**Лабораторная работа № 5**

«Создание даталогической модели базы данных»

Перед выполнением работы рекомендуется повторить теоретический материал, т.е. ещё раз пересмотреть данные видео («[Цели моделирования на даталогическом уровне](https://lms2.bsuir.by/mod/lti/view.php?id=75269)», «[Техники и инструменты моделирования на даталогическом уровне](https://lms2.bsuir.by/mod/lti/view.php?id=75270)», «[Пример моделирования на даталогическом уровне](https://lms2.bsuir.by/mod/lti/view.php?id=75271)») и перечитать раздел 4.2 «Проектирование на даталогическом уровне» [книги](https://svyatoslav.biz/relational_databases_book/).

Продолжите проектирование базы данных:

1. Опираясь на созданную ранее в лабораторной работе № 4 концептуальную модель, создайте логическую (даталогическую) модель с помощью Sparx Enterprise Architect (вы можете использовать пробную версию).
2. Проверьте полученный результат, ответив на ряд вопросов:

* Все ли объекты в базе данных соответствуют единым соглашениям об именовании объектов? Исправьте те, которые нарушают такие соглашения.
* Все ли атрибуты в базе данных имеют оптимальные типы данных? Если «нет», оптимизируйте те из них, которые требуют оптимизации.
* Все ли отношения в базе данных имеют оптимальные ключи? Если нет, оптимизируйте те из них, которые требуют оптимизации.
* Есть ли в базе данных избыточные или отсутствующие связи? Внесите необходимые исправления.
* Существует ли возможность аномалий операций вставки, обновления, удаления данных? Составьте список таких случаев и внесите исправления, чтобы избежать таких аномалий.
* Нарушает ли схема какие-либо «требования нормализации»? Если «да», доработайте схему, чтобы избежать таких нарушений.
* Добавьте в базу данных все необходимые индексы.

1. Вставьте в отчёт скриншот созданной модели.
2. Вставьте в отчёт описание созданной модели в виде таблицы, приведённой ниже.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Отношение БД | Атрибут отношения БД | Назначение/подробное описание атрибута | Тип данных | Комментарии |
| CPU | id | Идентификатор — первичный ключ, однозначно иденцифицирующий все атрибуты отношение | UNSIGNED INT |  |
| name | Название процессора | CHAR(50) |  |
| series | Линейка процессора | CHAR(50) |  |
| cores\_number | Количество ядер | UNSIGNED TINYINT |  |
| frequency | Частота | UNSIGNED INT |  |
| socket | Вид сокета | VARCHAR(45) |  |
| cache | Кэш(L3) | VARCHAR(45) |  |
| RAM | id | Идентификатор — первичный ключ, однозначно иденцифицирующий все атрибуты отношение | UNSIGNED INT |  |
| name | Название оперативной памяти | CHAR(50) |  |
| volume | Объём оперативной памяти в гигабайтах | UNIGNED TINYINT |  |
| pc-index | Частота оперативной памяти умноженная на 8(зачастую используется для более удобной маркировки и поиска именно в таком формате) | UNSIGNED INT |  |
| type | Поколение оперативной памяти(DDR1, DDR2 ...) | ENUM() |  |
| frequency | Частота оперативной памяти | UNSIGNED INT |  |
| HDD | id | Идентификатор — первичный ключ, однозначно иденцифицирующий все атрибуты отношение | UNSIGNED INT |  |
| name | Название жёсткого диска | CHAR(50) |  |
| volume | Объём в гигабайтах | UNSIGNED INT |  |
| buffer | Объём буфера в мегабайтах | UNSIGNED INT |  |
| energy\_consumption | Эергопотребление в Ваттах | UNSIGNED FLOAT |  |
| Spares | id | Идентификатор — первичный ключ, однозначно иденцифицирующий все атрибуты отношение | UNSIGNED INT |  |
| type | Тип запчасти(процессор, метизы, читящая жидкость и т.д.) | ENUM() |  |
| amount | Кол-во запчастей(нужно если это например 10000 маленьких болтов) | UNSIGNED INT |  |
| price | Цена жапчасти(за одну штуку) | UNSIGNED FLOAT |  |
| Personal computer | id | Идентификатор — первичный ключ, однозначно иденцифицирующий все атрибуты отношение | UNSIGNED INT |  |
| name | Название персонального компьютера | CHAR(50) |  |
| Customer | id | Идентификатор — первичный ключ, однозначно иденцифицирующий все атрибуты отношение | UNSIGNED INT |  |
| name | Имя клиента | VARCHAR(45) |  |
| phone\_number | Мобильный телефон клиента | VARCHAR(20) |  |
| email | Адрес электронной почты | VARCHAR(150) |  |
| address | Адрес прописки | VARCHAR(150) |  |
| Worker | id | Идентификатор — первичный ключ, однозначно иденцифицирующий все атрибуты отношение | UNSIGNED INT |  |
| name | Имя | CHAR(50) |  |
| specialization | Специализация | ENUM() |  |
| experience | Количество лет работы по специализации | UNSIGNED TINYINT |  |
| salary | ЗП | UNSIGNED FLOAT |  |
| Contract | id | Идентификатор — первичный ключ, однозначно иденцифицирующий все атрибуты отношение | UNSIGNED INT | Для подписания контракта требуется знать  директора, работника,  клиента  и его компьютер |
| price | Цена | UNSIGNED FLOAT |  |
| signing\_date | Дата заключения | DATE |  |
| completion\_date | Дата завершения(предположительная) | DATE |  |
| description | Описание условий и договорённостей, какой-либо доп. информации | VARCHAR(2000) |  |
| status | Состояние(в работе, завершён, отклонён, продлён) | ENUM() |  |
| decline\_reason | Причина расторжения контракта | VARCHAR(200) |  |
| Diagnositc | id | Идентификатор — первичный ключ, однозначно иденцифицирующий все атрибуты отношение | UNSIGNED INT |  |
| description | Подробное описание проводящихся работ | VARCHAR(1000) |  |
| price | Цена | UNSIGNED FLOAT |  |
| Cleaning | id | Идентификатор — первичный ключ, однозначно иденцифицирующий все атрибуты отношение | UNSIGNED INT |  |
| description | Подробное описание проводящихся работ | VARCHAR(1000) |  |
| price | Цена | UNSIGNED FLOAT |  |
| Repairing | id | Идентификатор — первичный ключ, однозначно иденцифицирующий все атрибуты отношение | UNSIGNED INT |  |
| description | Подробное описание проводящихся работ | VARCHAR(1000) |  |
| price | Цена | UNSIGNED FLOAT |  |
|  |  |  |  |  |
|  |  |  |  |  |

**ИНДЕКСЫ**

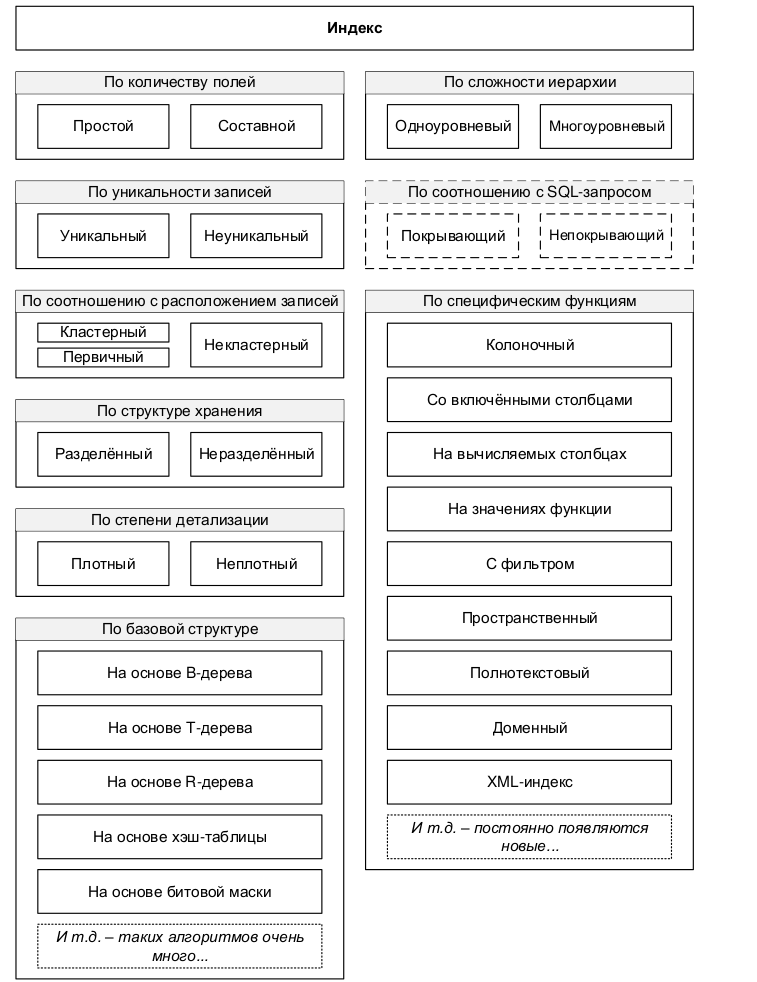
Индекс (index66) — специальная структура базы данных, используемая

для ускорения поиска записей и физического доступа к записям.

Упрощённо: механизм, значительно ускоряющий поиск необходимой ин-

формации в базе данных (по аналогии: для человека карта города явля-

ется таким «индексом», позволяющим быстро найти нужное здание)

 .

**По количеству полей**

Простой индекс (simple index, single-column index67) — индекс, построен-

ный на одном поле таблицы.

Упрощённо: индекс, включающий информацию о содержимом только од-

ного поля таблицы.

Составной индекс (composite index68) — индекс, построенный на двух и

более полях таблицы.

Упрощённо: индекс, включающий информацию о содержимом двух и бо-

лее полей таблицы.

**По уникальности записей**

Уникальный индекс (unique index70) — индекс, построенный на содержа-

щем уникальные значения поле (полях) таблицы.

Упрощённо: индекс, построенный на поле, являющемся первичным или

альтернативным ключом таблицы.

Неуникальный индекс (non-unique index, index71) — индекс, построенный

на не содержащем уникальные значения поле (полях) таблицы.

Упрощённо: «просто индекс» (для неуникального индекса в синтаксисе

SQL даже нет отдельного ключевого слова, просто пишется INDEX).

**По соотношению с расположением записей**

Кластерный индекс (clustered index72) — индекс, построенный на поле

(возможно, с неуникальными значениями), по которому произведено фи-

зическое упорядочивание данных в файле.

Упрощённо: по значению индексированного поля данные таблицы физи-

чески отсортированы на диске; значения поля могут повторяться.

Первичный индекс (primary index73) — индекс, построенный на поле с

уникальными значениями, по которому произведено физическое упорядо-

чивание данных в файле.

Упрощённо: по значению индексированного поля данные таблицы физи-

чески отсортированы на диске; значения поля не могут повторяться.

Некластерный индекс (non-clustered index74) — индекс, построенный на

поле, по которому не произведено физическое упорядочивание данных в

файле.

Упрощённо: «просто индекс», принципы упорядочивания в котором ни-

как не связаны с физическим расположением данных на диске.

**По структуре хранения**

Разделённый индекс (partitioned index76) — индекс, хранимый и обраба-

тываемый в виде отдельных частей (разделов, фрагментов) с целью по-

вышения производительности.

Упрощённо: индекс, разделённый на несколько фрагментов.

Неразделённый индекс (non-partitioned index, index77) — индекс, храни-

мый и обрабатываемый как единая структура данных.

Упрощённо: «просто индекс» (без применения специальных команд по

умолчанию все индексы создаются как неразделённые).

**По степени детализации**

Плотный индекс (dense index81) — индекс, содержащий указатель на рас-

положение записи для каждого значения индексируемого поля.

Упрощённо: в индексе хранятся адреса каждой записи таблицы.

Неплотный индекс (sparse index82) — индекс, содержащий указатель на

расположение блока записей для каждого значения (в случае их неуни-

кальности) или группы значений (в случае их уникальности) индексируе-

мого поля.

Упрощённо: в индексе хранятся адреса блоков (групп) записей.

**По базовой структуре**

Индекс на основе B-дерева (сбалансированного дерева) (B-tree83 in-

dex) — индекс, структурно организованный с использованием B-дерева

(сбалансированного дерева), оптимизированный для выполнения поиска

на основе диапазонов и для операций с большими блоками данных. До-

пускает хранение части индекса во внешней памяти.

Упрощённо: одна из основных форм организации индексов в большин-

стве СУБД и методов доступа. (Буква «B» в названии индекса идёт от

слова «balanced».)

Индекс на основе T-дерева (T-tree84 index) — индекс, структурно органи-

зованный с использованием T-дерева (разновидности сбалансированного

дерева, в котором вместо самих данных хранятся указатели на адреса

данных в памяти), оптимизированный для выполнения операций в случае,

когда и индекс и данные целиком находятся в оперативной памяти.

Упрощённо: индекс для СУБД и методов доступа, предполагающих хра-

нение всего объёма обрабатываемых данных в оперативной памяти.

(Буква «T» в названии индекса идёт от графической формы представ-

ления вершин T-дерева в статье, в которой оно было впервые пред-

ставлено.)

Индекс на основе R-дерева (R-tree85 index) — индекс, структурно органи-

зованный с использованием R-дерева (специальной формы представле-

ния географических и геометрических данных), оптимизированный для

выполнения операций со специфическими типами данных, хранящих гео-

графические координаты или информацию о геометрических фигурах.

Упрощённо: индекс для ускорения обработки географических и геомет-

рических данных (Буква «R» в названии индекса идёт от слова

«rectangle».).

Индекс на основе хэш-таблицы (hash-table86 index) — индекс, структурно

организованный с использованием хэш-таблицы (специальной структуры

для хранения пар ключ-значение), оптимизированный для выполнения по-

иска на основе строгого сравнения и обработки относительно редко изме-

няемых данных.

Упрощённо: одна из основных форм организации индексов в большин-

стве СУБД и методов доступа (наряду с индексами на основе B-

деревьев).

Индекс на основе битовой маски (bitmap87 index) — индекс, структурно

организованный с использованием битовой маски (специальной струк-

туры для хранения информации о наличии в поле того или иного значе-

ния), оптимизированный для работы со столбцами, количество различных

значений в которых относительно невелико.

Упрощённо: индекс хранит в очень компактной форме признак наличия

в некоторой ячейке таблицы одного из значений, присутствующих в со-

ответствующем столбце.

**По сложности иерархии**

Одноуровневый индекс (single-level index88) — индекс, структура кото-

рого является плоской, т.е. содержит ровно один уровень.

Упрощённо: индекс без иерархии.

Многоуровневый индекс (multi-level index89) — индекс, структура кото-

рого является иерархической, т.е. содержит два и более уровня. Такие

уровни могут содержать однотипную информацию или различаться по

своему назначению (как правило, листовые узлы будут отличаться от не-

листовых, в то время как все нелистовые вне зависимости от их уровня

выполняют одинаковую роль).

Упрощённо: индекс с иерархией (как правило, на основе дерева).

**По соотношению с SQL-запросом**

Покрывающий индекс (covering index90) — индекс, содержащий в явном

виде внутри себя информацию, достаточную для выполнения SQL-

запроса без обращения к данным, хранящимся вне этого индекса (в самой

таблице).

Упрощённо: индекс, информации в котором достаточно для выполнения

SQL-запроса без обращения к данным в таблице.

Непокрывающий индекс (non-covering index91) — индекс, позволяющий

лишь ускорить нахождение нужной информации, в то время как сама ис-

комая информация внутри индекса не содержится или содержится не пол-

ностью.

Упрощённо: «просто индекс», информации в котором недостаточно

для выполнения SQL-запроса без обращения к данным в таблице.

**По специфическим функциям**

Колоночный индекс (columnstore index92) — набор решений для хранения

и обработки данных таким образом, что базовой единицей является не

ряд (как в классическом случае), а столбец.

Упрощённо: индекс для быстрой обработки отдельных столбцов.

Индекс со включёнными столбцами (index with included columns93) — не-

кластерный{111} индекс, содержащий в своих листовых узлах информацию

из дополнительного поля, которое не используется при построении самого

индекса.

Упрощённо: индекс, позволяющий покрывать{125} больше запросов за

счёт содержания в себе дополнительных данных.

Индекс на вычисляемых столбцах (index on computed columns94) — ин-

декс, построенный на значениях виртуальных столбцов, данные которых

могут не храниться физически в базе данных.

Упрощённо: индекс на столбце, значение которого вычисляет сама

СУБД.

Индекс на значениях функций (function-based index95) — индекс, постро-

енный на результатах применения к хранящимся данным функции (встро-

енной или пользовательской).

Упрощённо: индекс на результатах применения к данным некоторой

функции (значение функции вычисляет сама СУБД).

Индекс с фильтром (filtered index96) — индекс, учитывающий из всего

множества значений индексируемого столбца лишь (небольшую) часть,

удовлетворяющую указанному при создании индекса условию.

Упрощённо: индекс на части значений столбца.

Пространственный индекс (spatial index98) — индекс, оптимизированный

для ускорения операций поиска значений в столбцах, хранящих данные

пространственного или геометрического типа.

Упрощённо: индекс на столбцах с пространственным или геометриче-

ским типом данных.

Полнотекстовый индекс (full text index99) — индекс, оптимизированный

для ускорения операций поиска вхождений подстрок в значения текстовых

полей.

Упрощённо: индекс для поиска текста в тексте.

Доменный индекс (domain index100) — индекс, используемый для работы

со специфическими данными в конкретной предметной области; взаимо-

действие с таким индексом реализуется через пользовательские подпро-

граммы (в отличие от «обычных» индексов, управление которыми уже ре-

ализовано в самой СУБД).

Упрощённо: индекс, логикой работы которого вы управляете сами.

XML-индекс (XML index103) — индекс, оптимизированный для обработки

XML-данных и ускорения поиска по путям и значениям внутри XML-

документов.

Упрощённо: индекс для ускорения работы с XML.