Основы проектирования баз данных

Лекция 6

База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе.

База данных обычно управляется ***системой управления базами данных (СУБД)***.

Данные вместе с СУБД, а также приложения, которые с ними связаны, называются ***системой баз данных***, или, для краткости, просто базой данных.

Типы баз данных:

- Реляционные базы данных. Данные в реляционной базе организованы в виде таблиц, состоящих из столбцов и строк.
- Распределенные базы данных. Состоят из двух или более частей, расположенных на разных серверах.
- Базы данных NoSQL. Это нереляционная база данных, дает возможность хранить и обрабатывать неструктурированные или слабоструктурированные данные.
- Базы данных OLTP. Предназначены для выполнения бизнес-транзакций, выполняемых множеством пользователей.

Создание базы данных и таблиц

1. Выделение сущностей и их атрибутов, которые будут храниться в базе данных, и формирование по ним таблиц. Атомизация сложных атрибутов на более простые.
2. Определение уникальных идентификаторов (первичных ключей) объектов, которые хранятся в строках таблицы
3. Определение отношений между таблицами с помощью внешних ключей
4. Нормализация базы данных

На первом этапе происходит выделение сущностей. ***Сущность*** (entity) представляет тип объектов, которые должны храниться в базе данных. Каждая таблица в базе данных должна представлять одну сущность. Как правило, сущности соответствуют объектам из реального мира.

У каждой сущности определяют набор атрибутов. ***Атрибут*** представляет свойство, которое описывает некоторую характеристику объекта. Каждый столбец должен хранить один атрибут сущности.

А каждая строка представляет отдельный объект или экземпляр сущности.

Восходящий и нисходящий подходы к выделению сущностей и атрибутов

Том посещает курс по математике, который преподает профессор Смит.

Сэм посещает курс по математике, которые преподает профессор Смит.

Том посещает курс по языку JavaScript, который преподает ассистент Адамс.

Боб посещает курс по алгоритмам, который преподает ассистент Адамс.

Сэм имеет следующие электронный адрес sam@gmail.com и телефон +1235768789.

Студент	Преподаватель	Курс
Имя студента	Имя преподавателя	Имя студента
Название курса	Должность преподавателя	Имя преподавателя
Дата рождения студента	Название курса	Название курса
Электронный адрес студента		
Телефон студента		

Атомизация атрибутов

При определении атрибутов происходит разделение сложных комплексных элементов на более простые. Так, в случае с именем студента его можно разбить на имя, фамилию и отчество. Это позволит впоследствии выполнять операции с этими подэлементами отдельно, например, сортировать студентов только по фамилии.

В то же время возможность разделения одного элемента на подэлементы не всегда может быть востребованной. В ряде задач это может быть просто не нужно. Выделять необходимо только те элементы, которые действительно нужны.

Домен

Каждый атрибут имеет домен (domain). Домен представляет набор допустимых значений для одного или нескольких атрибутов. По сути домен определяет смысл и источник значений, которые могут иметь атрибуты.

Домены могут отличаться для разных атрибутов, но также несколько атрибутов могут иметь один домен.

Определяя домен, мы сразу видим, какие данные и каких типов будут хранить атрибуты. Какое-то другое значение, которое не соответствует домену, атрибут иметь не может.

- Имя. Домен представляет все возможные имена, которые могут использоваться. Каждое имя представляет строку длиной максимум 20 символов (маловероятно, что нам могут встретиться имена свыше 20 символов).
- Фамилия. Домен представляет все возможные фамилии, которые могут использоваться. Каждая фамилия представляет строку длиной максимум 20 символов.
- Год рождения. Домен представляет все года рождения. Каждый год является числовым значением от 1950 до 2017.

Определитель NULL

При определении атрибутов и их домена необходимо проанализировать, а может ли у атрибута отсутствовать значение. Определитель NULL позволяет задать отсутствие значения. Например, в примере выше у студента обязательно должно быть какое-либо имя, поэтому недопустима ситуация, когда у атрибута, который представляет имя, отсутствует значение.

Как правило, большинство современных реляционных СУБД поддерживают определитель NULL и позволяют задать его допустимость для столбца таблицы.

Ключи

Ключи представляют способ идентификации строк в таблице. С помощью ключей можно связывать строки между различными таблицами в отношения.

Первичный ключ (primary key) непосредственно применяется для идентификации строк в таблице. Он должен соответствовать следующим ограничениям:

- должен быть уникальным все время
- должен постоянно присутствовать в таблице и иметь значение
- не должен часто менять свое значение. В идеале вообще не должен изменять значение.
- как правило, первичный ключ представляет один столбец таблицы, но также может быть составным и состоять из нескольких столбцов.

Базы данных могут содержать таблицы, которые связаны между собой различными связями. ***Связь*** (relationship) представляет ассоциацию между сущностями разных типов.

Для организации связи используются внешние ключи. ***Внешний ключ (foreign key)*** представляет один или несколько столбцов из одной таблицы, который одновременно является ключом из другой таблицы. Внешний ключ необязательно должен соответствовать первичному ключу из главной таблицы.

Связи между таблицами бывают следующих типов:

- Один к одному (объекту одной сущности можно сопоставить только один объект другой сущности)
- Один к многим (несколько строк из дочерний таблицы зависят от одной строки в родительской таблице)
- Многие ко многим (одна строка из таблицы А может быть связана с множеством строк из таблицы В. В свою очередь одна строка из таблицы В может быть связана с множеством строк из таблицы А)

Нормализация

Нормализация представляет процесс разделения данных по отдельным связанным таблицам. Нормализация устраняет избыточность данных (data redundancy) и тем самым избежать нарушения целостности данных при их изменении, то есть избежать аномалий изменения (update anomaly).

В ненормализованной форме таблица может хранить информацию о двух и более сущностях. Также она может содержать повторяющиеся столбцы. Также столбцы могут хранить повторяющиеся значения. В нормализованной же форме каждая таблица хранит информацию только об одной сущности.

Нормализация предполагает применение нормальных форм к структуре данных.

Каждая нормальная форма (за исключением первой) подразумевает, что к данным уже была применена предыдущая нормальная форма.

База данных считается ***нормализованной***, если к ней применяется третья нормальная форма и выше.

Существуют 7 нормальных форм:

- Первая нормальная форма (1NF) предполагает, что сохраняемые данные на пересечении строк и столбцов должны представлять скалярное значение, а таблицы не должны содержать повторяющихся строк.

- Вторая нормальная форма (2NF) предполагает, что каждый столбец, не являющийся ключом, должен зависеть от первичного ключа.
- Третья нормальная форма (3NF) предполагает, что каждый столбец, не являющийся ключом, должен зависеть только от первичного ключа.
- Нормальная форма Бойса-Кодда (BCNF) является немного более строгой версией третьей нормальной формы.
- Четвертая нормальная форма (4NF) применяется для устранения многозначных зависимостей где столбец с первичным ключом имеет связь один-ко-многим со столбцом, который не является ключом. Эта нормальная форма устраняет некорректные отношения многие-ко-многим.

- Пятая нормальная форма (5NF) разделяет таблицы на более малые таблицы для устранения избыточности данных. Разбиение идет до тех пор, пока нельзя будет воссоздать оригинальную таблицу путем объединения малых таблиц.
- Шестая нормальная форма (domain key normal form / 6NF). Каждое ограничение в связях между таблицами должно зависеть только от ограничений ключа и ограничений домена, где домен представляет набор допустимых значений для столбца. Эта форма предотвращает добавление недопустимых данных путем установки ограничения на уровне отношений между таблицами, но не на уровне таблиц или столбцов.

Первая нормальная форма

Первая нормальная форма предполагает, что таблица не должна содержать повторяющихся столбцов или таких столбцов, которые содержат наборы значений.

Ненормализованная таблица в этом случае может содержать одну или несколько повторяющихся групп данных.

Повторяющаяся группа - это группа из одного или нескольких атрибутов таблицы, в которой возможно наличие нескольких значений для ключевого атрибута таблицы.

Том посещает курс по математике, который преподает профессор Смит.

Сэм посещает курс по математике, которые преподает профессор Смит.

Том посещает курс по языку JavaScript, который преподает ассистент Адамс.

Боб посещает курс по алгоритмам, который преподает ассистент Адамс.

Сэм имеет следующие электронный адрес sam@gmail.com и телефон +1235768789.

Таблица StudentCourses

StudentId	Name	Emails	CourseId	Course	Date	TeacherId	Teacher
1	Том		1	Математика	11/06/2017	1	Смит
1	Том		2	JavaScript	14/06/2017	2	Адамс
2	Сэм	sam@gmail.com sam@hotmail.com	3	Алгоритмы	12/06/2017	2	Адамс
3	Боб		1	Математика	13/06/2017	1	Смит

Таблица Emails

Email	StudentId
sam@gmail.com	2
sam@hotmail.com	2

Таблица StudentCourses

StudentId	Name	CourseId	Course	Date	TeacherId	Teacher
1	Том	1	Математика	11/06/2017	1	Смит
1	Том	2	JavaScript	14/06/2017	2	Адамс
2	Сэм	3	Алгоритмы	12/06/2017	2	Адамс
3	Боб	1	Математика	13/06/2017	1	Смит

Вторая нормальная форма

Во второй нормальной форме каждый столбец в таблице, который не является ключом, должен зависеть от ключа.

Эта форма применяется к тем таблицам, которые имеют составной первичный ключ, то есть где первичный ключ состоит из нескольких атрибутов.

Вторая нормальная форма применяется только к тем таблицам, которые находятся в первой нормальной форме. После применения второй формы все столбцы таблицы зависят от первичного ключа.

Таблица

StudentId	Name
1	Том
2	Сэм
3	Боб

Таблица

CourseId	Course	TeacherId	Teacher
1	Математика	1	Смит
2	JavaScript	2	Адамс
3	Алгоритмы	2	Адамс

Таблица

StudentId	CourseId	Date
1	1	11/06/2017
1	2	14/06/2017
2	3	12/06/2017
3	1	13/06/2017

Третья нормальная форма

Третья нормальная форма предполагает, что каждый столбец, не являющийся ключом, должен зависеть только от столбца, который является ключом.

Если столбец зависит не только от первичного ключа, то данный столбец находится не в той таблице, в которой он должен находиться, либо же является производным от других столбцов.

При применении третьей нормальной формы таблица должна находиться во второй нормальной форме. 3NF позволяет значительно снизить избыточность данных.

Таблица

TeacherId	Teacher
1	Смит
2	Адамс

Таблица Courses

CourseId	Course	TeacherId
1	Математика	1
2	JavaScript	2
3	Алгоритмы	2