

Я.Е. ПРОКУШЕВ

БАЗЫ ДАННЫХ

*практикум
для студентов, обучающихся
по специальностям и направлениям подготовки
09.00.00 «Информатика и вычислительная техника»,
10.00.00 «Информационная безопасность»,
а также направлению подготовки 38.03.05 «Бизнес-Информатика»*

Санкт-Петербург
ИЦ «Интермедия»
2018

УДК 004.56(075.8)
ББК 32.81я73
П80

Автор:

Прокушев Ярослав Евгеньевич, канд. экон. наук, доцент, заведующий кафедрой организации и технологии защиты информации Белгородского университета кооперации, экономики и права

Главный редактор: Т.С. Кулакова
Техническая подготовка: В.Ю. Антипова
Дизайн обложки: А. Н. Федулова

П80 Прокушев Я.Е.

Базы данных: практикум/ Прокушев Я.Е. – СПб.: ИЦ «Интермедия», 2018. – 240 с. : илл.

В практикуме рассмотрено применение структурированного языка запросов SQL при создании и изменении баз данных и их объектов, а также выполнении операций обработки информации в базах данных. Рассмотрены вопросы создания программ и алгоритмов на языке SQL.

Книга предназначена для студентов, обучающихся по специальностям и направлениям подготовки, относящимся к УГСНП 09.00.00 «Информатика и вычислительная техника», 10.00.00 «Информационная безопасность», а также направления подготовки 38.03.05 «Бизнес-Информатика». Материалы книги могут быть полезны практическим специалистам в области информационных технологий.

ISBN 978-5-4383-0149-3

Учебное издание

Прокушев Ярослав Евгеньевич

Базы данных

Подписано в печать 11.09.2017. Формат 60 × 88 1/16. Печать цифровая
Усл. печ. л. 14,7. Тираж 500 экз. Заказ №
ООО «Издательский центр “Интермедия”». Адрес: 198334, Санкт-Петербург,
ул. Партизана Германа, 41-218. Отпечатано с готового оригинал-макета
в ООО «Арт-экспресс». Адрес: 199155, СПб., В.О., ул. Уральская, д. 17.

ISBN 978-5-4383-0149-3

© ООО «Издательский Центр “Интермедия”», 2018
© Прокушев Я.Е., 2018

СОДЕРЖАНИЕ

Введение.....	4
Лабораторная работа 1. Создание баз данных.....	6
Лабораторная работа 2. Создание таблиц в базах данных.....	32
Лабораторная работа 3. Изменение структуры таблиц.....	42
Лабораторная работа 4. Резервное копирование и восстановление пользовательских баз данных.....	56
Лабораторная работа 5. Однотабличные запросы.....	68
Лабораторная работа 6. Различные типы объединения таблиц.....	75
Лабораторная работа 7. Многотабличные запросы.....	82
Лабораторная работа 8. Итоговые запросы на выборку.....	91
Лабораторная работа 9. Внесение изменений в базу данных.....	99
Лабораторная работа 10. Подчиненные запросы.....	107
Лабораторная работа 11. Использование объектов, обеспечивающих целостность данных.....	114
Лабораторная работа 12. Работа с представлениями.....	125
Лабораторная работа 13. Применение индексов.....	130
Лабораторная работа 14. Транзакции.....	138
Лабораторная работа 15. Блокировки.....	148
Лабораторная работа 16. Работа с алгоритмическими конструкциями в SQL.....	165
Лабораторная работа 17. Использование храняемых процедур.....	178
Лабораторная работа 18. Использование функций.....	193
Лабораторная работа 19. Использование триггеров.....	203
Лабораторная работа 20. Использование курсоров.....	208
Список использованной литературы.....	222
Приложения.....	223

ВВЕДЕНИЕ

В практической деятельности специалисты самых разных направлений в области информационных технологий постоянно работают с базами данных. Это объясняется тем, что в настоящее время весьма значительная часть информации хранится и обрабатывается в структурированном виде – то есть в базах данных.

Поэтому для специалистов, как в области информационных технологий, так и защиты информации, жизненно необходимо знание языка SQL, используемого для создания объектов баз данных и манипулирования обрабатываемой в них информации.

В этой связи изучение баз данных предусматривает рассмотрение не только теоретических, но и прикладных аспектов работы с ними. Данные обстоятельства обусловили содержание практикума, в котором освещаются как теоретические аспекты, так и практические особенности использования команд языка SQL.

Практикум содержит 20 лабораторных работ, каждая из которых начинается с теоретического материала, знакомящего с основными понятиями и синтаксисом рассматриваемых команд.

Затем приводятся практические задания, предназначенные для выполнения и закрепления полученных теоретических знаний. С целью облегчения усвоения материала демонстрируются несколько решенных заданий, а затем приводятся аналогичные им примеры для самостоятельного выполнения. Каждая лабораторная работа заканчивается перечнем вопросов для самоконтроля приобретенных при ее выполнении знаний, умений и навыков.

В качестве среды для выполнения лабораторных работ была выбрана СУБД MS SQL Server, при этом почти весь предлагаемый материал предполагает использование языка SQL, а не графических возможностей утилиты администрирования SQL Server Management Studio.

Поскольку MS SQL Server поддерживает стандарт SQL-3, то после изучения предлагаемого в книге материала, можно будет успешно освоить работу и с другими популярными реляционными СУБД (Oracle, Postgre SQL, MySQL и т.п.).

При изучении материала, приведенного в практикуме, возможно использование дополнительной литературы по языку SQL.

В приложении 1 указан примерный перечень специальностей и направлений подготовки, для которых может быть использован в образовательном процессе предлагаемый вашему вниманию практикум.

Для выполнения лабораторных работ потребуется свободно распространяемая СУБД MS SQL Server Express Edition 2008 R2 SP2\ 2012\ 2014 или более «старшие» ее редакции. В приложении 2 продемонстрированы основные этапы установки СУБД MS SQL Server Express Edition 2014.

Дополнительные материалы к практикуму размещены на сайте издательства «Интермедия»¹.

Все сведения о физических и юридических лицах, имеющиеся в учебных базах данных, являются вымышленными.

Свои пожелания или критические замечания относительно материала книги можно высылать на электронный ящик автора:

yaroslav-prokushev@mail.ru

¹Режим доступа:

http://intermedia-publishing.ru/Prokushev_materials_db.zip

ЛАБОРАТОРНАЯ РАБОТА 5. ОДНОТАБЛИЧНЫЕ ЗАПРОСЫ

Цель работы: получение навыков создания однотоабличных запросов на языке SQL. Изучение способов формирования условий отбора данных.

Теоретическая информация

1. Синтаксис команды *Select*

Язык SQL предназначен для выполнения запросов, обрабатывающих данные. Инструкция *Select*, которая используется для построения SQL-запросов на выборку, является наиболее мощной из всех инструкций SQL. Несмотря на богатство возможностей этой инструкции, ее изучение желательно начать с создания выборки данных из одной таблицы. Упрощенный синтаксис команды *Select* приводится ниже [1; 2]:

```
Select select_list  
[Into new_table] From table_source  
[Where search_condition]  
[Group by group_by_expression]  
[Having search_condition]  
[Union]  
[Order by order_expression [Asc | Desc]]
```

Каждый из разделов (предложений) выполняет определенную функцию. Обязательными являются два первых раздела – *Select* и *From*. Остальные указываются в случае необходимости. Рассмотрим подробнее назначение каждого раздела.

В предложении *Select* указывается список столбцов, разделенных запятой. Для каждого элемента из этого списка в таблице результатов запроса создается столбец. Столбцы в таблице результатов располагаются в том порядке, в каком были указаны в разделе *Select*. Возвращаемый столбец может представлять собой:

– имя столбца, идентифицирующее один из столбцов, содержащихся в таблицах, которые перечислены в предложении *From*. СУБД берет значение этого столбца для каждой из строк

исходной таблицы и помещает его в соответствующую строку таблицы результатов запроса;

- константу, показывающую, что в каждой строке результатов запроса должно содержаться одно и то же значение;
- выражение, показывающее, что СУБД должна вычислять значение, помещаемое в таблицу результатов запроса, по формуле, определенной в нем.

В предложении *From* указывается список таблиц, которые содержат поля, указанные в разделе *Select*. Запросы, возвращающие информацию из одной таблицы, описаны в этой лабораторной работе. Более сложные конструкции, формирующие выборку данных из двух или более таблиц, рассмотрены в следующих разделах.

Предложение *Where* показывает, что в результат запроса следует включать только те строки, которые удовлетворяют определенным условиям. Раздел *Where* реализует следующие виды условий отбора строк:

- сравнение. Значение одной части выражения сопоставляется со значением другой. В операции сравнения используются шесть операторов сравнения (=, <>, <, <=, >, >=). SQL Server позволяет задавать неравенство (<>) с помощью символов «!=».

- проверка на принадлежность диапазону. Для формирования условия отбора из диапазона используется двойное неравенство. Для задания границ диапазона можно применять оператор *Between... And*.

- проверка на членство в множестве. Выполняется при помощи оператора *In*, проверяющего совпадение значения с одним из значений заданного дискретного множества. С помощью проверки *Not In* можно убедиться в том, что элемент данных не принадлежит заданному множеству.

- проверка на соответствие шаблону, выполняемая оператором *Like*. В данном случае проверяется, соответствует ли значение, содержащееся в столбце, определенному шаблону.

Подстановочный знак «%» при поиске заменяет любую последовательность символов. Подстановочный знак «_» (подчеркивания) заменяет один любой символ.

С помощью формы *Not Like* можно находить строки, которые не соответствуют шаблону.

.....
 Предложение *Group by* позволяет создать итоговый запрос. Итоговый запрос, вначале группирует строки базы данных по определенному признаку, а затем включает в результаты запроса одну итоговую строку для каждой группы.

Предложение *Having* показывает, что в результаты запроса следует включать только некоторые из групп, созданных с помощью предложения *Group by*. В этом предложении, как и в предложении *Where*, для отбора включаемых групп используются условия отбора.

Предложение *Order by* сортирует результаты запроса на основании данных, содержащихся в одном или нескольких столбцах. Если это предложение не показано, результаты запроса не будут отсортированы. Сортировка может выполняться по нескольким столбцам по возрастанию (*Asc*) или убыванию (*Desc*).

Предложение *Into* и указанное в нем имя таблицы определяет объект для вставки отобранных данных.

Предложение *Union* позволяет объединить результаты выполнения двух SQL запросов.

Рассмотрим построение интерактивных запросов, выполненных средствами Transact-SQL. Для дальнейшей работы потребуется база данных *Trader*, восстановленная в последнем задании предыдущего раздела.

Практические задания

Задание 1. Вывести список всех офисов с адресами, телефонами и кодом офиса.

Выполнение.

Создайте новое подключение к компоненту *Database Engine*. Установите в качестве текущей базу данных *Trader* и запишите следующий текст запроса:

```
select id_office, gorod, ul, dom, tel1, tel2 from Office
```

Задание 2. Вывести список всех сотрудников, содержащий их коды, имена, фамилии, рабочие телефоны и дату приема на работу. Выполнить самостоятельно.

Задание 3. Вывести название всех товаров с указанием их единиц измерения. Выполнить самостоятельно.

Задание 4. Вывести список сделок, выполненных работником с кодом 4. Показать все поля таблицы.

```
select * from Sdelka where rid_Contr=4
```

Задание 5. Показать список сделок по товару с идентификатором $RID_{tov} = 1$. Вывести все поля таблицы *Sdelka*. Выполнить самостоятельно.

Задание 6. Показать сумму каждой сделки по продаже товара с идентификатором 5 (тип сделки $type='sal'$). Выводятся все поля таблицы *Sdelka* и вычисляемое поле.

```
select *, (cena*Kol_vo) as Sum_Sdelka from Sdelka where rid_tov=5 and type='sal'
```

Задание 7. Вывести список кодов сотрудников, продававших товар с идентификатором $rid_{tov}=2$.

```
select distinct rid_sotr from Sdelka where type='sal' and rid_tov=2
```

Попробуйте выполнить тот же запрос без служебного слова *Distinct*. Что изменилось? Какова функция *Distinct*?

.....

Задание 14. Отобрать все сделки, заключенные с контрагентами 5, 8, 9, 12 на сумму более 120 000. Отсортировать по номеру контрагента. Выполнить самостоятельно.

Задание 15. Вывести всю информацию о товарах Морковь, Бананы, Апельсины.

```
select * from tovar where nazv_tov in ('Морковь','Бананы', 'Апельсины')
```

Задание 16. Вывести список фирм (код, наименование, страна), которые расположены не на территории России, Беларуси и Египта. Выполнить самостоятельно.

.....

Задание 24. Найти информацию о сделках, где коды контрагентов равны 5, 7, 9 или даты сделок регистрировались с 10 по

14 марта 2016. Упорядочить значения сначала по полю *rid_contr*, а затем по полю *date_sdelka*.

Контрольные вопросы

1. Каково назначение предложения *Select*?
2. Каково назначение предложения *From*?
3. Каково назначение предложения *Where*?
4. Каково назначение предложения *Order by*?
5. Какая инструкция используется для сортировки полученных записей?
6. Как осуществить проверку на принадлежность (или не-принадлежность) в диапазоне значений?
7. Как задать поиск по шаблону? Какие служебные символы используются для формирования маски поиска?
8. Как формируются составные условия отбора записей?
9. Какова функция служебного слова *Distinct*?