

Базы данных

Лекция 00 – Административа

Преподаватель: Поденок Леонид Петрович, 505а-5

+375 17 293 8039 (505а-5)

+375 17 320 7402 (ОИПИ НАНБ)

prep@lsi.bas-net.by

ftp://student:2ok*uK2@Rwox@lsi.bas-net.by

Кафедра ЭВМ, 2023

Оглавление

План учебной дисциплины.....	3
Место учебной дисциплины.....	3
Содержание учебной дисциплины.....	8
Литература основная.....	10
Литература дополнительная.....	11
Важные замечания относительно литературы.....	12
Основная вычислительная среда.....	14
Приложения для работы с ER-моделями и прочим, что имеет отношение к БД.....	15
Лабораторные работы.....	16
Замечания по лабораторным работам.....	18
Состав документации, которую следует представить для сдачи лабораторной работы.....	18
Общие требования к оформлению ПЗ.....	22
Варианты тем лабораторных работ.....	24
Низкоуровневые операции логического уровня в таблицах.....	26

План учебной дисциплины

Лекций:	4+5+4+3 (32)
Лабораторных:	5+4+4+3 (32)
Самостоятельная работа:	44
Зачет	

Место учебной дисциплины.

Учебная дисциплина «Базы данных» предназначена для формирования расширенных знаний в области представления и обработки данных в рамках обучения студентов методикам и способам создания и проектирования программного обеспечения.

Цель преподавания учебной дисциплины: подготовка специалистов, владеющих теоретическими и практическими знаниями в области проектирования реляционных баз данных, использования их в решении различных производственных задач, а также их настройки и обслуживания.

Задачи изучения учебной дисциплины:

- формирование знаний о способах представления, хранения и обработки данных в системах баз данных;
- формирование знаний о методологиях проектирования баз данных и информационных систем, формирование практических навыков проектирования **реляционных баз данных**;
- формирование знаний и навыков практического использования языка **SQL** в создании, обслуживании и работе с базами данных;
- формирование знаний и навыков создания пользовательских приложений для работы с базами данных.

В результате изучения учебной дисциплины «Базы данных» формируются следующие компетенции:

академические:

- уметь применять базовые научно-теоретические знания для решения теоретических и практических задач;
- владеть системным и сравнительным анализом;
- уметь работать самостоятельно;
- владеть междисциплинарным подходом при решении проблем;
- иметь навыки, связанные с использованием технических устройств, управлением информацией и работой с компьютером;
- уметь учиться, повышать свою квалификацию в течение всей жизни;
- владеть основными методами, способами и средствами получения, хранения, переработки информации с использованием компьютерной техники;

социально-личностные:

- уметь работать в команде;

профессиональные:

- определять цели проектирования объектов профессиональной деятельности, критерии эффективности проектных решений, ограничений;
- реализовывать системный анализ объекта проектирования и предметной области, их взаимосвязей;
- разрабатывать требования и спецификации объектов профессиональной деятельности на основе анализа запросов пользователей, моделей предметной области и возможностей технических средств;
- выбирать средства вычислительной техники (ВТ), средства программирования с целью их применения для эффективной реализации аппаратно-программных комплексов;

- проектировать математическое, лингвистическое, информационное и программное обеспечение вычислительных систем (ВС) и автоматизированных систем на основе современных методов, средств и технологий проектирования, в том числе с использованием систем автоматизированного проектирования;

- работать с нормативно-технической документацией;

- систематизировать результаты и составлять отчеты по выполненной работе, обеспечивать контроль качества выполнения работ.

В результате изучения учебной дисциплины студент должен:

знать:

- термины и компоненты систем баз данных;

- модели представления данных;

- методологию проектирования баз данных;

- языки для работы с базами данных;

- программные интерфейсы для работы с базами данных.

уметь:

- применять модели данных для представления предметной области;

- создавать эффективные структуры баз данных для решения производственных задач;

- работать с базами данных на языке SQL;

- создавать прикладные программы, взаимодействующие с базами данных.

владеть:

- инструментальными средствами управления базами данных;

- инструментальными средствами разработки прикладного программного обеспечения нацеленного на работу с базами данных.

иметь представление:

- об организации и функционировании систем управления базами данных (СУБД);

- о тенденциях развития и перспективных направлениях использования баз данных.

Перечень учебных дисциплин, усвоение которых необходимо для изучения данной учебной дисциплины (программа).

Конструирование программ и языки программирования

Раздел 1. Язык программирования C++, в частности, темы:

- объекты и массивы объектов в C++ (**в БД нет ни объектов, ни методов**);
- потоки ввода-вывода в C++ (???)
- файловая система в C++ (???)
- дополнительные приемы конструирования программ.

На самом деле потребуется:

- уверенное владение клавиатурным вводом с терминала;
- Язык программирования C (или C++ в части императивного и структурного программирования);
- забыть ОО парадигму и вспомнить теорию множеств;
- прочитать внимательно документы:
 - ГОСТ 19.105 Общие требования к программным документам;
 - ГОСТ 19.106 Требования к программным документам, выполненным печатным способом;
 - ГОСТ 19.404 Пояснительная записка. Требования к содержанию и оформлению;
 - ГОСТ 19.701 Схемы алгоритмов, программ, данных и систем.

Содержание учебной дисциплины

№ тем	Наименование разделов, тем	Содержание
1	Введение в системы баз данных	Системы баз данных и информационные системы. Компоненты систем баз данных и их виды. Архитектура ANSI-SPARC. Функции и структура СУБД. Архитектура информационных систем.
2	Проектирование баз данных	Жизненный цикл информационных систем. Общая классификация моделей данных. Методология проектирования баз данных.
3	Концептуальные модели данных	Семантическое моделирование данных. Виды концептуальных моделей данных. ER-модель. ER-диаграмма.
4	Реляционная модель данных	Дореляционные логические модели данных. Структура реляционных данных. Целостность реляционных данных. Преобразование ER-диаграммы в реляционную схему данных. Реляционная алгебра и реляционное исчисление. Нормализация реляционных данных.
5	Языки баз данных	Язык SQL: идентификаторы, функции, типы данных, создание схемы данных, создание таблиц и индексов, выборка и модификация данных, представления, транзакции, управление доступом к данным. Язык QBE в сравнении с SQL.

№ тем	Наименование разделов, тем	Содержание
6	Физические модели данных	Файловые структуры данных и методы доступа: неупорядоченные, упорядоченные и хешированные файлы, индексы.
7	Разработка приложений	Виды приложений для работы с базами данных. Внедрение SQL-операторов в прикладные программы. Технологии и программные интерфейсы для доступа к базам данных.
8	Перспективы развития баз данных	Постреляционные и нереляционные СУБД. Хранилища данных. Интерактивная аналитическая обработка данных.
8	Локальные БД	Berkeley DB

Литература основная

1. Коннолли, Т. Базы данных: проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг, А. Страчан. – К.; М.; СПб.: ИД «Вильямс», 2003. – 1440 с.
2. **Дейт, К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL / К. Дж. Дейт. - СПб.: Символ-Плюс, 2010. – 480 с.**
3. **Дейт, К. Дж. Введение в системы баз данных / К. Дж. Дейт. – К.; М.; СПб.: ИД «Вильямс», 2006. – 1328 с.**
4. Грабер, М. SQL для простых смертных / М. Грабер. – М., «Лори», 2014. – 383 с.
5. Грабер, М. SQL / М. Грабер. – М., «Лори», 2003. – 664 с.
6. Гарсиа-Молина, Г. Системы баз данных. Полный курс / Г. Гарсиа-Молина, Д. Ульман, Д. Уидом. - К.; М.; СПб.: ИД «Вильямс», 2003. – 1088 с.
7. Yadava H. The Berkeley DB Book. Apress – 2007
8. Васильев А.Ю. Работа с PostgreSQL: настройка и масштабирование. – 2017.
9. Моргунов Е. П. PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с.: ил.
10. Лузанов П. и др. PostgreSQL для начинающих. – 2017.
11. Калабухов, Е. В. Базы данных, знаний и экспертные системы : лаб. практикум для студ. спец. I-40 02 01 «Вычислительные машины, системы и сети» всех форм обуч. / Е. В. Калабухов. – Минск : БГУИР, 2008. – 32 с. : ил.

Литература дополнительная

Агальцов, В.П. Базы данных. В 2-х т.Т. 1. Локальные базы данных: Учебник / В.П. Агальцов. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 352 с.

Агальцов, В.П. Базы данных. В 2-х т. Т. 2. Распределенные и удаленные базы данных: Учебник / В.П. Агальцов. – М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 272 с.

Голицына, О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - М.: Форум, 2012. - 400 с.

Кириллов, В.В. Введение в реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. - СПб.: БХВ-Петербург, 2012. - 464 с.

Карпова, И.П. Базы данных: Учебное пособие / И.П. Карпова. - СПб.: Питер, 2013. - 240 с.

Пирогов, В.Ю. Информационные системы и базы данных: организация и проектирование: Учебное пособие / В.Ю. Пирогов – СПб.: БХВ-Петербург, 2009. - 528 с.

Редько, В.Н. Базы данных и информационные системы / В.Н. Редько, И.А. Бассараб - М.: Знание, 2011. – 602 с.

Постолит, А. Visual Studio .NET: разработка приложений баз данных / А. Постолит. – СПб: БХВ, 2009. - 544 с.

Важные замечания относительно литературы

Представленный выше перечень взят из официальной программы, именно поэтому он здесь. Его наличие не означает, что все эти книги следует читать. Многие из них, что касается разработки реляционных баз данных, содержат информацию крайне сомнительного характера.

UML

Во многих из них можно встретить предложение использовать UML в качестве средства, якобы позволяющего эффективно проектировать объект автоматизации на всех уровнях.

Однако, следует заметить, что UML в своей основе является языком [графического описания] объектных моделей и он не в состоянии описывать системы хранения и манипулирования данными, в том числе и представленными в виде таблиц (реляционные системы).

Отличия в процессе проектирования появляются в процессе первичного рассмотрения задачи и серьезно углубляются на уровне модели «Сущность-связь».

ООП

Объектный подход и соответствующий язык можно использовать при проектировании систем, которые естественным образом могут быть представлены/описаны в рамках ОО модели, которая строится на принципах – класс, объект, абстракция данных, инкапсуляция, наследование, полиморфизм подтипов.

При проектировании реляционных (табличных) систем он не годится – здесь нет ни классов, ни объектов, ни методов, ни наследования, ни диаграмм использования, только теория множеств в чистом виде, сортировка и поиск.

Вера в то, что ООП – это серебряная пуля или «одна идеальная парадигма» – это ловушка.

Диаграммы использования

Особо следует отметить вред вариантов использования (use cases) – это метод, который навязывает частное решение, которое ограничивает схему данных и впоследствии это решение оказывается невозможно ни модифицировать, ни расширить.

При разработке баз данных следует всячески избегать подходов с вариантами использования на уровне пользователя – сегодня данные в организации принято использовать одним способом, завтра другим, а база данных должна соответствовать любому из вариантов, поскольку данные и связи между ними могут оставаться теми же.

Поэтому в первую очередь следует проектировать модель данных без принятия во внимание того, как пользователи используют эти данные сегодня, поскольку практика показывает, что после создания базы данных варианты использования данных меняются кардинально – старые методы перестают быть актуальными и появляются новые.

СУБД всегда определяет полное множество вариантов использования данных (операций) в рамках конкретной схемы данных – пользователю остается лишь **выбрать и реализовать** те, которые ему подходят.

Как произносится правильно:

Термин	Транскрипция 1	Транскрипция 1
SQL	[ˈɛsˈkjuˈɛl]	[ˈsiːkwəl];
PostgreSQL	[ˈpoʊstˈɡrɛsˈsiːkwəl]	[ˈpoʊstˈɡrɛsˈkjuˈɛl]

Основная вычислительная среда

ОС: Linux, MS Windows

СУБД: PostgreSQL, Berkeley DB

PostgreSQL – продвинутая система управления объектно-реляционными базами данных

Пакет postgresql-server содержит программы, необходимые для создания и запуска сервера PostgreSQL, который позволяет создавать и поддерживать базы данных PostgreSQL.

Базовый пакет postgresql содержит клиентские программы, необходимые для доступа к серверу СУБД PostgreSQL, а также документацию в формате HTML для всей системы. Эти клиентские программы могут располагаться на той же машине, что и сервер PostgreSQL, или на удаленной машине, который обращается к серверу PostgreSQL через сетевое соединение.

Berkeley DB – The Berkeley DB database library for C

База данных Berkeley (Berkeley DB) — это программный инструментарий, обеспечивающий **встроенную поддержку** баз данных как для традиционных, так и для клиент-серверных приложений. Berkeley DB включает в себя B+tree, расширенное линейное хеширование, методы доступа к записям фиксированной и переменной длины, транзакции, блокировку, ведение журнала, кэширование в разделяемой памяти и восстановление базы данных. Berkeley DB поддерживает API C, C++ и Perl. Он используется многими приложениями, включая Python и Perl.

Приложения для работы с ER-моделями и прочим, что имеет отношение к БД

В отличие от ООП, где есть UML и куча софта для рисования диаграмм, программных средств для разработки модели «Сущность-связь», годной для преобразования в реляционную, за исключением входящих в состав проприетарных СУБД, практически нет.

Тем не менее, программные средства, предназначенные для разработки реляционной модели, присутствуют почти в каждой из СУБД. Поэтому проектирование баз данных с самого начала чаще всего ведется именно на этом уровне – реляционном.

Однако, программа курса требует и включает в качестве лабораторной работы разработку модели «сущность-связь». Поэтому нужен инструмент для вычерчивания такого рода диаграмм. В качестве такого инструмента можно использовать любые схеморисовалки, позволяющие оформить схему в соответствии с требованиями ГОСТ 19 и практиками полиграфии русского языка в отношении надписей на блоках. Например, visio в MS Office или draw в Libre/Openoffice.

Есть также и другие схемочертилки.

Что касается реляционных диаграмм, то несмотря на большое количество рисовалок, сопровождающих различные СУБД, подходящих для выполнения лабораторных работ практически нет.

Дело в том, что отрисовка схем ими выполняется после того, как будут созданы сами таблицы и ограничения, в основном для фиксации существующих схем данных, что в корне противоречит самой сути использования реляционных диаграмм, особенно в рамках лабораторных работ.

Лабораторные работы

Студент выбирает тему и последовательно ее развивает в рамках лабораторных работ. Все лабораторные должны быть правильно оформлены.

№ 1 «Создание ER-диаграммы»

Выполняется концептуальное проектирование БД с использованием ER-модели представления данных (модели «сущность-связь») с учетом семантических ограничений заданной предметной области и представить модель в виде ER-диаграммы.

№ 2 «Создание реляционной схемы данных»

Выполняется логическое проектирование БД. Спроектированная в ЛР № 1 ER-диаграмма преобразуется в реляционную схему данных и оформляется в графическом виде.

№ 3 «Реализация базы данных»

Установка и настройка PostgreSQL. Изучение консольного клиента `psql`.

Реализация схемы базы данных по ранее построенной реляционной схеме данных (см. лабораторную работу № 2). Требуется сформировать SQL-запросы для создания таблиц базы данных и выполнить их в СУБД, а также заполнить таблицы данными с помощью оператора `INSERT`.

№ 4 «Реализация SQL-запросов на простую выборку данных»

В лабораторной работе выполняется создание простых запросов на выборку данных на языке SQL с использованием предложений `SELECT`, `FROM` (`JOINS`), `WHERE` и `ORDER BY` оператора `SELECT`. В работе также требуется рассмотреть использование скалярных функций.

№ 5 Выборка данных с использованием подзапросов, агрегатных функций, группировки и операций над множествами

Используются предложение `GROUP BY` оператора `SELECT`, операции над множествами (`UNION`, `INTERSECT`, `MINUS`).

Лабораторная работа № 6 (на автомат)

Освоение низкоуровневых механизмов взаимодействия с базами данных.

Часть 1. Создание локальной базы данных класса ACID с помощью библиотеки Berkeley DB.

В лабораторной работе выполняется создание совместно используемой несколькими клиентами базы данных с помощью библиотеки Berkeley DB.

БД содержит несколько таблиц, связанных отношениями 1:M, N:M, N:1.

Часть 2. Создание интерактивного приложения для работы с локальной БД, поддерживающего основные операции – вставка/удаление записей, просмотр таблиц, получение/изменение записей, агрегатные операции.

Замечания по лабораторным работам

1) Лабораторные выполняются индивидуально.

2) Студент выбирает одну из тем, представленных ниже, и все пять лабораторных работ выполняет в ее рамках.

3) Структура лабораторной работы должна быть максимально близка к реальности.

В подавляющем случае никакому хозяину/руководителю предприятия, которое вы выбрали для автоматизации в рамках лабораторных, никакая база данных не нужна – ему удобнее вести вообще все делопроизводство шариковой ручкой в журналах.

Причем в двух – один для себя, другой для проверяющих. Поэтому каждый из вас должен стоять на позициях работников из проверяющих органов, например, следователя или налогового инспектора, если ваша модель имеет дело с деньгами. Это значит, например, что если у вас в столовой есть сотрудники и должности, то следователь, расследующий отравление гречневой кашей, должен иметь возможность получить из вашей БД информацию о всех, кто работал на должности повара в период с и по, а также что он там такого мог сварить. Ну и далее в этом ключе.

Состав документации, которую следует представить для сдачи лабораторной работы

1) Пояснительная записка в формате PDF для каждой из лабораторных работ. На титульном листе обязательно должно присутствовать название выбранной темы и название лабораторной работы. Номер темы необязателен. Требования к оформлению текстовой части см. ниже.

2) SQL-скрипты для лабораторных №№ 3, 4, 5. Состав скриптов см. ниже.

Пояснительная записка и скрипты должны располагаться в каталоге labN, где N – номер лабораторной. Каталог архивируется архиватором tar и *может быть* сжат компрессором, который tar умеет использовать.

lab1

Структура описания ПЗ следующая:

1. Описание ER-модели «Название темы»

1.1. Сущности

Описание всех сущностей и их атрибутов. Все сущности и их атрибуты должны быть поименованы на русском языке. Для сущностей приводятся их абстрактные типы без какой либо привязки к типам языков программирования («деньги», «дата», ...)

Количество сущностей — не менее семи.

1.2. Связи

Здесь описание всех связей между сущностями.

Количество связей многие ко многим — не менее трех.

2. Схема ER-модели.

lab2

Структура описания ПЗ следующая:

1. Описание реляционной модели «Название темы»

1.1. Перечень всех таблиц и каким сущностям они соответствуют. Перечень всех полей и каким атрибутам они соответствуют. Типы не указываем. Перечень ограничений. Перечень атрибутов, требующих сортировки при визуализации.

1.2. Связи

Здесь описание всех связей между сущностями.

2. Схема ER-модели из лабораторной №1.

3. Схема реляционной модели.

lab3

Структура описания ПЗ следующая:

1. Схема реляционной модели из лабораторной №2.
2. Описывается скрипт создания, описываются ограничения. Обосновываются выбранные типы данных для полей.
3. Описывается скрипт заполнения.

В составе архива с отчетом по лабораторной два скрипта — скрипт создания БД (`create.sql`) и скрипт ее заполнения (`insert.sql`). Скрипты создания и заполнения должны иметь комментарии для получения представления, что они делают.

Сами скрипты копировать в ПЗ не нужно.

lab4

Структура описания ПЗ следующая:

1. Схема реляционной модели из лабораторной №2.
2. Краткое описание каждого из скриптов получения данных из базы (будет использоваться в качестве комментария в скрипте с запросами).

В этой лабораторной три скрипта и файл с результатами:

`create.sql` из lab3;

`insert.sql` из lab3;

`select.sql` с запросами.

`results.txt`

lab5

Структура описания ПЗ следующая:

1. Схема реляционной модели из лабораторной №2.

2. Коротко описывается каждый скрипт получения данных из базы (будет использоваться в качестве комментария в скрипте с запросами).

В этой лабораторной четыре скрипта и файл с результатами:

create.sql из lab3;

insert.sql из lab3;

select.sql с запросами из lab4;

results.txt

select5.sql с запросами;

results5.txt.

Схемы выполняются на отдельной странице в ориентации «Ландшафт» с центрированной подрисуночной подписью в формате ГОСТ 7-32 (Рисунок 2 – Схема ER-модели «Название темы»).

Выход запросов не должен быть пустым, поэтому БД должна быть пополнена данными.

В ряде случаев может быть изменена и ее структура. В таком случае все изменения добавления в схему БД и в скрипты подробно описываются в ПЗ соответствующей лабораторной работы.

Лабораторные 3, 4 и 5 выполняются в консольном клиенте psql. Устанавливается режим протоколирования с выводом в файл. Этот файл редактируется, в частности, размечается по отдельным запросам комментариями, и из него формируется файл с результатами.

Все должно быть читабельно и аккуратно.

Скрипты следует форматировать отступами, чтобы была видна их структура.

Общие требования к оформлению ПЗ

Формат страницы текстовой части А4. Все листы кроме первого нумеруются внизу по центру.

Поля:

левое — 3 см;

правое — 1 см;

верхнее — 2 см;

нижнее — 2 см.

Абзац начинается с абзацного отступа размером в 5 букв 'н' (как учили в школе).

Межстрочный интервал одинарный.

Шрифт абзаца пропорциональный типа serif (Times New Roman, PT Astra Serif, Liberation Serif, ...), кегль 12-14 pt.

Шрифт имен таблиц, полей и прочих идентификаторов по тексту – моноширинный с отличием единицы от буквы 'l' и с плотностью, равной плотности основного шрифта абзаца (Hack, Fira Code, gost, Intel One Mono).

Переносы в тексте включены, в заголовках отключены.

Перечисления

Ненумерованные перечисления (перечни, списки) ничем не обозначаются, либо используется дефис, после которого идет неразрывный пробел.

Именованные перечисления оформляются, как ненумерованные без дефиса, имя при этом выделяется гарнитурой или полужирным.

Нумерованные перечисления нумеруются цифрой со скобкой или малой буквой русского алфавита, после которой идет неразрывный пробел.

Оформление схем

Толщина линий блоков и линий – 0,25..0,3 мм.

Надписи в блоках шрифт типа sans (Arial, PT Sans, PT Astra Sans, Liberation Sans, ...) тем же размером, что и текст абзаца.

Концы линий связи помечаются цифрами или буквами.

Варианты тем лабораторных работ

- 1) Автозаправка.
- 2) Автосалон.
- 3) Автостоянка.
- 4) Автошкола.
- 5) Аэропорт.
- 6) Банк.
- 7) Больница.
- 8) Военкомат.
- 9) Гостиница.
- 10) Грузоперевозки.
- 11) Детский сад.
- 12) Железнодорожный вокзал (автовокзал, такси и т.п.).
- 13) Завод.
- 14) Зоопарк.
- 15) Кафе (бар).
- 16) Кинотеатр.
- 17) Локальная компьютерная сеть (кабельное телевидение).
- 18) Магазин (музыкальный, продовольственный и т.п.).
- 19) Налоговая инспекция.
- 20) Общежитие.
- 21) Оператор связи.
- 22) Организация концертов.
- 23) Поликлиника.
- 24) Прокат видеодисков.

- 25) Ресторан.
- 26) СТО.
- 27) Столовая.
- 28) Студия звукозаписи (киностудия).
- 29) Туристическое агентство.
- 30) Футбольный клуб.
- 31) Школа.
- 32) Бассейн.

Низкоуровневые операции логического уровня в таблицах

TABLE LIST:LST (ID, NAME, DATE, ...)

TABLE ITEM:ITM (ID, LIST_ID, PART_NO, NAME, PROVIDER, QUANT, ...)

PRIMKEY LST:BY_ID (ID) # LST:ID

PRIMKEY ITM:BY_ID (ID) # ITM:ID

INDEX LST:BY_NAME (NAME)

INDEX LST:BY_DATE (DATE)

INDEX ITM:BY_NAME (NAME)

INDEX ITM:BY_NAME_INLIST (LIST_ID, NAME)

INDEX ITM:BY_PNO_INLIST (LIST_ID, PART_NO) # кандидат в PrimKey (?поставщик?)

Простейшие сериальные операции

SET(TABLE) – установка курсора в «сыром» порядке записей

NEXT(TABLE) – просмотр таблицы в направлении с начала к концу

PREV(TABLE) – просмотр таблицы в направлении с конца к началу

INSERT(RECORD, TABLE) – вставка в таблицу без обновления индексов

Прямой доступ по первичному ключу

GET(RECORD, TABLE, KEY) – получить строку (запись) из таблицы по значению перв. ключа

PUT(RECORD, TABLE, KEY) – обновить строку (запись) в таблице по значению перв. ключа

Добавление/удаление

ADD(RECORD, TABLE) – добавить строку (запись) в таблицу и обновить все индексы

DEL(RECORD, TABLE, KEY) – удалить строку (запись) из таблицы и обновить все индексы

Прямой доступ по ключу

GET(RECORD, TABLE, KEY, KEY_VALUE) – получить строку по значению ключа

PUT(RECORD, TABLE, KEY, KEY_VALUE) – обновить строку по значению ключа

ADD(RECORD, TABLE) – добавить строку (запись) в таблицу и обновить все индексы

DEL(RECORD, TABLE, KEY) – удалить строку (запись) из таблицы и обновить все индексы

Сериальные операции в порядке индексов

SET(TABLE, KEY, KEY_VALUE) – установить курсор доступа в порядке индекса

NEXT(TABLE, KEY) – просмотр таблицы в направлении с начала к концу

PREV(TABLE, KEY) – просмотр таблицы в направлении с конца к началу