

DB LEC 03

isagila

Собрано 09.09.2023 в 12:57



Содержание

1. Лекции	3
1.1. Лекция 23.09.01.	3
1.2. Лекция 23.09.08.	3

1. Лекции

1.1. Лекция 23.09.01.

TODO: Без понятия, буду ли писать лекции по БД, но пусть пока все остается как есть.

Информация бывает трех видов

1. Сигнал.

Например, для человека, который не знает какой-либо язык, текст на этом языке будет сигналом: т.е. можно понять, что какая-то информация была передана, но принять (понять) эту информацию нельзя.

2. Знание.

Студент, слушающий лектора и пишущий конспект, получает знания.

3. Данные.

Данные это формализованные знания. В отличие от знаний данные не искажаются в процессе передачи. Если лектор передаст студентам конспект лекции, который он написал сам, то он передаст именно данные.

Def 1.1.1. Данные это поддающиеся многократной интерпретации представления информации в формализованном виде пригодном для передачи, интерпретации и обработки.

Все данные проходят путь вида

Сбор → Обработка → Передача → Хранение → Представление

TODO: Да... вводная лекция она такая... не очень большая — не знаю, что еще можно добавить.

1.2. Лекция 23.09.08.

Уровни архитектуры данных

1. Внешний.

Решаем проблему представления данных пользователю (как агрегировать и детализировать данные?).

2. Концептуальный.

Занимаемся выделением сущностей, решаем проблемы безопасности (разрешения + ограничения). Определяем семантику (например договариваемся, что кредитный рейтинг будет храниться в виде числа от 1 до 5 и сопоставляем каждому числу некоторое словесное описание, которое записано в документации, но не в самой базе данных).

3. Внутренний.

Решаем проблемы хранения данных на физическом уровне.

Для простоты выделяют три модели данных, которые связаны с некоторыми описанными выше уровнями архитектуры данных.

1. Модель Сущность–Связь (внешний + концептуальный уровни).

2. Логическая (даталогическая модель) модель (все уровни).

3. Физическая модель (концептуальный + внутренний уровни).

Модель Сущность–Связь

Def 1.2.1. Сущность это множество экземпляров (реальных или абстрактных) однотипных объектов предметной области.

Def 1.2.2. Сущность называется сильной, если ее экземпляры могут существовать независимо. Слабые сущности могут существовать только при наличии одного или нескольких экземпляров сильной сущности.

Пример 1.2.3. Пусть у нас есть две сущности: Студент и Группа. Студент будет сильной сущностью, т.к. он может существовать и без группы, а Группа будет слабой сущностью, т.к. она не может существовать без студентов. Этот пример очень условный: деление сущностей на слабые и сильные зависит от обстоятельств. В некоторых случаях Группа вполне может быть сильной сущностью.

Каждая сущность обладает одним или несколькими атрибутами. Атрибуты делятся на

1. Простые (например дата рождения)

2. Составные (например адрес, если хранить город, улицу, дом и т.п. отдельно)

Замечание 1.2.4. Один и тот же атрибут (например, ФИО) может в разных случаях быть как простым (если его хранить как одну строчку), так и составным (если отдельно хранить фамилию, отдельно имя и отдельно отчество).

Также атрибуты можно делить по другому принципу.

1. Обязательные (например email в некоторых случаях может быть обязательным атрибутом).

2. Необязательные (например отчество может быть необязательным атрибутом).

Помимо этого, атрибуты делятся на

1. Однозначные (например дата рождения, она у каждого ровно одна).

2. Многозначные (например номер телефона, у кого-то может быть несколько номеров телефона).

Для каждого атрибута мы определяем домен, т.е. множество допустимых значений. Доменом может быть перечисление, множество всех натуральных чисел, регулярное выражение и т.д.

Чтобы показать отношения между сущностями, используются связи. На уровне Сущность–Связь они обычно несут семантическую нагрузку, например Студент **принадлежит** Группе. Существует три вида связи

1. Один-к-одному (1:1).

Обычно все связи этого вида это искусственно выделенные атрибуты. Зачем же тогда нужна эта связь? Рассмотрим на примере. Допустим, у нас есть две сущности: Студент и Паспорт. Логично, что у каждого студента один паспорт и у каждого паспорта один студент, т.е. связь вида 1:1. Почему же нельзя добавить паспорт в атрибуты сущности Студент? Так можно сделать, но связь вида 1:1 может быть удобна в целях безопасности: в таком случае все паспорта можно будет хранить отдельно от студентов и (например) дополнительно шифровать (шифровка же всей сущности Студент будет мало того, что излишне затратной, так еще и не очень нужной).

2. Один-ко-многим (1:N).

3. Многие-ко-многим (N:N).