

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**Методические рекомендации для
самостоятельной работы обучающихся по дисциплине**

Б1.В.20 SQL-программирование

Направление подготовки (специальность) 09.03.01 Информатика и вычислительная техника

Профиль образовательной программы Автоматизированные системы обработки информации и управления

Форма обучения очная

СОДЕРЖАНИЕ

1. Организация самостоятельной работы	3
2. Методические рекомендации по самостоятельному изучению вопросов	7
3. Методические рекомендации по подготовке к занятиям	9

1. ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1.1. Организационно-методические данные дисциплины

№ п.п.	Наименование темы	Общий объем часов по видам самостоятельной работы (из табл. 5.1 РПД)				
		подготовка курсового проекта (работы)	подготовка реферата/эссе	индивидуальные домашние задания (ИДЗ)	самостоятельное изучение вопросов (СИБ)	подготовка к занятиям (ПкЗ)
1	2	3	4	5	6	7
1	Тема 1 Определение структурированного языка запросов SQL				4	3
2	Тема 2 Эффективное выполнение запросов для извлечения данных				4	3
3	Тема 3 Построение нетривиальных запросов				4	3
4	Тема 4 Запросы модификации данных в реляционной таблице				4	3
5	Тема 5 Понятие представлений				4	3
6	Тема 6 Определение функций пользователя, примеры их создания и использования				4	3
7	Тема 7 Хранимые процедуры				4	3
8	Тема 8 Триггеры				8	3

2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО САМОСТОЯТЕЛЬНОМУ ИЗУЧЕНИЮ ВОПРОСОВ

2.1 Наименование вопроса. Реляционная база данных, СУБД. Классификация команд SQL.

При изучении вопроса необходимо обратить внимание на следующие особенности:

Рассматриваемый язык SQL ориентирован на операции с данными, представленными в

виде логически взаимосвязанных совокупностей таблиц-отношений.

Важнейшая особенность его структур – ориентация на конечный результат обработки данных, а не на процедуру этой обработки.

Язык SQL сам определяет, где находятся данные, индексы и даже какие наиболее эффективные последовательности операций следует использовать для получения результата, а потому указывать эти детали в запросе к базе данных не требуется.

2.2 Наименование вопроса. Синтаксис оператора SELECT. Построение условий выбора данных с применением операторов сравнения, логических операторов и логических связей.

При изучении вопроса необходимо обратить внимание на следующие особенности:

Оператор SELECT – один из наиболее важных и самых распространенных операторов SQL. Он позволяет производить выборки данных из таблиц и преобразовывать к нужному виду полученные результаты. Будучи очень мощным, он способен выполнять действия, эквивалентные операторам реляционной алгебры, причем в пределах единственной выполняемой команды. При его помощи можно реализовать сложные и громоздкие условия отбора данных из различных таблиц.

Оператор SELECT – средство, которое полностью абстрагировано от вопросов представления данных, что помогает сконцентрировать внимание на проблемах доступа к данным.

2.3 Наименование вопроса. Способ построения подзапросов. Способ построения подзапросов, возвращающих множественные и единичные значения с использованием операторов EXISTS, ALL, ANY.

При изучении вопроса необходимо обратить внимание на следующие особенности:

Часто невозможно решить поставленную задачу путем одного запроса. Это особенно актуально, когда при использовании условия поиска в предложении WHERE значение, с которым надо сравнивать, заранее не определено и должно быть вычислено в момент выполнения оператора SELECT. В таком случае приходят на помощь законченные операторы SELECT, внедренные в тело другого оператора SELECT. Внутренний подзапрос представляет собой также оператор SELECT, а кодирование его предложений подчиняется тем же правилам, что и основного оператора SELECT.

2.4 Наименование вопроса. Целостность данных. Целостность сущностей и ссылочная целостность.

При изучении вопроса необходимо обратить внимание на следующие особенности:

Чтобы информация, хранящаяся в базе данных, была однозначной и непротиворечивой, в реляционной модели устанавливаются некоторые ограничительные условия – правила, определяющие возможные значения данных и обеспечивающие логическую основу для поддержания корректных значений.

Ограничения целостности позволяют свести к минимуму ошибки, возникающие при обновлении и обработке данных.

2.5 Наименование вопроса. Роль представлений в вопросах безопасности данных. Процесс управления представлениями: создание, изменение, применение, удаление представлений.

При изучении вопроса необходимо обратить внимание на следующие особенности:

У СУБД есть две возможности реализации представлений. Если его определение простое, то система формирует каждую запись представления по мере необходимости, постепенно считывая исходные данные из базовых таблиц. В случае сложного определения СУБД приходится сначала выполнить такую операцию, как материализация представления, т.е. сохранить информацию, из которой состоит представление, во временной таблице. Затем система приступает к выполнению пользовательской команды и формированию ее результатов, после чего временная таблица удаляется.

Представление - это предопределенный запрос, хранящийся в базе данных, который выглядит подобно обычной таблице и не требует для своего хранения дисковой памяти. Для хранения представления используется только оперативная память. В отличие от других объектов базы данных представление не занимает

дисковой памяти за исключением памяти, необходимой для хранения определения самого представления.

2.6 Наименование вопроса. Типы функций. Встроенные функции языка SQL.

При изучении вопроса необходимо обратить внимание на следующие особенности:

Функция может содержать один или несколько входных параметров либо не содержать ни одного. Каждый параметр должен иметь уникальное в пределах создаваемой функции имя и начинаться с символа "@". После имени указывается тип данных параметра. Дополнительно можно указать значение, которое будет автоматически присваиваться параметру (DEFAULT), если пользователь явно не указал значение соответствующего параметра при вызове функции.

С помощью конструкции RETURNS скаляр_тип_данных указывается, какой тип данных будет иметь возвращаемое функцией значение. Дополнительные параметры, с которыми должна быть создана функция, могут быть указаны посредством ключевого слова WITH. Благодаря ключевому слову ENCRYPTION код команды, используемый для создания функции, будет зашифрован, и никто не сможет просмотреть его. Эта возможность позволяет скрыть логику работы функции. Кроме того, в теле функции может выполняться обращение к различным объектам базы данных, а потому изменение или удаление соответствующих объектов может привести к нарушению работы функции.

2.7 Наименование вопроса. Примеры создания, изменения и использования хранимых процедур с параметрами. Определение входных и выходных параметров. Примеры создания и вызова хранимых процедур.

При изучении вопроса необходимо обратить внимание на следующие особенности:

Хранение процедур в том же месте, где они исполняются, обеспечивает уменьшение объема передаваемых по сети данных и повышает общую производительность системы. Применение хранимых процедур упрощает сопровождение программных комплексов и внесение изменений в них. Обычно все ограничения целостности в виде правил и алгоритмов обработки данных реализуются на сервере баз данных и доступны конечному приложению в виде набора хранимых процедур, которые и представляют интерфейс обработки данных. Для обеспечения целостности данных, а также в целях безопасности, приложение обычно не получает

прямого доступа к данным – вся работа с ними ведется путем вызова тех или иных хранимых процедур.

Подобный подход делает весьма простой модификацию алгоритмов обработки данных, тотчас же становящихся доступными для всех пользователей сети, и обеспечивает возможность расширения системы без внесения изменений в само приложение: достаточно изменить хранимую процедуру на сервере баз данных. Разработчику не нужно перекомпилировать приложение, создавать его копии, а также инструктировать пользователей о необходимости работы с новой версией. Пользователи вообще могут не подозревать о том, что в систему внесены изменения.

2.8 Наименование вопроса. Определение триггера, область его использования, место и роль триггера в обеспечении целостности данных. Типы триггеров. Программирование триггера.

При изучении вопроса необходимо обратить внимание на следующие особенности:

Триггер представляет собой специальный тип хранимых процедур, запускаемых сервером автоматически при попытке изменения данных в таблицах, с которыми триггеры связаны. Каждый триггер привязывается к конкретной таблице. Все производимые им модификации данных рассматриваются как одна транзакция. В случае обнаружения ошибки или нарушения целостности данных происходит откат этой транзакции. Тем самым внесение изменений запрещается. Отменяются также все изменения, уже сделанные триггером. Создает триггер только владелец базы данных. Это ограничение позволяет избежать случайного изменения структуры таблиц, способов связи с ними других объектов и т.п.

3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ЗАНЯТИЯМ

3.1 Лабораторная работа №1, 2 ЛР-1, 2 Определение структурированного языка запросов SQL.

При подготовке к занятию необходимо обратить внимание на следующие моменты.

- Обратить внимание на способ нормализации таблиц и методы создания БД в оболочке MySQL
- Уметь работать с простейшими запросами, научить выводить поля из одной таблицы, из различных таблиц

3.2 Лабораторная работа №3, 4 ЛР-3, 4 Эффективное выполнение запросов для извлечения данных.

При подготовке к занятию необходимо обратить внимание на следующие моменты.

- Уметь создавать запросы с условием, в т.ч. с использованием логических операторов, сортировать данные по возрастанию и убыванию
- Уметь находить сумму полей, максимальное, минимальное, среднее значения полей, подсчитывать количество записей

3.3 Лабораторная работа №5, 6 ЛР-5, 6 Построение нетривиальных запросов.

При подготовке к занятию необходимо обратить внимание на следующие моменты.

- Знать основные строковые и числовые функции
- Уметь использовать оператор условного перехода CASE

3.4 Лабораторная работа №7, 8 ЛР-7, 8 Запросы модификации данных в реляционной таблице.

При подготовке к занятию необходимо обратить внимание на следующие моменты.

- Уметь создавать простые подзапросы
- Уметь использовать связанные подзапросы

3.5 Лабораторная работа №9, 10 ЛР-9, 10 Понятие представлений.

При подготовке к занятию необходимо обратить внимание на следующие моменты.

- Уметь соединять таблицы различными способами
- Уметь создавать запросы на добавление, удаление и изменение данных в таблицах

3.6 Лабораторная работа №11, 12 ЛР-11, 12 Определение функций пользователя, примеры их создания и использования.

При подготовке к занятию необходимо обратить внимание на следующие моменты.

- Знать понятие представление, уметь создавать представления, изменять данные в представлениях

- Уметь обрабатывать информацию из базы данных MySQL с помощью прикладных программ на примере Visual Basic

3.7 Лабораторная работа №13, 14 ЛР-13, 14 Хранимые процедуры.

При подготовке к занятию необходимо обратить внимание на следующие моменты.

- Уметь обрабатывать информацию из базы данных MySQL с помощью прикладных программ на примере DELPHI

- Уметь обрабатывать информацию из базы данных MySQL с помощью прикладных программ на примере C++

3.8 Лабораторная работа №15 ЛР-15 Триггеры.

При подготовке к занятию необходимо обратить внимание на следующие моменты.

- Уметь обрабатывать информацию из базы данных MySQL с помощью прикладных программ на примере C#