

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

Факультет среднего профессионального образования

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ
САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ**

ОПЦ.07 Информационные технологии в профессиональной деятельности

Специальность 25.02.08 Эксплуатация беспилотных авиационных систем

Форма обучения очная

Оренбург 2024 г.

СОДЕРЖАНИЕ

1. Организация самостоятельной работы	3
2. Общие подходы к этапам выполнения самостоятельной работы	4
3. Методические указания по изучению отдельных вопросов	7

1 семестр

1. Организация самостоятельной работы

1.1. Организационно-методические данные дисциплины

№ п.п	Наименование тем	Количество часов по видам самостоятельной работы		
		вопросы, выделенные на самостоятельное изучение (написание конспекта)	подготовка доклада	другие виды самостояте льной работы
1	Раздел 4. Введение в базы данных	6		
	Итого:	6		

2. ОБЩИЕ ПОДХОДЫ К ЭТАПАМ ВЫПОЛНЕНИЯ САМОСТОЯТЕЛЬНЫХ РАБОТ

Выполнение самостоятельной работы должно способствовать более полному усвоению программного материала.

Методические указания для выполнения самостоятельной работы составлены в соответствии с рабочей программой по дисциплине, предусматривают самостоятельное изучение многих экономических проблем, позволяет ознакомиться с экономическими аспектами деятельности организации, фирмы, отрасли, получить необходимые знания о важнейших показателях работы организации, их смысле, методике расчета.

Цель методических указаний – оказать помощь студентам в работе с учебной и научной литературой, в овладении определенными учебными умениями и навыками к которым относится:

- умение накапливать информацию;
- умение творчески ее перерабатывать;
- умение выдавать новую информацию;
- умение находить на все это время.

Самостоятельная работа студентов является основным способом овладения учебным материалом в свободное от обязательных учебных занятий время.

Самостоятельная работа студентов проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений студентов;
- углубления и расширения теоретических знаний;
- формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развития познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развития исследовательских умений.

В учебном процессе среднего специального учебного заведения выделяют два вида самостоятельной работы:

- аудиторная;
- внеаудиторная.

Для успешного овладения навыками самостоятельной работы студенту обязательно следует научиться работать с учебной литературой. Для чего необходимо знать основные пути к литературным источникам, знать, как их находить в библиотечных фондах, уметь выбрать в них нужную информацию, правильно ее обрабатывать.

Цели самостоятельной внеаудиторной работы студентов:

- закрепление, углубление, расширение и систематизация знаний, самостоятельное овладение новым учебным материалом;
- формирование профессиональных явлений;

- формирование умений и навыков самостоятельного умственного труда;
- мотивирование регулярной целенаправленной работы по освоению специальности;
- развитие самостоятельного мышления;
- формирование убежденности, волевых черт характера, способности к самоорганизации.

Основные виды самостоятельной работы студентов:

1. Первичное овладение знаниями (усвоение нового материала):

- чтение дополнительной литературы;
- составление плана текста, конспектирование прочитанного;
- выписки из текста;
- ознакомление с нормативными документами.

2. Закрепление и систематизация знаний:

- работа с конспектами лекций;
- изучение нормативных документов;
- ответы на контрольные вопросы;
- подготовка к выступлению на занятиях.

3. Применение знаний, формирование умений:

- решение задач по образцу, вариативных задач;
- выполнение ситуационных производственных задач;
- выполнение индивидуальных заданий.

Основные этапы выполнения самостоятельной работы:

1. Изучить содержание задания.

2. Подобрать литературу для получения ответов на задания.

3. Составить план выполнения задания:

3.1. Выбрать вопросы для изучения.

3.2. Определить сроки выполнения задания.

3.3. Согласовать с преподавателями намеченный план

4. Выполнить составленный план.

5. Убедиться, что задание выполнено:

5.1. Оценить в полном ли объеме материал.

5.2. Обдумать собранную информацию, обобщите ее.

5.3. Выяснить дополнительные вопросы, возникшие в ходе выполнения задания.

5.4. Изложить результаты выполнения задания в соответствии с указанием преподавателя. Студентам необходимо знать, что для каждого этапа выполнения плана самостоятельной работы они могут обращаться за консультацией к преподавателю или руководителю для дальнейшей работы и ее корректировки.

Рассмотрим основные правила выполнения и требования, предъявляемые к этим работам.

Методические указания по написанию конспекта по вопросам, выделенных на самостоятельное изучение

Что такое «конспект»? Конспект – это, прежде всего, изложение. Два главных его признака: он должен быть написан, информация сжатая. Многие до сих пор путают два понятия: краткое изложение основных положений и конспектирование. Но конспектирование – более широкий термин. Он предполагает сжатое изложение всей темы, а не только её тезисов.

Цели конспектирования:

- 1) переработка информации и трансформация её вида;
- 2) выделение в тексте самого необходимого с целью решения определённой задачи, ответа на определённый вопрос;

Виды конспектов:

- 1) свободный – пересказ материала своими словами;
- 2) смешанный – пересказ материала, дополненный словами автора;
- 3) текстуальный – состоит из цитат из текста, выражающих основную мысль и передающих главную ценность написанного;
- 4) плановый – составляется на основе плана.

Конспекты составляются в **письменном виде** в отдельной тетради. Объем каждого конспекта не менее двух страниц.

Памятка по составлению конспекта:

1. Читая изучаемый материал в первый раз, разделите его на основные смысловые части, выделите главные мысли, сделайте выводы.
2. Наиболее значимые положения (тезисы) последовательно и кратко изложите в тетради своими словами или приведите в виде цитат.
3. В конспект включите также выводы, конкретные факты и примеры (без подробного описания).
4. Отдельные слова можно писать сокращенно, допустимо применять условные обозначения.
5. Чтобы конспект был нагляднее и удобнее для проверки преподавателем, **применяйте разнообразные способы подчеркивания, используйте ручки разных цветов.**

3. Методические указания по изучению отдельных вопросов

Самостоятельная работа 1 (6 часов)

Тема «Введение в базы данных»

Цели:

- формирование умений и навыков самостоятельного умственного труда;
- развитие интереса у студентов к изучаемому материалу, привлечение внимания каждого студента к тому, что объясняет преподаватель;
- воспитание профессионализма и активизации учебной деятельности студентов.

Вопросы для самостоятельного изучения:

1. Непосредственное управление данными во внешней памяти
2. Управление буферами оперативной памяти
3. Управление транзакциями
4. Журнализация
5. Поддержка языков БД

Изучив эту тему, вы узнаете, что:

Эта функция включает обеспечение необходимых структур внешней памяти как для хранения данных, непосредственно входящих в БД, так и для служебных целей, например, для убыстрения доступа к данным в некоторых случаях (обычно для этого используются индексы). В некоторых реализациях СУБД активно используются возможности существующих файловых систем, в других работа производится вплоть до уровня устройств внешней памяти. Но подчеркнем, что в развитых СУБД пользователи в любом случае не обязаны знать, использует ли СУБД файловую систему, и если использует, то как организованы файлы. В частности, СУБД поддерживает собственную систему именования объектов БД.

2. Управление буферами оперативной памяти

СУБД обычно работают с БД значительного размера; по крайней мере этот размер обычно существенно больше доступного объема оперативной памяти. Понятно, что если при обращении к любому элементу данных будет производиться обмен с внешней памятью, то вся система будет работать со скоростью устройства внешней памяти. Практически единственным способом реального увеличения этой скорости является буферизация данных в оперативной памяти. При этом, даже если операционная система производит общесистемную буферизацию (как в случае ОС

UNIX), этого недостаточно для целей СУБД, которая располагает гораздо большей информацией о полезности буферизации той или иной части БД. Поэтому в развитых СУБД поддерживается собственный набор буферов оперативной памяти с собственной дисциплиной замены буферов.

Заметим, что существует отдельное направление СУБД, которое ориентировано на постоянное присутствие в оперативной памяти всей БД. Это направление основывается на предположении, что в будущем объем оперативной памяти компьютеров будет настолько велик, что позволит не беспокоиться о буферизации. Пока эти работы находятся в стадии исследований.

3. Управление транзакциями

Транзакция - это последовательность операций над БД, рассматриваемых СУБД как единое целое. Либо транзакция успешно выполняется, и СУБД фиксирует (COMMIT) изменения БД, произведенные этой транзакцией, во внешней памяти, либо ни одно из этих изменений никак не отражается на состоянии БД. Понятие транзакции необходимо для поддержания логической целостности БД. Если вспомнить наш пример информационной системы с файлами СОТРУДНИКИ и ОТДЕЛЫ, то единственным способом не нарушить целостность БД при выполнении операции приема на работу нового сотрудника является объединение элементарных операций над файлами СОТРУДНИКИ и ОТДЕЛЫ в одну транзакцию. Таким образом, поддержание механизма транзакций является обязательным условием даже однопользовательских СУБД (если, конечно, такая система заслуживает названия СУБД). Но понятие транзакции гораздо более важно в многопользовательских СУБД.

То свойство, что каждая транзакция начинается при целостном состоянии БД и оставляет это состояние целостным после своего завершения, делает очень удобным использование понятия транзакции как единицы активности пользователя по отношению к БД. При соответствующем управлении параллельно выполняющимися транзакциями со стороны СУБД каждый из пользователей может в принципе ощущать себя единственным пользователем СУБД (на самом деле, это несколько идеализированное представление, поскольку в некоторых случаях пользователи многопользовательских СУБД могут ощутить присутствие своих коллег).

С управлением транзакциями в многопользовательской СУБД связаны важные понятия *сериализации транзакций* и *сериального плана выполнения смеси транзакций*. Под сериализацией параллельно выполняющихся транзакций понимается такой порядок планирования их работы, при котором суммарный эффект смеси транзакций эквивалентен эффекту их некоторого последовательного выполнения. Сериальный план выполнения смеси транзакций - это такой план, который приводит к сериализации транзакций. Понятно, что если удастся добиться действительно сериального выполнения смеси транзакций, то для каждого пользователя, по инициативе которого образована транзакция, присутствие других

транзакций будет незаметно (если не считать некоторого замедления работы по сравнению с однопользовательским режимом).

Существует несколько базовых алгоритмов сериализации транзакций. В централизованных СУБД наиболее распространены алгоритмы, основанные на синхронизационных захватах объектов БД. При использовании любого алгоритма сериализации возможны ситуации конфликтов между двумя или более транзакциями по доступу к объектам БД. В этом случае для поддержания сериализации необходимо выполнить откат (ликвидировать все изменения, произведенные в БД) одной или более транзакций. Это один из случаев, когда пользователь многопользовательской СУБД может реально (и достаточно неприятно) ощутить присутствие в системе транзакций других пользователей.

4. Журнализация

Одним из основных требований к СУБД является надежность хранения данных во внешней памяти. Под надежностью хранения понимается то, что СУБД должна быть в состоянии восстановить последнее согласованное состояние БД после любого аппаратного или программного сбоя. Обычно рассматриваются два возможных вида аппаратных сбоев: так называемые мягкие сбои, которые можно трактовать как внезапную остановку работы компьютера (например, аварийное выключение питания), и жесткие сбои, характеризующиеся потерей информации на носителях внешней памяти. Примерами программных сбоев могут быть: аварийное завершение работы СУБД (по причине ошибки в программе или в результате некоторого аппаратного сбоя) или аварийное завершение пользовательской программы, в результате чего некоторая транзакция остается незавершенной. Первую ситуацию можно рассматривать как особый вид мягкого аппаратного сбоя; при возникновении последней требуется ликвидировать последствия только одной транзакции.

Понятно, что в любом случае для восстановления БД нужно располагать некоторой дополнительной информацией. Другими словами, поддержание надежности хранения данных в БД требует избыточности хранения данных, причем та часть данных, которая используется для восстановления, должна храниться особо надежно. Наиболее распространенным методом поддержания такой избыточной информации является ведение журнала изменений БД.

Журнал - это особая часть БД, недоступная пользователям СУБД и поддерживаемая с особой тщательностью (иногда поддерживаются две копии журнала, располагаемые на разных физических дисках), в которую поступают записи обо всех изменениях основной части БД. В разных СУБД изменения БД журналируются на разных уровнях: иногда запись в журнале соответствует некоторой логической операции изменения БД (например, операции удаления строки из таблицы реляционной БД), иногда - минимальной внутренней операции модификации страницы внешней памяти; в некоторых системах одновременно используются оба подхода.

Во всех случаях придерживаются стратегии "упреждающей" записи в журнал (так называемого протокола Write Ahead Log - WAL). Грубо говоря, эта стратегия

заключается в том, что запись об изменении любого объекта БД должна попасть во внешнюю память журнала раньше, чем измененный объект попадет во внешнюю память основной части БД. Известно, что если в СУБД корректно соблюдается протокол WAL, то с помощью журнала можно решить все проблемы восстановления БД после любого сбоя.

Самая простая ситуация восстановления - индивидуальный откат транзакции. Строго говоря, для этого не требуется общесистемный журнал изменений БД. Достаточно для каждой транзакции поддерживать локальный журнал операций модификации БД, выполненных в этой транзакции, и производить откат транзакции путем выполнения обратных операций, следуя от конца локального журнала. В некоторых СУБД так и делают, но в большинстве систем локальные журналы не поддерживают, а индивидуальный откат транзакции выполняют по общесистемному журналу, для чего все записи от одной транзакции связывают обратным списком (от конца к началу).

При мягком сбое во внешней памяти основной части БД могут находиться объекты, модифицированные транзакциями, не закончившимися к моменту сбоя, и могут отсутствовать объекты, модифицированные транзакциями, которые к моменту сбоя успешно завершились (по причине использования буферов оперативной памяти, содержимое которых при мягком сбое пропадает). При соблюдении протокола WAL во внешней памяти журнала должны гарантированно находиться записи, относящиеся к операциям модификации обоих видов объектов. Целью процесса восстановления после мягкого сбоя является состояние внешней памяти основной части БД, которое возникло бы при фиксации во внешней памяти изменений всех завершившихся транзакций и которое не содержало бы никаких следов незаконченных транзакций. Для того, чтобы этого добиться, сначала производят откат незавершенных транзакций (undo), а потом повторно воспроизводят (redo) те операции завершенных транзакций, результаты которых не отображены во внешней памяти. Этот процесс содержит много тонкостей, связанных с общей организацией управления буферами и журналом. Более подробно мы рассмотрим это в соответствующей лекции.

Для восстановления БД после жесткого сбоя используют журнал и архивную копию БД. Грубо говоря, архивная копия - это полная копия БД к моменту начала заполнения журнала (имеется много вариантов более гибкой трактовки смысла архивной копии). Конечно, для нормального восстановления БД после жесткого сбоя необходимо, чтобы журнал не пропал. Как уже отмечалось, к сохранности журнала во внешней памяти в СУБД предъявляются особо повышенные требования. Тогда восстановление БД состоит в том, что исходя из архивной копии по журналу воспроизводится работа всех транзакций, которые закончились к моменту сбоя. В принципе, можно даже воспроизвести работу незавершенных транзакций и продолжить их работу после завершения восстановления. Однако в реальных системах это обычно не делается, поскольку процесс восстановления после жесткого сбоя является достаточно длительным.

5. Поддержка языков БД

Для работы с базами данных используются специальные языки, в целом называемые *языками баз данных*. В ранних СУБД поддерживалось несколько специализированных по своим функциям языков. Чаще всего выделялись два языка - *язык определения схемы БД (SDL - Schema Definition Language)* и *язык манипулирования данными (DML - Data Manipulation Language)*. SDL служил главным образом для определения логической структуры БД, т.е. той структуры БД, какой она представляется пользователям. DML содержал набор операторов манипулирования данными, т.е. операторов, позволяющих заносить данные в БД, удалять, модифицировать или выбирать существующие данные. Мы рассмотрим более подробно языки ранних СУБД в следующей лекции.

В современных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных. Стандартным языком наиболее распространенных в настоящее время реляционных СУБД является язык SQL (Structured Query Language). В нескольких лекциях этого курса язык SQL будет рассматриваться достаточно подробно, а пока мы перечислим основные функции реляционной СУБД, поддерживаемые на "языковом" уровне (т.е. функции, поддерживаемые при реализации интерфейса SQL).

Прежде всего, язык SQL сочетает средства SDL и DML, т.е. позволяет определять схему реляционной БД и манипулировать данными. При этом именование объектов БД (для реляционной БД - именование таблиц и их столбцов) поддерживается на языковом уровне в том смысле, что компилятор языка SQL производит преобразование имен объектов в их внутренние идентификаторы на основании специально поддерживаемых служебных таблиц-каталогов. Внутренняя часть СУБД (ядро) вообще не работает с именами таблиц и их столбцов.

Язык SQL содержит специальные средства определения ограничений целостности БД. Опять же, ограничения целостности хранятся в специальных таблицах-каталогах, и обеспечение контроля целостности БД производится на языковом уровне, т.е. при компиляции операторов модификации БД компилятор SQL на основании имеющихся в БД ограничений целостности генерирует соответствующий программный код.

Специальные операторы языка SQL позволяют определять так называемые представления БД, фактически являющиеся хранимыми в БД запросами (результатом любого запроса к реляционной БД является таблица) с именованными столбцами. Для пользователя представление является такой же таблицей, как любая базовая таблица, хранимая в БД, но с помощью представлений можно ограничить или наоборот расширить видимость БД для конкретного пользователя. Поддержание представлений производится также на языковом уровне.

Наконец, авторизация доступа к объектам БД производится также на основе специального набора операторов SQL. Идея состоит в том, что для выполнения операторов SQL разного вида пользователь должен обладать различными полномочиями. Пользователь, создавший таблицу БД, обладает полным набором

полномочий для работы с этой таблицей. В число этих полномочий входит полномочие на передачу всех или части полномочий другим пользователям, включая полномочие на передачу полномочий. Полномочия пользователей описываются в специальных таблицах-каталогах, контроль полномочий поддерживается на языковом уровне.

Более точное описание возможных реализаций этих функций на основе языка SQL будет приведено в лекциях, посвященных языку SQL и его реализации.

Изучение вопросов самостоятельно

При подготовке к вопросам необходимо акцентировать внимание на следующем:

1. Первый вопрос самостоятельного задания выполняется в рабочей тетради по дисциплине. При его выполнении необходимо отметить, что такое язык SQL.

2. Второй вопрос самостоятельного задания выполняется в рабочей тетради по дисциплине. При его выполнении необходимо ознакомиться с управлением буферами оперативной памяти.