



**Некоммерческое
акционерное
общество**

**АЛМАТИНСКИЙ
УНИВЕРСИТЕТ
ЭНЕРГЕТИКИ И
СВЯЗИ**

Кафедра
«Информационные
системы»

БАЗЫ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Методические указания по выполнению курсовых работ
для студентов специальности 5В070300 - Информационные системы

Алматы 2014

Составители: А.Т. Купарова, К.Б. Альмуратова. Базы данных в информационных системах. Методические указания к выполнению курсовых работ для студентов специальности 5В070300 - Информационные системы. – Алматы: АУЭС, 2014. – 21 с.

Методические указания содержат основные задачи курсовой работы, требования к ее выполнению, рекомендации и указания к выполнению курсовой работы и варианты заданий.

Методические указания предназначены для студентов специальности 5В070300 - «Информационные системы» и могут быть использованы студентами других специальностей при изучении технологий баз данных в системах обработки информации.

Табл. 1, библиогр. – 8 назв.

Рецензент: доцент кафедры РТ А.А.Куликов.

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2014 г.

© НАО «Алматинский университет энергетики и связи», 2014 г.

Введение

База данных (БД) - организованная в соответствии с определёнными правилами и поддерживаемая в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области (банк, библиотека, магазин, биржа и т.д.) и используемая для удовлетворения информационных потребностей пользователей. БД хранится и обрабатывается в вычислительной системе. Данные в БД логически структурированы (систематизированы) с целью обеспечения возможности их эффективного поиска и обработки в вычислительной системе. Сегодня БД можно встретить практически везде. Их используют в медицине, на транспорте, в правоохранительных органах, в городских справочных службах, на производстве и в учебных заведениях. БД могут содержать в себе различную информацию, получить которую можно в считанные секунды, нажав для этого всего лишь несколько клавиш на клавиатуре компьютера.

Для создания и использования БД служат системы управления базами данных (СУБД), которые занимают особое место в мире программного обеспечения и нашей повседневной жизни. СУБД обеспечивают реализацию новых концепций в организации информационных служб через создание информационных систем на основе технологии БД. В настоящее время широко применяются муниципальные, банковские биржевые информационные системы, информационные системы оптовой и розничной торговли, торговых домов, служб управления трудом и занятостью, справочной и аналитико-прогнозной котировочной информации и др. Как правило, работа этих систем осуществляется в локальных вычислительных сетях различной архитектуры или их объединениях, получивших название корпоративных сетей дальнейшая интеграция которых возможна с помощью глобальной сети Интернет.

подавляющее большинство современных СБД представляют собой системы реляционного типа, т.е. использующие реляционную модель данных. Данные в реляционных БД хранятся в таблицах – отношениях (relation). Реляционные СБД (РСБД) - это компьютеризованные системы хранения записей в табличном виде. Под БД в различных РСБД понимается табличное хранение данных, но название «база данных» может объединять не только таблицы, но и производные этих таблиц (в виде отчетов, форм, виртуальных таблиц - представлений), формы запросов, программные модули и т.д. СУБД, поддерживающие реляционную модель данных, называются реляционными СУБД (РСУБД). Стандартным языком взаимодействия с реляционными БД является язык запросов SQL, который реализуется в РСУБД на основе операций реляционной алгебры и реляционного исчисления.

Основной целью дисциплин, связанных с изучением БД («Системы баз данных», «Базы данных», «Системы управления базами данных») является получение знаний, умений и практических навыков проектирования систем баз данных и построения приложений баз данных.

Основной целью курсовой работы является приобретение практических навыков по разработке баз данных, программной реализации приложений БД и методов защиты БД для определенной предметной области на основе конкретной СУБД (Visual, FoxPro, MS Access, MS SQL, Server, My SQL, Informix и т.д.).

Для выполнения курсовой работы необходимо знание принципов и методов проектирования баз данных, языковых средств СУБД, принципов реализации приложений баз данных и наличие навыков работы в среде конкретной СУБД.

1 Основные задачи курсовой работы и требования к ее выполнению

1.1 Содержание курсовой работы

В курсовой работе должна быть разработана база данных и реализовано приложение БД с использованием методов защиты БД в среде конкретной СУБД для определенной предметной области (ПО). Для выполнения этой задачи необходимо выполнить: анализ ПО; определить функции, подлежащие реализации в системе; выделить параметры ПО, необходимые для выполнения индивидуального задания; выбрать метод защиты БД. На основе проведенного анализа осуществляется постановка задачи, разработка информационной и даталогической моделей ПО, алгоритмов решения задачи, их реализация.

Курсовая работа состоит из трех разделов: разработки информационной модели и проектирования базы данных; разработки приложения БД с использованием методов и средств защиты; разработки инструкций для работы с БД:

Рекомендуемая последовательность этапов по выполнению курсовой работы:

1) изучение и анализ предметной области, определение форм входных и выходных документов, определение основных объектов ПО (не менее двух), их существенных свойств (не менее четырех) и взаимосвязей между объектами;

2) разработка информационной модели предметной области, представление выбранных объектов, свойств, взаимосвязей в терминах реляционной модели (не менее четырех) и взаимосвязей между объектами;

3) выбор СУБД и разработка даталогической модели предметной области, создание базы данных в выбранной СУБД. Обязательным является использование основных типов полей: символьного, числового, логического и типа даты;

4) загрузка базы данных (не менее 15 записей в каждой таблице);

5) разработка методов и средств защиты БД (вход в систему по паролю, хранение пароля в зашифрованном виде, хранение данных в зашифрованном виде);

- б) разработка функциональной структуры приложения, обеспечивающего работу с базой данных;
- 7) разработка алгоритмов решения задач каждой функции приложения;
- 8) реализация приложения в среде выбранной СУБД;
- 9) разработка инструкций пользователя (конечного пользователя, программиста, администратора БД).

Выполнение курсовой работы завершается оформлением работы и вычерчиванием на отдельных листах структуры базы данных и приложения, а также граф-схем алгоритмов решения задач приложения, алгоритмов шифрования.

1.2 Оформление и защита курсового проекта

Курсовой проект оформляется объемом 25-30 страниц на белой бумаге (текст печатается с одной стороны) формата А4 (210*297), к которой прикладываются листинг (исходной текст) программы, распечатки исходных кодов экранных форм, шаблонов отчетов, а также функциональная структура разработанной системы и структура БД, граф-схемы алгоритмов (при использовании форматов А3, А4).

Основная и графическая часть курсовой работы должны быть оформлены в соответствии с требованиями государственных стандартов.

Для сплошного текста по всем четырём сторонам листа остаются поля (левое-не менее 30 мм, правое – не менее 10 мм, верхнее-не менее 20 мм, нижнее-не менее 20 мм), используется шрифт Times New Roman №12 с полуторным междустрочным интервалом (Times New Roman №14 с одинарным междустрочным интервалом).

Текст основной части делится на разделы, разделы делятся на подразделы. Разделы должны иметь порядковые номера в пределах всей работы, обозначаемые арабскими цифрами, после номера раздела точка не ставится. Наименование разделов записывают прописными буквами с абзацного отступа, в конце наименования раздела точка не ставится. Каждый раздел рекомендуется начинать с нового листа.

Подразделы должны иметь нумерацию в пределах каждого раздела. Номер подраздела состоит из номера раздела и подраздела, разделенных точкой, после номера подраздела точка не ставится. Наименование подраздела оформляется как заголовок строчными буквами, первая буква - прописная. В конце наименования подраздела точка не ставится.

Основная часть должна включать:

- 1) титульный лист с темой курсовой работы, указанием министерства, ВУЗа, кафедры, дисциплины, руководителя и его должности, номера группа и фамилии студента, даты выполнения работы;
- 2) «ЗАДАНИЕ», которое содержит номер варианта и его расшифровку;

3) «СОДЕРЖАНИЕ», которое включает номера и наименования разделов, подразделов, а также таких элементов, как задание, введение, заключение, список литературы и приложение с указанием номеров листов (страниц), на которых они размещены:

4) «ВВЕДЕНИЕ», в котором кратко характеризуется современное состояние вопросов автоматизации обработки информации в той ПО, которой посвящена работа, описывается постановка задачи и цель работы, инструментальная среда реализации;

5) раздел первый – «РАЗРАБОТКА ИНФОРМАЦИОННОЙ МОДЕЛИ И ПРОЕКТИРОВАНИЕ БД», в котором описываются формы входных и выходных документов, информационная модель ПО, выбор СУБД, структура БД и структуры таблиц, приведены рисунки (или распечатки) структуры БД и таблиц, их содержимое;

6) раздел второй – «РАЗРАБОТКА ПРИКЛАДНОЙ ПРОГРАММЫ», в котором приводится и описывается функциональная структура разработанного приложения, описываются алгоритмы решения задач каждой функции системы и реализация приложения в среде конкретной СУБД (проектирование меню, форм и отчетов с указанием используемых объектов, их свойств и с ссылкой на приложение А), описывается алгоритм и программная реализация защиты БД;

7) раздел третий – «ИНСТРУКЦИИ», в которых описываются инструкции для конечного пользователя, программиста и администратора БД, приводится интерфейс пользователя;

8) «ЗАКЛЮЧЕНИЕ» с оценкой разработанных БД и приложения в целом с точки зрения функционального соответствия потребностям конечного пользователя и дружелюбности интерфейса, с предложениями по расширению функциональных возможностей приложения и обеспечения большей комфортности при работе с ним;

9) «СПИСОК ЛИТЕРАТУРЫ», где источники располагаются в порядке появления ссылок на них в тексте (ссылки по тексту на источники обязательны);

10) «ПРИЛОЖЕНИЕ А», в которое включаются тексты программ, распечатки исходных кодов экранных форм, шаблоны отчетов и других компонентов приложения;

11) «ПРИЛОЖЕНИЕ Б», в которое включается структура разработанной системы и БД, граф - схемы алгоритмов (при использовании форматов А3, А4).

При этом все листы (страницы) должны иметь сквозную нумерацию, начиная с титульного листа и включая приложение. Номер листа пишется в правом верхнем углу без точки (на титульном листе номер страницы не проставляется). Наименования, включённые в содержание, записывают строчными буквами (кроме первой). Слова «СОДЕРЖАНИЕ», «ВВЕДЕНИЕ», «ЗАКЛЮЧЕНИЕ», «СПИСОК ЛИТЕРАТУРЫ», записываются в виде заголовка прописными буквами и не нумеруются. Каждое приложение

начинается с нового листа. Если их более одного, то приложения обозначают заглавными буквами русского алфавита. После слова «Приложение» следует буква, обозначающая его последовательность. Слово «Приложение» пишут строчными буквами с первой прописной и надпись располагают наверху посередине страницы. Если приложение располагается на нескольких листах, то на каждом последующем листе необходимо писать «Продолжение приложения А».

Курсовая работа может включать таблицы и рисунки (графики, схемы, диаграммы и т. п.), которые нумеруются (если их в тексте больше одного) в пределах документа.

Слово «Таблица» записывается в левом углу с абзацного отступа с прописной буквы, после номера таблицы ставится дефис, после которого с прописной буквы идет название таблицы. Слово «Рисунок» записывается под рисунком с абзацного отступа с прописной буквы, после номера рисунка ставится дефис, после которого с прописной буквы идет название рисунка. После названия таблицы (рисунка) точка не ставится.

Допускается нумерация таблиц (рисунков) в пределах раздела. В этом случае номер таблицы (рисунка) состоит из номера раздела и порядкового номера таблицы (рисунка), разделенных точкой. На все таблицы (рисунки) должны быть приведены ссылки в тексте документа.

К защите допускаются работы, выполненные в соответствии с приведёнными выше требованиями, после проверки преподавателем. На защите студент должен показать знание основных определений и понятий, используемых в работе, уметь обосновать выбор принятого решения, а также знать в подробностях все разработанные вопросы.

Защита курсовой работы проводится с демонстрацией работы разработанного приложения на компьютере.

2 Рекомендации к выполнению курсовой работы

2.1 Разработка информационной модели и проектирование базы данных

Выполняется анализ предметной области, выбранной по таблице 1 в соответствии с номером по списку в группе или в соответствии с профилем работы. Определяются объекты, которые должны интересовать конечного пользователя. В случае затруднения определения объектов используются объекты, указанные в таблице 1 (по два объекта для каждой предметной области). Количество объектов должно быть не меньше двух. Определяются атрибуты (существенные свойства) объектов, интересующие пользователя и разрабатывается информационная модель решаемой задачи в виде объектов и взаимосвязей типа «один-к-одному», «один-ко-многим», «многие-ко-многим». Тип взаимосвязи определяется связями между объектами, существующими в реальном мире.

Для каждого объекта определяется отношение (в виде таблицы) с учетом необходимых атрибутов. Необходимо предусмотреть атрибуты, играющие роль первичных ключей и внешних ключей для установления взаимосвязей между отношениями. При наличии взаимосвязи «многие-ко-многим» необходимо ввести промежуточное отношение для промежуточного объекта, которое будет содержать идентификаторы связываемых объектов. Присутствие промежуточного объекта необходимо по причине невозможности установления в информационной модели между двумя объектами связи «многие-ко-многим» в большинстве реляционных СУБД.

Выбирается СУБД, и в соответствии с требованиями выбранной СУБД, создаются таблицы и определяются параметры полей (не менее четырех в каждой таблице). Обязательным является использование основных типов полей: символьного, числового, логического (или денежного), поля типа даты (или дата/время).

Выполняется загрузка БД путем внесения в таблицы записей (не менее 15 в каждой), устанавливаются между таблицами взаимосвязи в соответствии с информационной моделью.

Таблица 1 - Варианты тем курсовой работы

Вариант	Предметная область	Объект 1	Объект 2
1	Автобусные перевозки	Расписание рейсов	Пассажиры
2	Морские перевозки	Расписание рейсов	Пассажиры
3	Воздушные перевозки	Расписание рейсов	Пассажиры
4	Железнодорожные перевозки	Расписание рейсов	Пассажиры
5	Больница	Врачи	Пациенты
6	Поликлиника	Врачи	Пациенты
7	Библиотека	Книги	Читатели
8	Аптека	Поставщики	Медикаменты
9	Деканат	Студенты	Учебный паспорт
10	Выпуск изделий	Цех	Продукция
11	Склад	Поставщики	Товар
12	Магазин	Поставщики	Товар
13	Гостиница	Номер в гостинице	Проживающие
14	Спортивные соревнования	Расписание соревнований	Спортсмены
15	Автоинспекция	Водители	Автомобили
16	Ресторан	Меню	Счета
17	Книгообмен	Карточка предложений	Карточка спроса
18	Обмен квартир	Карточка предложений	Карточка спроса

Окончание таблицы 1

Вариант	Предметная область	Объект 1	Объект 2
19	Биржа труда	Карточка предложений	Карточка спроса
20	Кинотеатр	Расписание сеансов	Посетители

Возможен выбор предметной области, отсутствующей в таблице 1, по согласованию с преподавателем.

2.2 Разработка прикладной программы

При разработке приложения БД необходимо реализовать защиту БД, обеспечивающую шифрование БД, контроль доступа (вход в систему по паролю) и т.д.

Функции приложения определяются, исходя из требований конечного пользователя. Приложение должно обеспечивать следующие функции:

- работу с БД (добавление, корректировку и удаление записей в БД);
- обслуживание клиентов: покупателей товара или билетов (на транспорт, на сеанс кино), читателей, пациентов врача, владельцев автомашин, проживающих в гостинице и т.д. Например: продажа билетов, возврат билетов и выдача на печать бланка билета; выдача и прием книг, печать формуляра читателя; регистрация поступления и продажи товара, печать чека; оформление заказа, печать квитанций заказа, выдача заказа; регистрация автомобиля, снятие автомобиля с учета, печать справки о регистрации; прием заказа в ресторане, аннулирование заказа, печать счета; прием заказа на бронь в гостинице, снятие брони, заселение, выписка проживающего, печать счета; прием на работу, увольнение с работы, печать справки о причине увольнения и т.д.;
- получение справочной информации по БД (не менее трех типов справки). Например, наличие свободных мест в гостинице, билетов на указанный рейс или сеанс кино; наличие рейса по указанному маршруту; наличие указанного товара, указанного блюда, указанной книги; адрес указанного поставщика, владельца автомашины, заказчика, пациента, врача и т.д.;
- получение отчетов, в т.ч. и со статистическими данными (не менее трех типов) по данным базы данных. Например, количество проданных билетов за день; сумма полученных денег от продажи билетов на указанный рейс; количество пациентов с указанным диагнозом; средний возраст больного, врача, читателя и т.д.;
- окончание работы прикладной программы и выход в среду используемой СУБД или ОС.

Также при разработке приложения обязательно использование стандартного языка баз данных SQL и визуального инструментария для создания удобного пользовательского интерфейса.

2.3 Разработка инструкций для работы с базой данных

Разрабатываются инструкции пользователя, сопровождающего программиста и администратора БД.

Инструкции пользователя должны содержать инструкции по работе с приложением для конечного пользователя с иллюстрациями в виде форм интерфейса.

Инструкции для сопровождающего программиста должны содержать описание необходимых технических и программных средств, описание файлов системы, структуры их хранения.

Инструкции для администратора БД должны содержать указания о способе хранения и изменения пароля для входа в систему, методах и средствах администрирования БД (при использовании сервера БД).

Заключение

Выполнение курсовой работы по дисциплинам, связанных с изучением технологии баз данных («Системы баз данных», «Базы данных», «Системы управления базами данных») позволит студентам приобрести практические навыки по разработке БД и построению приложений БД с использованием методов защиты и безопасности для определенной предметной области на основе конкретной СУБД.

В процессе выполнения курсовой работы студентами приобретаются необходимые практические знания принципов и методов проектирования баз данных, языковых средств СУБД, принципов проектирования приложений баз данных, навыки работы в среде конкретной СУБД.

Знания и навыки, полученные в процессе выполнения курсовой работы, в дальнейшем могут быть использованы при изучении дисциплин, связанных с разработкой различных систем на основе технологии баз данных, а также при выполнении дипломного проектирования.

Приложение А

Стандартный язык баз данных SQL

Команда выборки данных – SELECT:

```
SELECT [DISTINCT] [<псевдоним>.] <выражение> [AS] <колонка>][,  
[<псевдоним>]<выражение> [AS<колонка>]..  
FROM <имя_таблицы1>[<псевдоним1>][,<имя_таблицы2>  
[<псевдоним2>].]  
[[INTO <получатель>]/[TO FILE <имя_файла> [ADDITIVE]/TO  
PRINTER]]  
[NOCONSOLE] [PLAIN] [NOWAIT]  
[WHERE <условие_связи> [AND <условие_связи>]  
[AND/OR <условие_связи> ]]  
[GROUP BY <колонка> [, <колонка>...]] [HAVING <условие_отбора>]  
[ORDER BY <колонка> [ASC/DESC] [, <колонка> [ASC/DESC]...]]
```

Включение опции DISTINCT исключает возможность вывода одинаковых строк в выборке. Перед словом FROM перечисляются отбираемые <выражения>, а после – имена таблиц, из которых берутся данные. <Выражение> может быть полем записи из таблицы, константой (выводимой в каждой строке выборки), функцией от переменных (полей) и т.п. Если <выражение> является именем поля, то оно может быть составным (с включением имени таблицы данных или псевдонима), в особенности если выборка делается из нескольких таблиц, где имена полей совпадают. Псевдонимом может быть не только официальный псевдоним (ALIAS) таблицы данных, но и любое другое имя, которое присваивается в команде SELECT. Это задаваемое временное имя указывается в опции <псевдоним> после слова FROM за именем таблицы и используется только в данной команде SELECT в других ее опциях (локальный псевдоним). Если необходимо построить выборку из всех полей таблицы данных, вместо их перечня можно указать символ *. В результате выполненной выборки получается совокупность колонок, заголовками которых могут быть имена исходных полей.

Например, для вывода всех записей из таблицы STUD используется команда:

```
SELECT * FROM STUD
```

Команду SELECT можно использовать для одновременной выборки данных из нескольких таблиц. Если имена полей, выбираемых из разных таблиц совпадают, то такие колонки получают совпадающие имена, к которым присоединяется одна из букв (по алфавиту), например, FAM_A, FAM_B и т.д. Аналогичным образом даются имена колонкам, полученным в результате вычисления выражений. Их имена состоят из слова EXP и последовательных чисел (EXP_1, EXP_2 и т.д.). Исключения составляют выражения, использующие соответственные функции SQL: AVG, MIN, MAX,

SUM, COUNT. Последняя функция может иметь в качестве аргумента звездочку (COUNT(*)), что означает подсчет всех записей, попавших в выборку. Имена колонок в этом случае будут включать имена функций. Вместо имен, формируемых по умолчанию, можно назначить колонкам другие имена, указав их после слова AS в виде <выражение> AS <новое_имя_колонки>.

Например, для вывода минимального, максимального и среднего значений зарплаты (поле ZARP) из таблицы KADR необходимо использовать команду:

```
SELECT MIN (ZARP), MAX(ZARP), AVG(ZARP)
FROM KADR
```

Для вывода из таблицы KADR фамилий (поле FAM) и табельных номеров (поле TAB) с другими именами колонок (Фамилии и Таб_номер) используется команда:

```
SELECT FAM AS Фамилии, TAB AS Таб_номер
FROM KADR
```

Для вывода всех фамилий из двух таблиц данных KADR и STUD с использованием локальных псевдонимов P и T, а также с заменой при выводе имен колонок FAM_A и FAM_B KFIO и STFIO соответственно, используется команда:

```
SELECT P.FAM AS KFIO, T.FAM AS STFIO
FROM KADR P, STUD T
```

В команде SELECT можно задать достаточно сложные условия для выборки данных в запрос. Условие связи применяется в случае, если выборка делается более, чем из одной таблицы данных, и определяет критерий объединения данных из разных таблиц. В условии связи указываются поля из разных таблиц с псевдонимами и используются знаки отношений =, #, =, >=, <, <=. Допускается задание нескольких критериев, соединенных знаком AND. Условие выборки строится аналогично, но из выражений только для одной таблицы, и допускается использование логических операторов OR, AND и NOT.

Условия могут содержать следующие операторы SQL: LIKE, BETWEEN, IN. Эти операторы можно комбинировать с помощью связок OR, AND, NOT и скобок.

Оператор LIKE позволяет построить условие сравнения по шаблону, где символ "_" указывает единичный неопределенный символ в строке, а символ "%" – любое их количество. Формат оператора LIKE:

```
<выражение> LIKE <шаблон>.
```

Оператор BETWEEN задает начальное и конечное значение диапазона и проверяет, находится ли выражение, стоящее слева от оператора, в указанном диапазоне. Формат оператора BETWEEN:

```
<выражение> BETWEEN <нижнее значение> AND <верхнее значение>.
```

Оператор IN проверяет, находится ли выражение, стоящее слева от слова IN, среди перечисленных справа от него. Формат оператора IN:

<выражение> IN (<выражение1>, <выражение2>,...).

Например, для выборки фамилий (поле FAM), начинающихся на букву "А", всех мужчин (поле POL="М") из таблицы KADR. DBF используется команда:

```
SELECT FAM FROM KADR WHERE POL="М" AND  
FAM LIKE "А"
```

Для вывода из таблицы STUD фамилий студентов (поле FAM), получающих стипендии (поле STIP) от 1500 до 2000 тг., используется команда:

```
SELECT FAM FROM STUD WHERE STIP  
BETWEEN 1500 AND 2000
```

Для вывода из таблицы KADR фамилий сотрудников (поле FAM), а из таблицы TABEL – соответствующего количества рабочих дней (поле WD) для записей, у которых совпадают табельные номера, необходимо использовать команду с условием связи (в качестве псевдонимов таблиц указаны имена таблиц):

```
SELECT KADR.FAM, TABEL.WD FROM KADR, TABEL WHERE  
KADR.TAB=TABEL.TAB
```

Для упорядочения по заданной колонке или колонкам используется опция ORDER BY. По умолчанию сортировка выполняется по возрастанию (ASC), но может быть задана и по убыванию (DESC). Указание колонки упорядочения может выполняться именем или номером колонки. Например, для вывода фамилий (поле FAM) и группы (поле GRUPPA), из таблицы данных STUD в порядке возрастания групп и фамилий необходимо использовать команду:

```
SELECT FAM, GRUPPA FROM STUD  
ORDER BY GRUPPA, FAM
```

В опции ORDER BY обычно нельзя использовать вычисляемые выражения. В случае необходимости упорядочения по колонке с вычисляемыми значениями указывается номер колонки. Например, для вывода фамилий (поле FAM) и премии, равной половине зарплаты (величина премии вычисляется по значению поля ZARP), из таблицы данных KADR в порядке возрастания премии используется команда:

```
SELECT FAM AS Фамилия, ZARP AS Премия  
FROM KADR ORDER BY 2,1
```

В этой команде фамилии сотрудников, имеющих одинаковую премию, выводятся в алфавитном порядке, т.к. упорядочение данных выполняется сначала по второй колонке (премия), затем по первой колонке (фамилия).

Опция GROUP BY команды SELECT позволяет сгруппировать записи с одинаковым значением указанной колонки (или колонок). Опция GROUP BY задает колонки, по которым производится группирование выходных данных. Все записи таблицы, для которых значения колонок совпадают, отображаются в выборке единственной строкой. Группирование удобно для получения

некоторых сводных характеристик группы (суммы, среднего значения, количества записей в группе и т.д.).

Опция **HAVING** <условие_отбора> задает критерий отбора данных в каждую сформированную в процессе выборки группу, т.е. выполняет роль опции **WHERE**, но для группируемых данных.

Например, для вывода названий всех групп (поле **GRUPPA**) количества студентов и суммы стипендий (поле **STIP**) для каждой группы из таблицы данных **STUD** используется команда:

```
SELECT GRUPPA, COUNT(FAM), SUM(STIP)
FROM STUD GROUP BY GRUPPA
```

Внутри групп можно создавать подгруппы. При группировании данных в опции **HAVING** можно использовать агрегатные функции **SQL**. Например, для вывода названий всех лабораторий (поле **LAB**), количества сотрудников и значения суммарной зарплаты (поле **ZARP**) из таблицы данных **KADR** используется следующая команда (информация выводится только для лабораторий, где количество сотрудников больше 5):

```
SELECT LAB, SUM(ZARP), COUNT(*) FROM KADR
GROUP BY LAB HAVING COUNT(*)>5
```

Для указания объекта получателя данных выборки используется опция **INTO** или **TO**:

```
SELECT <выражение> FROM <имя_таблицы>
[INTO TABLE <имя_таблицы>] / [INTO CURSOR <имя_курсора>] /
[INTO ARRAY <имя_массива>] /
[TO FILE <имя_файла> [ADDITIVE]] /
[TO PRINTER] [NOCONSOLE] [PLAIN] [NOWAIT]
```

Типы возможных получателей данных:

TABLE <имя_таблицы> - получателем является новая таблица с указанным именем.

CURSOR <имя_курсора> - результат запроса помещается в курсор с указанием именем. Курсор – это временный набор данных, который может быть областью памяти или временным файлом и имеет режим "только чтение". Данные курсора могут быть, например, предъявлены в команде **BROWSE**, напечатаны, из них может образовано меню и т.д. Курсор может быть обработан другой командой **SELECT**. К колонкам курсора надо обращаться по имени этих колонок с префиксом – именем курсора (через точку).

ARRAY <имя_массива> - в качестве получателя результата запроса будет использован новый двумерный массив с указанным именем.

Кроме того, данные выборки можно переслать в файл или на принтер. Для этого в команде указывается получатель **TO FILE** <имя_файла> **[ADDITIVE]** / **TO PRINTER** и выборка посылается в текстовый файл с указанным именем или на принтер. Если используется слово **ADDITIVE**, то выборка будет добавлена в конец существующего файла без его перезаписи.

Следующие опции имеют смысл только при выдаче на экран (команда используется без опций INTO или TO):

NOCOSOLE – выборка не выдается на экран;

PLAIN – заголовки колонок не выдаются;

NOWAIT – не делается пауза при заполнении экрана.

Использование подзапросов в условиях команды SELECT позволяет создавать сложные запросы. Подзапрос заключается в круглые скобки и представляет собой вложенную команду SELECT, вложение подзапросов неограниченно. Причем, основной запрос и используемый в нем подзапрос могут обращаться как к одним и тем же таблицам, так и к разным таблицам баз данных. В подзапросе нельзя использовать опции ORDER BY и INTO.

Запрос с подзапросом может быть не коррелирован или коррелирован. Некоррелированный подзапрос выполняется в первую очередь, затем полученный результат подставляется в условие и выполняется внешний запрос. Например, для выборки фамилий сотрудников (поле FAM) из таблицы KADR с заработной платой (поле ZARP) выше средней можно использовать запрос с подзапросом (для таблицы KADR используется локальный псевдоним S1):

```
SELECT S1.FAM FROM FROM KADR S1 WHERE ZARP >
(SELECT AVG(ZARP) FROM KADR)
```

В коррелированном подзапросе внутренний запрос ссылается на внешний запрос и выполняется поочередно для каждой строки внешнего запроса (многократно). Выбирается первая строка во внешнем запросе, для нее выполняется внутренний запрос, затем вторая строка и т.д. Например, для выборки фамилий сотрудников (поле FAM) и зарплаты (поле ZARP) из таблицы KADR с заработной платой выше средней по каждой лаборатории можно использовать запрос с подзапросом (для таблицы KADR используются локальные псевдонимы S1 во внешнем запросе и S2 в подзапросе для обеспечения сравнения):

```
SELECT FAM, ZARP FROM KADR S1 WHERE ZARP >=
(SELECT AVG (ZARP) FROM KADR S2
WHERE S2.LAB=S1.LAB)
```

Результаты нескольких выборок можно объединить в одном запросе, используя оператор объединения UNION. Результатом будет множество, состоящее из всех строк, входящих в какую-либо выборку или в нескольких выборок. Но при этом результаты исходных выборок должны иметь одинаковое число полей (столбцов), тип и ширина i-го поля одной выборки должны совпадать с типом и шириной i-го поля любой другой выборки. При использовании опции UNION часто оказывается полезным включение константы в получаемый результат выборки. Заголовки колонок в выборке определяются первым запросом. Например, текстовую константу можно использовать в качестве поясняющего текста при выборе из таблицы STUD фамилий студентов (поле FAM), получающих стипендию (поле STIP) больше

2000 или проживающих в г.Алматы (город проживания указывается в поле ADDRESS):

```
SELECT FAM AS Фамилии, "стипендия больше 2000" AS
Признак_выборки FROM STUD WHERE STIP > 2000
UNION
SELECT FAM, "город Алматы" FROM STUD WHERE ADDRESS LIKE
"%Алматы "
```

Оператором UNION можно соединить любое число команд SELECT, но опция ORDER BY в запросе с использованием оператора UNION может входить только в последнее предложение SELECT. При указании критерия упорядочивания указываются номера полей в получаемой выборке. Например, при выборке из таблицы KADR фамилий сотрудников (поле FAM), имеющих заработную плату (поле ZARP) меньше 10000 или проживающих в г. Алматы (город проживания указывается в поле ADDRESS), можно сначала упорядочить данные по второй колонке (признак выборки), а затем по первой колонке (фамилии в алфавитном порядке):

```
SELECT FAM, "заработная плата меньше 1000"
FROM KADR WHERE ZARP <10000
UNION
SELECT FAM, "город Алматы" FROM KADR
WHERE ADDRESS LIKE "%Алматы" ORDER BY 2,1
```

По умолчанию оператор UNION устраняет из результата повторяющиеся строки. Чтобы отобразить все строки, необходимо использовать оператор UNION с опцией ALL (UNION ALL).

Команда добавления записей в БД – INSERT:

```
INSERT INTO <имя_таблицы> [( <имя_поля1>[, <имя_поля2> ... ])]
VALUES (<выражение1> [, <выражение2>...])
```

Команда добавляет записи в конец существующей таблицы, используя выражения, перечисленные после слова VALUES. Выражения заносятся в указанные поля. Если опущены имена полей, выражения будут записываться в последовательные поля таблицы данных в соответствии с ее структурой.

Например, результат следующих двух команд будет одинаковым:

```
INSERT INTO KADR (FAM, ZARP) VALUES ('Киров',18200)
INSERT INTO KADR (ZARP, FAM) VALUES (18200, 'Киров')
```

В качестве выражений могут использоваться значения переменных или элементов массивов переменных:

```
INSERT INTO <имя_таблицы>
FROM ARRAY <имя_массива>
```

Команда добавляет записи в конец таблицы данных, используя данные, содержащиеся в указанном массиве. Данные из массива заносятся последовательно в поля, начиная с первого. Типы соответствующих полей и элементов массива должны совпадать.

Во многих СУБД (но не во всех) SQL-команду INSERT можно использовать с подзапросом – вложенной командой SELECT, если множество дополняемых данных является результатом запроса:

```
INSERT INTO <имя_таблицы> [(<имя_поля1>[, <имя_поля2> ...]])  
<подзапрос>
```

Например, дополнение в таблицу STUD1 фамилий студентов (поле FAM) из таблицы STUD2 осуществляется следующей командой:

```
INSERT INTO STUD1 (FAM) VALUES  
SELECT DISTINCT FAM FROM STUD2
```

Команда модификации данных в таблице БД – UPDATE:

```
UPDATE <имя_таблицы> SET <имя_поля1>= <выражение1> [,  
<имя_поля2> = <выражение2> .. ] [WHERE <условие>]
```

Замена значений происходит в записях, удовлетворяющих условию, указанному после опции WHERE. Если эта опция отсутствует, то замена значений происходит во всех записях таблицы. Например, для замены фамилии (поле FAM) 'Ахметов' на 'Ахмет' в таблице STUD необходимо использовать следующую команду:

```
UPDATE STUD FAM = 'Ахмет' WHERE FAM = 'Ахметов'
```

При модификации данных можно использовать подзапрос – вложенную команду SELECT для формирования операнда условия:

```
UPDATE <имя_таблицы> SET <имя_поля>=<выражение>  
[WHERE <операнд_условия> <оператор_условия> (<подзапрос>)]
```

Например, установка надбавки к стипендии студентов (поле STIP) на 25% в таблице STUD1, если фамилии этих студентов (поле FAM) имеются в таблице STUD2, реализуется следующей командой:

```
UPDATE STUD1 SET STIP = 1.25*STIP  
WHERE FAM IN (SELECT FAM FROM STUD2)
```

Команда удаления записей – DELETE:

```
DELETE FROM <имя_таблицы> [WHERE <условия>]
```

Удаляются записи, удовлетворяющие условию, указанному после опции WHERE. Если эта опция отсутствует, то удаляются все записи из таблицы.

При удалении можно использовать подзапрос – вложенную команду SELECT для формирования операнда условия:

```
DELETE FROM <имя_таблицы>  
[WHERE <операнд_условия> (<подзапрос>)]
```

Команда DELETE помечает записи на удаление без физического удаления записей. Физическое удаление записей можно реализовать командой PACK. Например, для пометки на удаление данных о сотруднике с табельным номером 65 (поле TAB) из таблицы KADR используется команда:

```
DELETE FROM KADR WHERE TAB = 65
```

Для пометки на удаление всех записей из таблицы KADR необходима команда:

DELETE FROM KADR

Пометка на удаление из таблицы STUD1 данных о студентах, если фамилии этих студентов (поле FAM) имеются в таблице STUD2, реализуется следующей командой:

**DELETE FROM STUD1 WHERE FAM IN
(SELECT FAM FROM STUD2)**

Список литературы

1. Баженова И.Ю. Основы проектирования приложений баз данных.-М.: «Бином», 2013.
2. Сыдықов Б.Д. ж/е т.б. Деректер қоры.- А.: «Нур-Принт», 2012.
3. Ахметова О.С. Базы данных.-А.: «КазНПУ», 2012.
4. Сичкаренко В.А. SQL - 99. Руководство разработчика баз данных.-СПб.: ДиаСофт, 2002.
5. Бидайбеков Е.Ы. Базы данных и информационные системы.-А., 2010.
6. Леонард Л. И др. Разработка приложений на основе Microsoft SQL Server 2008.-М., 2010.
7. Тернстрем Т. Microsoft SQL Server 2008 . Разработка баз данных.-М., 2010.
8. Макленнен Д. SQL Server 2008 Data mining интеллектуальный анализ данных.-СПб.: «БХВ-Петербург», 2009.
9. Вишневский А. SQL Server. Эффективная работа.- Спб.: «Питер», 2009.

Содержание

Введение	3
1 Основные задачи курсовой работы и требования к ее выполнению	4
1.1 Содержание курсовой работы	4
1.2 Оформление и защита курсового проекта	5
2 Рекомендации к выполнению курсовой работы	7
2.1 Разработка информационной модели и проектирование базы данных	7
2.2 Разработка прикладной программы	9
2.3 Разработка инструкций для работы с базой данных	10
Заключение	10
Приложение А	11
Список литературы	19

Айжан Токжумаевна Купарова

Камшат Бимуратовна Альмуратова

БАЗЫ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Методические указания по выполнению курсовых работ
для студентов специальности 5В070300 - Информационные системы

Редактор Л.Т.Сластикова

Специалист по стандартизации Н.К.Молдабекова

Подписано в печать __.__.__.

Тираж 30 экз.

Объем 1,3 уч.-изд. л.

Формат 60x84 1/16

Бумага типографическая №1

Заказ _____. Цена 650 тг.

Копировально-множительное бюро
некоммерческого акционерного общества
Алматинского университета энергетики и связи
050013 Алматы, Байтурсынова, 126