integer NOT NULL ENABLE,
    Warehouse_code Integer NOT NULL ENABLE,
    Count Integer,
    CONSTRAINT "pk_item_warehouse" PRIMARY KEY("Item_code", "Warehouse_code")
ENABLE,
    CONSTRAINT "fk_Item_warehouse_Item_code" FOREIGN KEY("Item_code")
REFERENCES Item(Code) ON DELETE CASCADE ENABLE,
    CONSTRAINT "fk_Item_warehouse_Warehouse_code" FOREIGN
KEY("Warehouse_code") REFERENCES Warehouse(Code) ON DELETE CASCADE
ENABLE,
    CONSTRAINT "ch_Item_warehouse_Quantity" CHECK(Quantity>=0) DEFERRABLE
INITIALLY DEFERRED ENABLE
);

```

2. Для созданных выше таблиц написать SQL-выражения для следующих задач:

2.1. Добавить новый товар 2579 с артикулом AAJ235679M123879KLOY; название = Скрепка канцелярская; товарная группа не определена. Этот товар имеется на складе 1 в количестве 500 штук и на складе 2 в количестве 300 штук.

Решение.

```

INSERT INTO Item(Code, Name, Article, Item_category_id)
VALUES (2579, 'Скрепка канцелярская', 'AAJ235679M123879KLOY', NULL);

```

2.2. Удалить все товары, которых нет ни на одном из складов.

Решение.

```

DELETE FROM Item
WHERE Item.Code NOT IN (SELECT Item_code
                        FROM Item_warehouse
                        WHERE Count>0);

```

2.3. Изменить товарную категорию у товаров, у которых название товарной категории = 'SPEC_1', на товарную категорию с названием 'SPEC_2'.

Решение.

```

UPDATE Item_category
SET Name = 'SPEC_2'
WHERE Name = 'SPEC_1';

```

3. Создать представление, содержащее следующие поля: код товара, артикул товара, название товара, название товарной категории товара, суммарное количество товара на всех складах.

Решение.

```

CREATE OR REPLACE VIEW ItemsQty AS
SELECT Item.Code, Item.Article, Item.Name ItemName,
(SELECT Item_category.Name FROM Item_category
WHERE Item.Item_category_id=Item_category.Id) ItemCategoryName,
(SELECT SUM(Item_warehouse.Count) FROM Item_warehouse
WHERE (Item.Code = Item_warehouse.Item_code)) ItemCount
FROM Item;

```

Список заданий к зачету

Зачет выставляется по результатам тестового задания и краткого собеседования со студентом после его проверки. Тестовое задание аналогично по своей структуре заданиям из контрольной работы. Собеседование может включать теоретические вопросы из списка, приведенного ниже.

Список теоретических вопросов для зачета.

1. Архитектуры обработки данных.
2. Архитектура СУБД Oracle. Архитектура сервера Oracle.
3. Основные объекты СУБД Oracle.
4. Особенности управления одновременным доступом.
5. DML. Обзор. В частности, вставка строк, модификация строк, удаление.
6. DDL. Обзор. В частности, создание и удаление таблиц.
7. Работа с представлениями. Последовательности.
8. Ограничения целостности.
9. Управляющие структуры PL/SQL.
10. SQL в PL/SQL. Динамический SQL.
11. Курсоры.
12. Процедуры и функции.
13. Пакеты.
14. Обработка исключительных ситуаций.
15. Триггеры базы данных.
16. Средства обеспечения целостности данных. Управление транзакциями.

2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания

2.1. Шкала оценивания сформированности компетенций и ее описание

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

Пороговый уровень - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

Продвинутый уровень - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

Высокий уровень - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Код компетенции	Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
				Пороговый уровень	Продвинутый уровень	Высокий уровень
Профессиональные компетенции						
ПК-2	Задания для самостоятельной работы по темам 2, 3, 4. Задания для домашней работы. Контрольная работа. Заче	1	Знать: <ul style="list-style-type: none">- особенности системы управления базами данных Oracle и ее месте среди других реляционных технологий создания баз данных;- основы архитектуры системы управления базами данных Oracle;- объекты базы данных Oracle;- язык SQL для системы Oracle;- язык PL/SQL. Уметь: <ul style="list-style-type: none">- проектировать	<ul style="list-style-type: none">- особенности системы управления базами данных Oracle и ее месте среди других реляционных технологий создания баз данных;- основы архитектуры системы управления базами данных Oracle;- объекты базы данных Oracle;- язык SQL для системы Oracle;- язык PL/SQL.	<ul style="list-style-type: none">- особенности системы управления базами данных Oracle и ее месте среди других реляционных технологий создания баз данных;- основы архитектуры системы управления базами данных Oracle;- объекты базы данных Oracle;- язык SQL для системы Oracle;- язык PL/SQL;- проектирование схемы реляционной базы данных в 3-й нормальной форме;- разработка SQL запросов для решения широкого круга задач;- использование представлений;- разработка пакетов	<ul style="list-style-type: none">- особенности системы управления базами данных Oracle и ее месте среди других реляционных технологий создания баз данных;- основы архитектуры системы управления базами данных Oracle;- объекты базы данных Oracle;- язык SQL для системы Oracle;- язык PL/SQL;- проектирование схемы реляционной базы данных в 3-й нормальной форме;- разработка SQL запросов для решения широкого круга задач;- использование представлений;- разработка пакетов

	т.	<p>схему реляционной базы данных в 3-й нормальной форме;</p> <ul style="list-style-type: none"> - разрабатывать SQL-запросы для решения широкого круга задач; - использовать представления; - разрабатывать пакеты процедур и функций для работы с базой данных; - использовать курсоры. <p>Владеть навыками:</p> <ul style="list-style-type: none"> — использования программных средств проектирования и разработки баз данных от компании Oracle; - создания и отладки SQL-запросов; - создания и отладки подпрограмм на языке PL/SQL. 		<ul style="list-style-type: none"> - использование представлений; - разработка пакетов процедур и функций для работы с базой данных; - использование курсоров. 	<p>процедур и функций для работы с базой данных;</p> <ul style="list-style-type: none"> - использование курсоров; — использование программных средств проектирования и разработки баз данных от компании Oracle; - создание и отладка сложных SQL-запросов; - создание и отладка подпрограмм на языке PL/SQL; - понимание механизмов блокирования и управления одновременным доступом в СУБД Oracle.
--	----	--	--	---	---

3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объеме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;
- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;

- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Шкала оценивания успеваемости. Оценивание результатов обучения студентов по дисциплине «Система управления базами данных Oracle» осуществляется по регламенту текущего контроля и промежуточной аттестации.

Текущий контроль в семестре проводится с целью обеспечения своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы студентов.

Текущий контроль проводится в виде контрольных работ, заданий для самостоятельной работы и заданий для домашней работы. Критериями оценивания степени овладения умениями и навыками, полученными в результате освоения данной дисциплины, являются следующие критерии.

Контрольная работа.

Показатели	Критерии
Понимание условия задачи	-Хорошее оформление работы, четкие рисунки и чертежи.
План решения задачи	-Рациональный способ решения.
Осуществление решения	-Решение задачи в общем виде. - Правильность решений.
Проверка правильности решения задачи	-Краткое объяснение решения. -Анализ полученных результатов.

Шкала оценивания решения задачи:

0 баллов – полное отсутствие решения; 0,5 балла – частичное выполнение критерия; 0,8 балла – полное выполнение критерия с незначительными ошибками, 1 балл – полное выполнение критерия.

Суммируются баллы по всей контрольной работе. За контрольную выставляется оценка в зависимости от количества полученных баллов:

менее 50% от максимально возможного количества баллов - неудовлетворительно,
50-75% от максимально возможного количества баллов - удовлетворительно,
76-85% от максимально возможного количества баллов - хорошо,
86-100% от максимально возможного количества баллов – отлично.

Задания для самостоятельной работы.

Оценка	Критерии
Отлично	Код построен концептуально правильно. В коде отсутствуют синтаксические ошибки. Код корректно работает на любых входных данных. Студент легко ориентируется в коде работы, вносит поправки. Студент применяет и объясняет изучаемые

	<p>элементы языка программирования.</p> <p>Используются эффективные способы решения.</p> <p>Изучаемые технологии применяются в полной мере.</p>
Хорошо	<p>Код построен концептуально правильно.</p> <p>В коде отсутствуют синтаксические ошибки.</p> <p>Код корректно работает на большинстве входных данных. Возможно наличие входных данных, которые код обрабатывает некорректно. В этом случае студент корректирует свой код.</p> <p>Студент легко ориентируется в коде работы, вносит поправки, возможно не для всех заданных вопросов.</p> <p>Студент применяет и объясняет основные изучаемые элементы языка программирования, возможно неполное понимание всех нюансов применяемых средств программирования.</p> <p>Используются неэффективные методы решения</p> <p>Изучаемые технологии применяются не в полной мере.</p>
Удовлетворительно	<p>Код построен концептуально правильно.</p> <p>В коде могут присутствовать незначительные синтаксические ошибки.</p> <p>В коде отсутствуют синтаксические ошибки.</p> <p>Код корректно работает не на всех входных данных. Возможно наличие входных данных, которые код обрабатывает некорректно. Доработка программного кода не всегда приводит к желаемому результату.</p> <p>Студент ориентируется в коде работы, но затрудняется внести поправки в случае достаточно сложных вопросов</p> <p>Студент применяет, но не объясняет основные изучаемые элементы языка программирования.</p> <p>Используются неэффективные методы решения.</p> <p>Изучаемые технологии применяются не в полной мере.</p>
Неудовлетворительно	<p>Задача решена неверно или решена совсем другая задача.</p> <p>Программа не выполняется или выполняется неверно для большинства тестов.</p> <p>Отсутствует понимание кода программы и применяемых средств программирования.</p> <p>Код является дубликатом уже существующего решения.</p> <p>Студент не может написать код для решения простых задач.</p>

Задания для домашней работы.

Оценка	Критерии
--------	----------

Отлично	<p>Подготовлен отчет.</p> <p>Правильно построен алгоритм решения задачи.</p> <p>Правильно построено архитектурное решение задачи</p> <p>Программа реализована и выполняется в соответствии с поставленной задачей.</p> <p>Разработаны и выполняются полные наборы тестов для проверки работы программы.</p> <p>Программа правильно выполняется для всех тестов.</p> <p>Студент легко ориентируется в коде работы, вносит поправки.</p> <p>Студент применяет и объясняет изучаемые элементы языка программирования.</p> <p>Используются эффективные способы решения.</p> <p>Изучаемые технологии применяются в полной мере.</p>
Хорошо	<p>Подготовлен отчет.</p> <p>Правильно построен алгоритм решения задачи</p> <p>Правильно построено архитектурное решение задачи.</p> <p>Программа реализована и выполняется в соответствии с поставленной задачей</p> <p>Разработаны и выполняются наборы тестов для проверки работы программы.</p> <p>Программа правильно выполняется для всех тестов, возможно после доработки программного кода.</p> <p>Студент ориентируется в коде работы, вносит поправки, возможно не для всех заданных вопросов.</p> <p>Студент применяет и объясняет основные изучаемые элементы языка программирования, возможно неполное понимание всех нюансов применяемых средств программирования.</p> <p>Используются неэффективные методы решения.</p>
Удовлетворительно	<p>Алгоритм решения задачи построен правильно за исключением отдельных небольших частей решения.</p> <p>Архитектурное решение задачи в основном правильное</p> <p>Программа реализована и выполняется.</p> <p>Разработаны и выполняются незначительные наборы тестов для проверки работы программы</p> <p>Программа правильно выполняется не для всех тестов, доработка программного кода не всегда приводит к желаемому результату</p> <p>Студент ориентируется в коде работы, но затрудняется внести поправки в случае достаточно сложных вопросов</p> <p>Студент применяет, но не объясняет основные изучаемые элементы языка программирования.</p> <p>Используются неэффективные методы решения.</p>
Неудовлетворительно	<p>Задача решена неверно или решена совсем другая задача.</p>

	<p>Программа не выполняется или выполняется неверно для большинства тестов.</p> <p>Отсутствует понимание кода программы и применяемых средств программирования</p> <p>Код является дубликатом уже существующего решения.</p> <p>Студент не может написать код для решения простых задач.</p>
--	--

3.2 Описание процедуры выставления оценки

Шкала оценивания успеваемости текущего контроля и промежуточной аттестации.

В зависимости от уровня сформированности компетенции по окончании освоения дисциплины студенту выставляется оценка «зачтено» или «не зачтено»

Студент получает один вопрос и одно практическое задание.

Результатом оценивания ответов на вопросы является сумма баллов, которая складывается с индивидуальным семестровым рейтингом каждого студента и определяет оценку:

более 50% от максимального количества баллов – «зачтено»;

менее 50% от максимального количества баллов – «не зачтено».

Оценка «зачтено» выставляется студенту, который:

- прочно усвоил предусмотренный программный материал;
- правильно, аргументировано ответил на все вопросы, с приведением примеров;
- показал глубокие систематизированные знания, владеет приемами рассуждения и сопоставляет материал из разных источников: теорию связывает с практикой, другими темами данного курса, других изучаемых предметов.

Дополнительным условием получения оценки «зачтено» могут стать хорошие успехи при выполнении самостоятельной и домашней работ, контрольной работы, систематическая активная работа на практических занятиях.

Оценка «не зачтено» Выставляется студенту, который не справился с 50% вопросов и заданий, в ответах на другие вопросы допустил существенные ошибки. Не может ответить на дополнительные вопросы, предложенные преподавателем. Целостного представления о взаимосвязях, компонентах дисциплины у студента нет.

Приложение №2 к рабочей программе дисциплины «Система управления базами данных Oracle»

Методические указания для студентов по освоению дисциплины

Основной формой изложения учебного материала по дисциплине «Система управления базами данных Oracle» являются лекции, проводимые в компьютерном классе с использованием презентаций. В ходе лекции делаются остановки, чтобы студенты могли освоить на практике некоторые аспекты излагаемого материала. Это связано с тем, что значительная часть курса связана с программированием, эффективное изучение которого достигается при сочетании теории и практики.

Для успешного освоения дисциплины важно решение большого количества практических прикладных задач. Часть лабораторных работ основана на материалах Oracle Academy Advanced Computer Science (ЯрГУ им. П.Г. Демидова является участником этой программы). Происходит постоянная интеграция материалов Oracle Academy в учебный процесс. Студентам рекомендуется дома установить свободно распространяемую СУБД Oracle Database 11g XE вместе с дополнительными инструментами (Oracle SQL Developer и Oracle SQL Developer Data Modeler). Это программное обеспечение доступно для скачивания на сайте Oracle (<http://oracle.com>) после регистрации. В установочный комплект входит и демонстрационная база данных HR, на которой построены задания курса. Другая часть лабораторных работ связана с определенной предметной областью, которую предлагается проанализировать, построить соответствующую концептуальную модель и разработать базу данных. По этой части лабораторных работ также предусмотрены задания, связанные с построением SQL запросов и разработки подпрограмм на языке PL/SQL.

Основной целью лабораторных работ является формирование у студентов понимания теоретического материала, изложенного на лекциях, через решение практических заданий, получение навыков работы с базами данных Oracle. Здесь преподавание строится на сочетании коллективной работы группы с самостоятельной индивидуальной работой студентов. Успеваемость студентов во время лабораторных работ фиксируется и учитывается на зачете.

Выполнение домашних индивидуальных работ имеет большое значение для освоения дисциплины, т.к. связано с самостоятельной работой над проектом и решением целого ряда задач. Задания выдаются на длительный период с обязательным предъявлением результатов. Как правило, это проектирование базы данных для некоторой предметной области с последующей ее реализацией в системе управления базами данных Oracle. Для всех студентов имеется одно общее задание и индивидуальные постановки частных задач (описания предметных областей). Зачет автоматом может быть поставлен лишь студенту, успешно выполнившему домашнюю индивидуальную работу. При возникновении любых вопросов как по технологической части, так и по содержанию индивидуальных заданий следует обращаться к преподавателю за консультацией. При сдаче этой работы необходимо уделять внимание общим требованиям, предъявляемым преподавателем для всех студентов. В частности, должен прилагаться отчет в бумажной форме о проделанной работе и полученных результатах. Программный продукт должен сопровождаться набором тестов для проверки функционала и соответствия работы предъявляемым требованиям.

Для проверки и контроля усвоения материала, приобретенных практических навыков работы с базой данных Oracle в течение обучения проводятся мероприятия текущей аттестации в виде контрольной работы. При написании работы можно пользоваться любыми материалами: важно понимание материала и оперативное решение поставленных задач.

В конце семестра изучения дисциплины студенты сдают зачет. По итогам работы за семестр (успеваемость во время лабораторных работ, результаты контрольной работы,

результаты выполнения домашней работы) некоторым студентам предлагается зачет автоматом. Остальным студентам предлагается список вопросов для подготовки. Перед зачетом предусмотрена групповая консультация. Некоторым студентам для получения зачета предлагается выполнить практическую задачу (из списка лабораторных работ либо некоторую часть домашнего задания).

Освоить самостоятельно вопросы, излагаемые в процессе изучения дисциплины «Система управления базами данных Oracle» сложно, но возможно. Особенно это касается студентов, имеющих опыт работы с базами данных Oracle. Поэтому посещение аудиторных занятий не является обязательным для получения зачета.

Тесты для самоподготовки

Вопрос 1. Какие три типа процессов выполняются в экземпляре сервера СУБД Oracle (выберите три)?

- 1. серверные;
- 2. фоновые;
- 3. подчиненные;
- клиентские;
- операционные;

Вопрос 2. В каком режиме работы сервера Oracle количество серверных процессов ограничено (имеется пул процессов)?

- 1. выделенного сервера;
- 2. разделяемого сервера (MTS);

Вопрос 3. Какие группы памяти существуют у сервера Oracle (выберите три)?

- 1. System Global Area (SGA);
- 2. Process Global Area (PGA);
- 3. User Global Area (UGA);
- 4. Thread Global Area (TGA);
- 5. Memory Pool;

Вопрос 4. В какой области памяти сервера Oracle находится буфер журнала повторного выполнения (выберите один)?

- 1. System Global Area (SGA);
- 2. Process Global Area (PGA);
- 3. User Global Area (UGA);
- 4. Thread Global Area (TGA);
- 5. Memory Pool;

Вопрос 5. Как называется наименьшая единица выделения пространства в Oracle (выберите один)?

- 1. сегмент;
- 2. блок;
- 3. экстенд;
- 4. раздел;

Вопрос 6. Какие три уровня настроек региональных настроек существует в сервере Oracle?

- 1. базы данных;
- 2. экземпляра;
- 3. сессии;
- 4. рабочего пространства;
- 5. сервиса базы данных;

Вопрос 7. При выполнении операций сервером Oracle возвращается некоторый результат в виде числа и текстового сообщения. Какие коды свидетельствуют об ошибочном результате выполнения операции?

- 1. отрицательные;

2. положительные;
3. код с нулевым значением;

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

В качестве учебно-методического обеспечения рекомендуется использовать литературу, указанную в разделе № 7 данной рабочей программы.

Для получения справочных руководств по базам данных, языкам SQL и PL/SQL также рекомендуется использовать сайт компании Oracle <http://www.oracle.com>. Материалы находятся в свободном доступе после регистрации.

Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет (http://lib.uniyar.ac.ru/opac/bk_login.php) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ (http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/паролю.

3. Электронная картотека «Книгообеспеченность» (http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека «Книгообеспеченность» доступна в сети университета и через Личный кабинет.

Для работы в аудиториях на лабораторных занятиях и дома необходимо следующее программное обеспечение Oracle:

1. СУБД Oracle Database 11g XE;
2. Oracle SQL Developer;
3. Oracle SQL Developer Data Modeler.

Это программное обеспечение доступно для скачивания на сайте Oracle (<http://www.oracle.com>) после регистрации.