

Федеральное агентство по образованию
ГОУ ВПО «Кемеровский государственный университет»
Кафедра ЮНЕСКО по Новым информационным технологиям

«УТВЕРЖДАЮ»
Декан математического факультета

_____ Н.Н. Данилов
« » _____ 20 г.

РАБОЧАЯ ПРОГРАММА

по курсу *"Базы данных и СУБД". Часть II*

Специальность: **010503.65 «Математическое обеспечение и администрирование информационных систем»**, ОПД.Ф.05

Факультет: *Математический*

Курс:	III		
Семестр:	5		
Лекции:	32	Экзамен:	5 семестр
практические занятия:			
лабораторные занятия:	32		
коллоквиум:	4		
самостоятельные занятия:	30		
Всего часов:	98		
Составитель:	доцент, Гудов А.М.		

Кемерово, 2008

Рабочая программа дисциплины федерального компонента цикла ОПД.Ф.05 «Базы данных и СУБД» составлена на основании Государственного образовательного стандарта высшего профессионального образования второго поколения по специальности 010503 «Математическое обеспечение и администрирование информационных систем», утвержденном 10 марта 2000 г., № 72 МЖД/ СП.

Рабочая программа обсуждена на заседании кафедры

Протокол № _____ от « _____ » _____ 200__ г.

Зав.кафедрой _____ К.Е. Афанасьев

Одобрено методической комиссией

Протокол № _____ от « _____ » _____ 200__ г.

Председатель _____

1. Пояснительная записка

Рабочая программа полностью соответствует Государственному образовательному стандарту высшего профессионального образования второго поколения по специальности 010503 «Математическое обеспечение и администрирование информационных систем».

Актуальность и значимость дисциплины. Дисциплина является одной из основных в цикле общих профессиональных дисциплин, в рамках которой изучаются основные понятия, базовые методы и алгоритмы манипуляций с данными, основные элементы и технологии, используемые в организации современных СУБД.

Цель и задачи изучения дисциплины. Главная цель преподавания курса - освоение базовых знаний по вопросам организации параллельных и распределенных баз данных и систем управления ими, а также основ проектирования приложений в рамках архитектуры «клиент-сервер» с использованием СУБД.

Объектами изучения в данной дисциплине являются: основные структуры, модели, методы и технологии, используемые для хранения и анализа данных; средства и методы проектирования БД различной архитектуры; хранилища данных и методы доступа к ним; языковые средства современных СУБД.

Основными задачами изучения дисциплины являются:

- знакомство с основными технологиями построения современных баз данных и СУБД;
- знакомство с технологиями распределенных и параллельных СУБД;
- приобретение навыков работы с БД и хранилищами данных;
- знакомство с OLAP-технологиями;
- приобретение навыков создания приложений в архитектуре «клиент-сервер»;
- приобретение навыков использования процедурного языка PL/SQL для создания приложений, манипулирующих с данными на стороне сервера БД.
- подготовка студентов к изучению других дисциплин по информационным технологиям.

Место дисциплины в профессиональной подготовке специалистов. Курс занимает особое место в учебном плане среди дисциплин факультета по его значению. Вместе с курсами по программированию, курс БД и СУБД составляет основу образования студента в части информационных технологий. Курс рассчитан на студентов-математиков, имеющих подготовку по математике и информатике в объеме программы средней школы. В течение преподавания курса предполагается, что студенты знакомы с основными понятиями ал-

гебры, комбинаторики, логики, информатики, которые читаются на факультете перед изучением данной дисциплины.

Требования ГОС ВПО к обязательному минимуму содержания дисциплины.

Индекс	Наименование дисциплин и их основные разделы	Всего часов
ОПД	ОБЩЕПРОФЕССИОНАЛЬНЫЕ ДИСЦИПЛИНЫ	
<i>ОПД.Ф.00</i>	<i>Федеральный компонент</i>	
ОПД.Ф.05	<p>БАЗЫ ДАННЫХ И СУБД</p> <p>Базы данных (БД) и системы управления базой данных (СУБД). Выбор системы управления базами данных. Жизненный цикл базы данных. Уровни моделей и этапы проектирования БД. Инфологическое моделирование. Языковые средства современных СУБД. Даталогическое моделирование. Проектирование на физическом уровне. Средства и методы проектирования БД. Реляционные СУБД. СУБД на инвертированных файлах. Гипертекстовые и мультимедийные БД. XML-серверы. Объектно-ориентированные БД. Распределенные БД. Коммерческие БД.</p> <p>Организация процессов обработки данных в БД. Ограничения целостности. Технология оперативной обработки транзакции (OLTP–технология). Информационные хранилища. OLAP-технология.</p> <p>Проблема создания и сжатия больших информационных массивов, информационных хранилищ и складов данных. Основные математические методы, применяемые при сжатии информации. Фрактальные методы в архивации. Управление складами данных.</p>	

Структура учебной дисциплины. Дисциплина включает следующие разделы: распределенные и параллельные СУБД; параллельные архитектуры баз данных, объектно ориентированные базы данных, многомерные базы данных и OLAP-технологии, процедурный язык PL/SQL, проектирование приложений с использованием конструкций PL/SQL и триггеров базы данных.

Особенности изучения дисциплины. Особенностью курса является его «разделение» на два больших блока: изучение теоретического материала на лекционных занятиях и практическое изучение языка PL/SQL, а также создание приложения на его основе на лабораторных занятиях и в рамках самостоятельной работы студента.

Форма организации занятий по дисциплине. По дисциплине читаются лекции в течение одного семестра по два часа в неделю. В течение одного семестра лабораторные занятия в объеме двух часов в неделю в компьютерном классе. Кроме того, студенту предлагается выполнить семестровое задание в рамках самостоятельной работы. Один раз в семестре проводится коллоквиум по дисциплине, состоящий из теоретической части и двух практических заданий.

Требования к уровню усвоения содержания материала. В результате изучения курса студенты должны усвоить основные теоретические и практические вопросы, определенные содержанием дисциплины, научиться пользоваться полученными знаниями в смежных предметах, научиться применять технологии, язык манипулирования данными, процедурный язык, учебную и методическую литературу для решения поставленных задач.

Виды контроля знаний и их отчетности. По разделам основной части курса предусмотрены самостоятельные задания, семестровые задания, для выполнения которых требуются элементы исследовательской работы, коллоквиум, тест. По итогам изучения курса предусмотрен экзамен.

Критерии оценки знаний студентов. Предусмотрена рейтинговая система оценки всех видов деятельности. Максимальное число баллов устанавливается преподавателем в зависимости от набора, степени сложности и обязательности выполнения каждого из заданий и видов деятельности. Каждый вид деятельности оценивается следующим образом:

- работа на лекции – 1 балл за каждое занятие;
- выполнение самостоятельных заданий – максимально 2 балла каждое (0 – задание не выполнено, 1 – задание выполнено полностью, 2 – использован оптимальный алгоритм, особые конструкции языка и т.д.);
- семестровые задания – 10 баллов за выполнение работы;
- коллоквиум – максимальное количество баллов устанавливается преподавателем (рекомендуется написать 3 теоретических вопроса и решить 2 практические задачи);
- общее количество баллов за практическую работу в семестре определяется суммированием всех баллов, полученных за каждый вид деятельности.

Студент, набравший 50% и более от максимального количества баллов, выполнивший семестровое задание получает допуск к экзамену.

Экзамен проходит в 2 этапа:

1. Тестирование (100 баллов максимум, ниже 50 баллов – студент получает оценку «неудовлетворительно»).
2. Опрос по дополнительным вопросам (1 балл за каждый) или/и решение дополнительных практических задач (до 2-х баллов за каждую) при условии наличия пропусков лекций или лабораторных занятий.

Экзамен оценивается по бальной системе с учетом результатов теста; результатов ответа на дополнительные вопросы и/или решения дополнительных практических задач.

Итоговая оценка по дисциплине выставляется на основании общего количества баллов, полученных за работу в семестре и за экзамен. После суммирования всех баллов определяется максимальный балл. Оценка за дисциплину выставляется по следующему правилу: «отлично» – от 85% до 100% от максимального балла; «хорошо» – от 65% до 84%; «удовлетворительно» – от 40% до 64%; «неудовлетворительно» – менее 40%.

2. Тематический план

№	Название и содержание разделов, тем, модулей	Объем часов				Формы контроля
		Общий	Аудиторная работа		Самостоятельная работа (в часах)	
			лекции	практические		
1	2	3	4	5	6	7
Очная форма обучения						
1.	Распределенные и параллельные СУБД	6	6			тестовые задания
2.	Параллельные архитектуры баз данных	4	4			тестовые задания
3.	Объектно ориентированные базы данных	4	4			тестовые задания
4.	Многомерные базы данных	6	4	2		тестовые задания
5.	коллоквиум	4	2	2		Задания для коллоквиума
6.	Технологии баз данных для WWW	6	6			тестовые задания
7.	Введение в PL/SQL	4	2	30		тестовые задания, самостоятельные задания
8.	Проектирование приложений с использованием конструкций PL/SQL и триггеров базы данных	34	6		30	семестровое задание
	Итого:	98	34	34	30	

1. Содержание дисциплины

3.1 Содержание теоретического курса

1. Распределенные и параллельные системы управления базами данных
 Основные определения. Отличительные признаки распределенных и параллельных баз данных. Горизонтальная и вертикальная фрагментация. Понятия межзапросного, внутрizaпросного и внутриоперационного параллелизма. Понятия линейной расширяемости и линейного ускорения. Основные архитектуры параллельных систем. Обработка и оптимизация запросов в распределенных и параллельных системах. Управление доступом. Протоколы обеспечения надежности.

2. Параллельные архитектуры баз данных

Основные типы архитектур. Требования к параллельным СУБД. Факторы производительности параллельных СУБД. Вертикальный и горизонтальный параллелизмы. Системы со смешанной загрузкой. Обеспечение надежности функционирования параллельных СУБД.

3. Объектно ориентированные базы данных

Модель объектно ориентированной базы данных. Типы данных в ООБД. Оптимизация ядра ООСУБД. Языки запросов к ООБД. Механизмы управления транзакциями в ООСУБД. Особенности реализации некоторых ООСУБД.

4. Многомерные базы данных

Многомерные модели данных. Хранилище многомерных данных. Типы запросов к многомерным данным. Основные формы реализации многомерных БД.

5. Технологии баз данных для WWW

Основные классы задач для управления БД из WWW. Представление данных для задач Web/DB. Моделирование структуры WEB и запросы к базам данных. Основные языки запросов к данным под управлением WWW.

6. Введение в PL/SQL

Что такое PL/SQL? Основные отличия от языка SQL. Назначение и формы реализации хранимых приложений. Что будем изучать? И зачем? Анонимный и именованный блок. Структура блока. Процедура и функция. Формальные и фактические параметры. Исключения. Как выполнить блок? А как получить результат. Переменные. Базовые и сложные типы данных. Описание переменных, констант и курсоров. Курсор. Конструкция IF – THEN – ELSE – END IF. Оператор присваивания. Конструкция цикла. Что описать в исключении? Как вызвать исключение? Метки. Конструкция EXIT – WHEN. Пакеты: спецификация и тело. Где располагаются и выполняются хранимые процедуры. Пакеты хранимых процедур. Отладка и диагностика ошибок. Хранимые процедуры и словарь базы данных. Компиляция и выполнение. Связность объектов базы данных. Статус хранимой процедуры и пакета. Обработчик событий при обращении к объектам базы данных – триггер. Типы триггеров. Сколько раз выполнить триггер? Рекурсивный вызов триггеров. Когда нужен триггер? А когда он полезен? Процедура или триггер? Контроль над объектами базы данных с использованием триггеров. Аудит – это просто.

7. Проектирование приложений с использованием конструкций PL/SQL и триггеров базы данных

Основные приемы создания серверных приложений. Распределенные приложения. Задание для семестровой работы. Типовая модель данных. Как реализовать логику приложения, выполняющегося на сервере базы данных?

3.2 Содержание лабораторных занятий

- 1.
2. Первая программа на PL/SQL.
3. Работа с основными конструкциями PL/SQL.
4. Процедуры и функции.
5. Пакет.
6. Хранимые процедуры и функции.
7. Компиляция и отладка хранимых процедур и пакетов.
8. Связность объектов базы данных. Статус хранимых процедур и пакетов.
9. Основные приемы реализации приложений с использованием хранимых процедур и пакетов.
10. Триггера базы данных.
11. Контроль использования таблиц из приложений.

№ п/п	Тема занятия	Задания	
		В аудитории	Дома
1	Простой блок PL/SQL	1. Создайте процедуру, принимающую два числа через переменные. Первое нужно разделить на второе и к результату прибавить второе число. Результат должен быть записан в переменную PL/SQL и выведен на экран.	2. Создайте функцию, вычисляющую общее вознаграждение за год. Функции должны передаваться годовая зарплата и процент премиальных. Премиальные необходимо преобразовать из целого числа в десятичное. Если зарплата не определена, то функция должна выдавать значение «ноль». Если премия не определена, функция должна выдавать только зарплату.
2	Модульное программирование	1. Создайте процедуру для включения нового отдела в таблицу S_DEPT. 2. Создайте процедуру, обновляющую номер региона для существующего отдела. 3. Создайте процедуру для удаления отдела, созданного в упражнении 1.	4. Создайте процедуру NEW_EMP для включения записи о новом служащем в таблицу S_EMP.
3	Циклы	1. Создайте таблицу ZAKAZ со следующими полями (id number, order_name varchar2, product_id number, quantity number, comment varchar2). - Для всех заказчиков, общая сумма заказов	2. Создайте функцию вычисления либо общего количества лет, которые сотрудник проработал на фирме либо даты десятилетия его рабочей деятельности в зависимости

		<p>которых превышает 50000, внесите данные в таблицу. Для задания первичного ключа создайте последовательность.</p> <ul style="list-style-type: none"> - Для всех заказчиков не принадлежащих региону «Europe» добавить комментарий «out region» - Проверить количество обновленных строк. Если были обновлены все строки, то вывести сообщение «нет заказчиков из европейского региона». 	<p>от флага.</p> <p>Создайте таблицу STAG со следующими полями (id number, emp_id, FIO varchar2, ord_id number, ord_name varchar2, commentar varchar2). Заполните таблицу данными, имеющимися в базе (используйте данные из таблиц). Для задания первичного ключа создайте последовательность.</p> <ul style="list-style-type: none"> - Для всех сотрудников, внесенных в таблицу STAG, с четными номерами добавьте в столбец комментария сообщение «стаж XXX лет», где XXX- вычисленное количество отработанных сотрудником лет. - Для всех сотрудников, внесенных в таблицу STAG, с нечетными номерами добавьте в столбец комментария сообщение «До 10-летия раб. деят. Осталось XXX лет», где XXX – вычисленное количество лет, которые осталось проработать сотруднику до 10-тилетнего юбилея. - Определить количество сотрудников, проработавших на фирме более 10 лет. Вывести сообщение «XXX сотрудников проработало на фирме более
--	--	---	---

			<p>10 лет», где XXX – вычисленное количество сотрудников.</p> <p>Если у сотрудника в текущую дату (sysdate) – юбилей, то вывести на экран следующее сообщение «Поздравляем СОТРУДНИК_ИМЯ с 10-тилетием его рабочей деятельности!»</p>										
4	Управление потоками в блоках PL/SQL	<p>1. Скопируйте таблицу s_emp в s_emp_sору. Для s_emp_sору напишите набор функций:</p> <ul style="list-style-type: none"> – Вставка строки в таблицу. Если отдел не введен то сотрудник заносится в самый малочисленный отдел. – Выбор строки. Если указан id, то выводится строка с указанным ключом. Если нет, то происходит поиск по фамилии, имени, дате начала работы, году начала работы, отделу и заработной плате. – Удаление. Если указана id, то удаляется строка с указанным ключом. Если нет, то происходит поиск по фамилии, имени, дате начала работы, году начала работы, отделу и заработной плате. <p>2. Создайте процедуру SET_COMM, которая устанавливает процент комиссионных для служащего в зависимости от общего объема продаж</p> <ul style="list-style-type: none"> – Создайте параметр, чтобы можно было ввести номер служащего. – Подсчитайте сумму всех заказов, размещенным этим служащим. – Если эта сумма меньше 10000 установите процент комиссионных «10». – Если сумма находится в диапазоне от 10000 до 	<p>4. Создайте процедуру EMP_MESSAGE, которая выбирает фамилию служащего, дату начала работы и заработную плату по номеру служащего, задаваемому во времени выполнения. Выдайте на экран сообщение по любой комбинации нижеуказанных критериев (используйте вложенные операторы IF)</p> <table border="1" data-bbox="1054 1144 1444 1664"> <thead> <tr> <th>Критерий</th> <th>Сообщение</th> </tr> </thead> <tbody> <tr> <td>Зарплата более 1200</td> <td>Salary >1200</td> </tr> <tr> <td>Фамилия содержит "R"</td> <td>Name contains "R"</td> </tr> <tr> <td>Дата начала работы приходится на март</td> <td>March start date</td> </tr> <tr> <td>Ничего из вышеуказанного</td> <td>**None**</td> </tr> </tbody> </table>	Критерий	Сообщение	Зарплата более 1200	Salary >1200	Фамилия содержит "R"	Name contains "R"	Дата начала работы приходится на март	March start date	Ничего из вышеуказанного	**None**
Критерий	Сообщение												
Зарплата более 1200	Salary >1200												
Фамилия содержит "R"	Name contains "R"												
Дата начала работы приходится на март	March start date												
Ничего из вышеуказанного	**None**												

		<p>1000000, установите процент комиссионных «15».</p> <ul style="list-style-type: none"> – Если в таблице S_ORD нет заказов, размещенных данным служащим, установите процент комиссионных «0». <p>3. Создайте процедуру CUST_UPDATE с циклом для обработки кредитного рейтинга всех заказчиков по регионам. Не фиксируйте изменения</p> <ul style="list-style-type: none"> – Если номер региона четный, установите кредитный рейтинг “Excellent”, даже если он уже был таким; в противном случае установите кредитный рейтинг “Good”. – Когда строки будут обновлены, определите количество обновленных строк. В зависимости от того, каким будет это количество, выдайте на экран следующую информацию: <ul style="list-style-type: none"> - Если обновлено менее трех строк, сообщение : “Fewer then 3 customer records updated for region number X”, где X – номер региона. - В противном случае – “Y rows updated for region number X”, где Y – количество обновленных строк, X – номер региона. <p>2. Отмените изменения. Задайте точку останова на проверке количества обновленных строк.</p>	
5	Обработка исключений	<p>1. Создайте процедуру, которая бы обновляла номера регионов для служащих отделов. (Входные данные – название региона и новый номер отдела для данного служащего).</p> <ul style="list-style-type: none"> – Напишите обработчик исключений, который будет 	

		<p>выдавать сообщение о том, что указанный регион не существует.</p> <ul style="list-style-type: none"> – Напишите обработчик исключений, который бы выдавал пользователю сообщение о том, что для указанного региона уже есть отдел с таким названием. – Напишите обработчик исключений, который выдавал бы пользователю сообщение о том, что указанный номер отдела не существует (используйте атрибут SQL%NOTFOUND и возбудите исключение вручную). Напишите процедуру для вывода на фамилии и названия отдела для служащих, чья заработная плата лежит в диапазоне плюс-минус 100\$ от введенного значения. – Напишите обработчик исключений, который будет выдавать сообщение о том, что служащих с такой зарплатой нет. – Напишите обработчик исключений, который будет выдавать сообщение о том, что служащих с такой зарплатой несколько. Сообщение должно указывать, сколько сотрудников попадает в этот диапазон зарплат. 	
6	Курсоры	<p>1. Создайте процедуру TOP_DOGS1 для определения самых высокооплачиваемых служащих</p> <ul style="list-style-type: none"> – Для этого упражнения создайте новую таблицу с данными о служащих и их заработной плате. – Включите параметр, чтобы пользователь мог ввести нужное количество самых высокооплачиваемых служащих (n). – Создайте цикл FOR с кур- 	<p>7. Скопируйте таблицу s_emp в s_emp_copу. Для s_emp_copу напишите пакет функций:</p> <ul style="list-style-type: none"> – Вставка строки в таблицу. Если отдел не введен то сотрудник заносится в самый малочисленный отдел. – Выбор строки. Если указан id, то выводится строка с указанным ключом. Если нет, то

		<p>сором для выборки из таблицы S_EMP фамилий и заработной платы n самых высокооплачиваемых служащих.</p> <ul style="list-style-type: none"> – Сохраните фамилию и заработную плату в таблице TOP_DOGS. – Предполагается что двух служащих с одинаковой заработной платой не существует. – Проверьте особые случаи – например, с n=0 и с n, превышающим количество служащих в таблице S_EMP. – После каждого теста удаляйте данные из таблицы TOP_DOGS.Создайте хранимую процедуру ADD_STARS, которая в новом столбце STARS проставляет для каждого служащего по призовой звёздочке за каждый процент заработанных комиссионных. Используйте курсор и цикл WHILE.Для этого упражнения создайте в таблице S_EMP новый столбец для хранения звёздочек (*) – Определите процент комиссионных для каждого служащего, округлив его до ближайшего целого числа. Рассмотрите случай, когда служащий не получает комиссионных. – Добавляйте звёздочку в строку звёздочек за каждый процент комиссионных. Если, например, служащий получает 10 процентов комиссионных, символьная строка в столбце STARS должна содержать десять звёздочек. – Проставьте соответствующее количество звёздочек для каждого служащего в 	<p>происходит поиск по фамилии, имени, дате начала работы, году начала работы, отделу и заработной плате.</p> <ul style="list-style-type: none"> – Удаление. Если указана id, то удаляется строка с указанным ключом. Если нет, то происходит поиск по фамилии, имени, дате начала работы, году начала работы, отделу и заработной плате. <p>Разработать структуру логирования информации для таблицы s_emp_soru с возможностью восстановления данных за нужный период.</p> <p>8. Напишите процедуру, осуществляющую поиск в зависимости от входных параметров:</p> <ul style="list-style-type: none"> – Указан интервал времени. Поиск заказчиков у которых время, прошедшее от заказа до получения товара больше чем указанный интервал. – Наименование товара. Список заказчиков, закупавших товар с таким наименованием (при этом не поиск должен работать независимо от указанного регистра или числа пробелов). – Сумма. Заказчиков, приобретших товар на сумму более указанной. – Добавить в качестве входного параметра флаг. В первом случае он позволяет выбирать заказчиков по условиям независимо. Выбранных заказчиков без повторения
--	--	---	---

		<p>столбце STARS.</p> <p>3. Скопируйте процедуру TOP_DOGS1 из упражнения 1 и назовите новый вариант TOP_DOGS2. Измените процедуру TOP_DOGS2 с учётом случая, когда несколько служащих из упражнения 1 имеют одинаковую заработную плату. Для каждой фамилии в списке должны быть перечислены все служащие с такой же заработной платой. Выполните процедуру TOP_DOGS2. В качестве n введите число 6, 7 или 8. В этом случае в выходных данных должны появиться фамилии Ngao, Dumas и Quick-To_See. Если же n равно 9, 10 или 11, должны появиться фамилии Nagayama, Magee и Maduro. Не забывайте полностью удалять данные из таблицы TOP_DOGS после каждого теста.</p> <p>4. Напишите процедуру для печати фамилии служащих, чья заработная плата лежит в диапазоне плюс минус \$100 от введённого значения.</p> <ul style="list-style-type: none"> – Если служащего с такой заработной платой нет, пользователь должен получить соответствующее сообщение. Используйте исключение. – Если служащих с такой зарплатой более 3, сообщение должно указывать, сколько сотрудников попадёт в этот диапазон зарплат. <p>5. Создайте процедуру для определения: Сотрудников, работающих больше заданного числа лет.</p> <ul style="list-style-type: none"> – Для этих сотрудников определите менеджеров. Выведите без повторений полученный список менеджеров. 	<p>поместить в специальную таблицу и в отдельном столбце в ней ставить столько плюсииков, сколько пунктов для заказчика истинно.</p> <p>Второе значение флага позволяет выбирать заказчиков удовлетворяющих всем указанным пунктам.</p> <p>9. Создайте процедуру EMP_MESSAGE, которая выбирает фамилию служащего, дату начала работы и зарплату по номеру служащего заданному при выполнении. Выдайте на экран сообщение с любым сочетанием нижеуказанных критериев. Проверьте номера сотрудников 2,5,16,17 и 18.</p>
--	--	---	---

		<p>жеров с указанием рядом с каждым именем через запятую имён сотрудников из начального списка.</p> <ul style="list-style-type: none"> – Определите и выведите на экран менеджеров, принявших суммарные заказы более заданной суммы. <p>6. Напишите процедуру, определяющую рейтинг лучших товаров по: суммарной цене приобретённого за указанный срок товара;</p> <ul style="list-style-type: none"> – количеству единиц приобретённого за указанный срок товара; – количеству заказов, в которые входил приобретённого за указанный срок товар. <p>Т.е. первым выдаётся на печать товар, у которого максимальный показатель по выбранному критерию, затем товар с более низким показателем и т.д.</p> <p>Какой из критериев использовать определяется по входным параметрам. Если указано несколько критериев, то их приоритет соответствует списку, указанному выше.</p>	
7	Динамический SQL	<p>1. Напишите процедуру, возвращающую список товаров, количество каждого товара и его цену. Процедура должна иметь два входных параметра - название фирмы и название товара. Если указано название фирмы, выдаётся список всех купленных ею товаров. Если указано название фирмы и дополнительно указано наименование товара, то выдаётся товар, купленный фирмой, наименование которого совпадает с заданным наименованием. Если указано только наименование товара, то выдаётся указанный товар.</p> <p>Процедуру написать с исполь-</p>	

		зованием позиций предназначенных для размещения переменных PL/SQL, указываемых в предложении USING оператора EXECUTE IMMEDIATE.	
8	Триггеры DML	<ol style="list-style-type: none"> 1. Для отслеживания изменений информации о сотрудниках, создайте таблицу s_emp_log и напишите триггер, заносящий в s_emp_log информацию о времени изменения и информации о сотруднике до изменения. 2. Для отслеживания удалений информации о сотрудниках, напишите триггер, заносящий в s_emp_log удалённую информацию, а также время удаления. 3. Напишите триггер для таблице s_item, позволяющий контролировать изменение цены товара. Если новая цена отличается от старой более чем на 30%, выдаётся соответствующее сообщение и запрещается изменения данных. Реализовать используя исключения. 	<ol style="list-style-type: none"> 4. Требуется отслеживать статистические показатели, касающиеся продуктов. Т.е. для каждого существующего товара указывается количество заказчиков этого товара, количество заказанных единиц, сумма заказа. Результаты будут храниться в таблице major_stats. 5. Напишите триггер который разрешает работу команды DML с таблицами в нормальное рабочее время с 8:45 до 5:30, с понедельника по пятницу. Создайте процедуру SECURE_DML, которая при непопадании в эти часы выводит сообщения: "Data changes restricted to office hours".
9	Триггеры за-мещения	<ol style="list-style-type: none"> 1. Создайте представление, содержащее имя отдела, номер региона, Фамилию, имя, должность и номер сотрудника. Операции DML для данного представления определим по следующим правилам: <ul style="list-style-type: none"> – INSERT – Назначить отдел, введенному сотруднику. В результате s_dept (при необходимости s_region) обновляются. – UPDATE – Изменить отдел, назначенный сотруднику. Это может привести к обновлению или s_emp или s_dept, в зависимости от того, какой столбец представления обновляется. – DELETE – Очистить идентификатор отдела для со- 	

		<p>трудника (сотрудник не зачислен ни в один отдел). В результате s_emp обновляется: ID устанавливается в значении NULL.</p> <p>Создайте триггер, реализующий сформулированные выше правила и позволяющий правильно выполнять операции DML над созданным представлением.</p> <p>Входные данные процедуры – ФИО, должность сотрудника, название отдела. Зарботную плату сотруднику определить как среднюю з/п всех сотрудников фирмы.</p>	
--	--	--	--

3.3 Семестровая работа, ее характеристика

1. Для выполнения работы необходимо знание всего материала по двум частям курса “БД и СУД”.
2. В рамках работы предлагается реализовать элементы проекта создания ИС, который разрабатывался в рамках курса «Технологии разработки программного обеспечения», в части обеспечения будущей ИС интерфейсами доступа к данным, а также предварительной обработки результатов запросов перед их отправкой на сторону клиента, где они должны быть представлены средствами интерфейса пользователя.
3. Первым шагом выполнения работы должна быть разработка структуры хранимых данных (построение ER-диаграммы или диаграммы классов) и написание sql-скриптов, создающих и наполняющих первоначальными данными разработанную структуру.
4. При создании таблиц обязательным является установка ограничений на поля таблиц с целью обеспечения целостности данных.
5. Логика должна быть реализована в виде процедур и функций PL/SQL, объединенных в пакеты в соответствии общностью их функционала. Название пакета должно отражать его назначение. Например, пакет PERSON содержит процедуры и функции, обеспечивающие управление данными пользователей ИС.
6. Обращения к базе данных только через пакеты процедур и функций.
7. Вся работа с данными (добавление, изменение, удаление) должна быть реализована в виде процедур и функций PL/SQL. При добавлении данных в таблицу для генерации уникального идентификатора необходимо использовать последовательности.

8. Реализованный набор пакетов PL/SQL должен полностью покрыть заявленный в требованиях к ИС функционал и быть достаточным, для дальнейшей реализации интерфейса ИС.
9. Семестровые задания даются каждому студенту индивидуально.

10. Результатом выполнения семестровой работы является:

- Набор структур данных (таблиц), полностью поддерживающих необходимый функционал разрабатываемой системы. Скрипт (имя_файла_латинскими_символами.sql), реализующий процесс создания таблиц и занесения в них первичных данных.
- Описание структур данных, представленное в виде информационных объектов, сущностей или классов (ER-диаграмма или диаграмма классов).
- Пакет (набор пакетов) процедур и/или функций, реализованный с помощью PL/SQL и обеспечивающий выполнение предыдущих требований. Скрипт (имя_файла_латинскими_символами.sql), реализующий создание пакета.
- Отчет о выполнении семестровой работы в электронном виде, составленный в соответствии с требованиями к оформлению (<http://unesco.kemsu.ru/student/rule/rule.html>). К отчету должны быть присоединены в качестве приложений: ER-диаграмма (диаграмма классов), файлы со скриптами.

4. Учебно-методические материалы по дисциплине

Презентации в формате MS PowerPoint. Тексты лабораторных заданий в формате PDF.

Основная литература

1. Мейер М. Теория реляционных баз данных. М.: Мир, 1987. - 608 с.
2. Кириллов В.В. Основы проектирования реляционных баз данных. Учебное пособие. - СПб.: ИТМО, 1994. - 90 с.
3. Дейт К. Введение в системы баз данных. М.: Наука, 1980

4. Гудов А.М., Шмакова Л.Е. Введение в язык структурированных запросов SQL / Учебное пособие. – Кемерово, Кемеровский госуниверситет, 2001.- 118с.

Дополнительная литература

5. Озкарахан Э. Машины баз данных и управление базами данных. - М.: Мир, 1989
6. Ульман Д. Основы систем баз данных. - М.: Финансы и статистика, 1983. - 334 с.
7. Chen P.P. The Entity-Relationship Model: A Basis for the Enterprise View of Data. – ACM Transaction on Database Systems, 1976, Vol. 1, No. 1, pp. 9-36.
8. Материалы сайта <http://www.citforum.ru>

Список основной учебной литературы

<i>Сведения об учебниках</i>			<i>Количество экземпляров в библиотеке на момент утверждения программы</i>
<i>Наименование, гриф</i>	<i>Автор</i>	<i>Год издания</i>	
Теория реляционных баз данных	Мейер М.	1987	5
Основы проектирования реляционных баз данных	Кириллов В.В.	1994	2
Введение в системы баз данных	Дейт К.	1980	2
Введение в язык структурированных запросов SQL	Гудов А.М., Шмакова Л.Е.	2001	150

5. Формы текущего, промежуточного и рубежного контроля

Примерные варианты семестровых заданий

В рамках выполнения семестровой работы по курсу “технология разработки программного обеспечения” Вами была выбрана информационная система (ИС), определены для неё требования и построены модели. На основании этих данных необходимо реализовать серверную часть выбранной информационной системы, используя процедурный язык PL/SQL. В качестве сервера приложений, выступающего также в роле сервера базы данных, выступает Oracle 8i.

Вопросы и задачи к коллоквиуму

Для всех заданий: необходимо собрать все задания в пакет; пакет в виде скрипта, его создающего, сохранить в файле; дать файлу имя, состоящее из своей фамилии и номера группы; сохранить файл в специальном каталоге, указанном преподавателем.

1. Создайте процедуру для определения списка самых высокооплачиваемых сотрудников. Количество сотрудников в списке должно передаваться через входной параметр.
2. Создайте процедуру, выводящую список сотрудников в виде таблицы, состоящей из двух столбцов. В первом должны быть помещены фамилия и имя сотрудника, во второй – последовательность символов «*», где каждый символ соответствует 5-ти процентам заработанных сотрудником комиссионных.
3. Создайте процедуру, которая для всех служащих, имеющих одинаковую зарплату, выводит сумму и список сотрудников, получающих эту сумму.
4. Создайте процедуру, которая в зависимости от номера заказа выдает позиции этого заказа и число, вычисляемое по следующему правилу: если эта позиция была использована в других заказах, - количество таких заказов; если позиция не участвует больше в других заказах – номер текущего заказа со знаком «-». Номер заказа должен передаваться в процедуру через входной параметр.

Вопросы к экзамену

1. Каковы основные отличительные черты SQL и PL/SQL? Привести примеры задач, для которых специфичен каждый из языков.
2. Структура блока PL/SQL. Анонимные и именные блоки.
3. Типы данных в PL/SQL. Базовые и сложные типы. Переменные.
4. Процедуры и функции. Параметры. Пример описания процедуры и функции.
5. Циклические структуры PL/SQL. Привести примеры всех структур.
6. Условные конструкции PL/SQL. Привести пример.
7. Курсор. Каково отличие между явным и неявным курсором? Способы обработки записей в курсоре.
8. Исключения. Предопределенные исключения и исключения пользователя. Пример описания и возбуждения исключений.
9. Хранимые процедуры и функции. Чем они отличаются от обычных процедур и функций? Примеры описаний. Передача параметров.
10. Триггера базы данных. Классификация триггеров.
11. Пакет. Компиляция, вызов, отладка процедур из пакета.
12. Как можно контролировать обращения к таблицам базы данных через использование триггеров? Привести пример.
13. Как отладить триггер? Создать триггер, предложенный преподавателем. Отладить его.
14. Написать процедуру или функцию, реализующую задачу, предложенную преподавателем.
15. Связность объектов базы данных. Зависимость статуса пакета или процедуры от статуса объектов базы данных.
16. Провести диагностику процедуры, предложенной преподавателем. Отладить и сформировать в пакет. Провести всестороннее тестирование пакета.

17. Типы триггеров. Когда выполняется триггер? Рекурсивный вызов триггера. Манипуляции с триггерами.
18. Какие принципы должны учитываться при разработке приложения, использующего базы данных? Сервер или клиент – что лучше?

Задачи к экзамену

1. Создайте процедуру EMP_MESSAGE, которая выбирает фамилию служащего, дату начала работы и зарплату по номеру служащего заданному при выполнении. Выдайте на экран сообщение с любым сочетанием нижеуказанных критериев. Проверьте номера сотрудников 2,5,16,17 и 18. Подсказка: требуются вложенные предложения IF.
2. Напишите процедуру SALARY_RANGE для печати фамилий служащих, чья зарплата лежит в диапазоне плюс минус \$100 от введённого значения.
 - Если служащего с такой заработной платой нет, то пользователь должен получить соответствующее сообщение. Используйте исключения.
 - Если служащих с такой заработной платой более 3, сообщение должно указывать на то, сколько сотрудников попадает в этот диапазон зарплат.
3. Напишите триггер, который разрешает работу команды DML с таблицами в нормальное рабочее время с 8:45 до 5:30, с понедельника по пятницу. Создайте процедуру SECURE_DML, которая при непопадании в эти часы выводит сообщение: "Data changes restricted to office hours".
4. Создайте триггер для определения того, что вы не добавляете уже существующий продукт к таблице S_PRODUCT.