

Лабораторная по БД №1

2025-02-13 15:43

Status: #theory

Tags: Базы данных Лабы

Лабораторная по БД №1

вариант 13108

Деннис Недри, ухмыляясь, спустился на первый этаж, кивнул охраннику и прошел дальше в подвал. Пройдя мимо электромобилей, стоявших аккуратными рядами, он приблизился к "джипу", поставленному у самой стены; "джип" работал на бензине. Садясь в него, Недри заметил, что на пассажирском сиденье лежит какая-то странная серая труба.

!переписать текст:

Сущности:

1. People:

- Name
- Passport
- Job

2. Violations:

- Violation type
- Fine amount
- Status

3. Booking:

- Check-in date
- Check-out date

4. Hotel room:

- Price
- Floor
- Room type

5. Parking spot:

- Spot number

- Spot type
- Is occupied

6. Car:

- Car brand
- Color
- Car number

7. Car parking:

- Start time
- End time

Ассоциативные сущности: Car Parking, Booking

Стержневые сущности: People, Hotel room, Car, Parking spot

Характеристические сущности: Violations

Стержневые (основные) сущности

Это ключевые объекты, которые имеют независимое существование и могут быть связаны с другими сущностями:

- **People (Люди)** – содержит информацию о людях (ID, имя, паспорт, работа).
- **Hotel_Room (Гостиничные номера)** – описывает номера отеля (ID, цена, этаж, тип).
- **Car (Автомобиль)** – содержит данные об автомобиле (бренд, цвет, номер, ID владельца).
- **Parking_Spot (Парковочное место)** – фиксирует данные о парковочном месте (номер, тип, занятость).

Ассоциативные сущности

Эти сущности связывают две или более стержневые сущности и сами по себе не могут существовать отдельно:

- **Booking (Бронирование)** – связывает людей (People) и номера в отеле (Hotel_Room), фиксируя даты заезда и выезда.
- **Car_Parking (Парковка автомобиля)** – связывает людей (People) с парковочными местами (Parking_Spot), указывая время начала и окончания парковки.

Характеристические (описательные) сущности

Эти сущности зависят от стержневых и служат для хранения их дополнительных характеристик:

- **Violations (Нарушения)** – содержит информацию о нарушениях, связанных с людьми (People), включая тип нарушения, сумму штрафа и статус.

У базы должен быть смысл, подумать, кому и зачем она будет нужна.

сделать инфологическую модель (типа шаблона, допустимы любые виды связей)

сделать даталогическую модель (постгрис)

в таблице по 5-6 данных

понятие закрытого мира - все, чего нет в бд, не существует

Теория

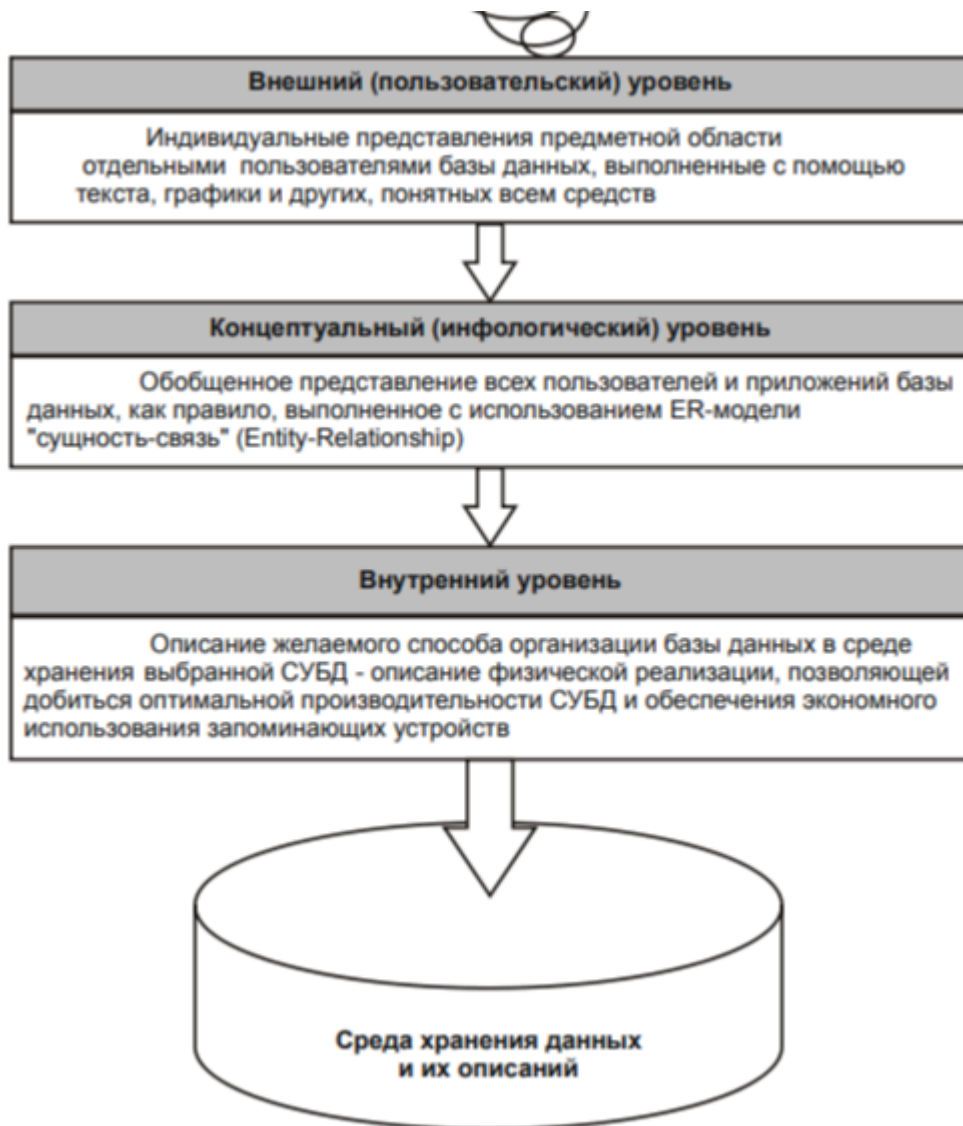
В 1975 году подкомитет SPARC (Standards Planning and Requirements Committee) американского национального института стандартов (American National Standards Institute, ANSI) выдвинул проект трехуровневой архитектуры СУБД (рис. 1.3):

1. Внешний (пользовательский).
2. Промежуточный (концептуальный).
3. Внутренний (физический).

В основе архитектуры ANSI/SPARC лежит концептуальный уровень. Этот уровень описывает данные и их взаимосвязи с наиболее общей точки зрения — концепции архитекторов (разработчиков) базы данных.

Внутренний уровень позволяет скрыть подробности физического хранения данных (носители, файлы ...) от концептуального уровня. Отделение внутреннего уровня от концептуального уровня обеспечивает, так называемую, физическую независимость данных.

На внешнем уровне описываются различные подмножества элементов концептуального уровня для представления данных различными пользователями и (или) их программами. Каждый пользователь получает в свое распоряжение часть представлений о данных, но полная концепция скрыта. Отделение внешнего уровня от концептуального уровня обеспечивает логическую независимость данных.



Так, между двумя сущностями, например, А и Б возможны четыре вида связей. Первый тип — связь "один-к-одному" (1:1): в каждый момент времени каждому представителю (экземпляру) сущности А соответствует 1 или 0 представителей сущности Б.

Второй тип — связь "один-ко-многим" (1:M): одному представителю сущности А соответствуют 0, 1 или несколько представителей сущности Б.

Так как между двумя сущностями возможны связи в обоих направлениях, то существует еще два типа связи "многие-к-одному" (M:1) и "многие-ко-многим" (M:M).

Стержневая сущность (стержень) — это независимая сущность, которая не является ни характеристикой, ни ассоциацией (см. далее).

Ассоциативная сущность (ассоциация) — это связь вида "многие-ко-многим" ("ко-многим" и т. д.) между двумя или более сущностями или экземплярами сущности. Ассоциации рассматриваются как полноправные сущности:

- они могут участвовать в других ассоциациях точно так же, как стержневые сущности;
- могут обладать свойствами, т. е. иметь не только набор ключевых атрибутов, необходимых для указания связей, но и любое число других атрибутов, характеризующих связь. Например, ассоциация СВИДЕТЕЛЬСТВООБРАКЕ (см. рис. 2.6) содержит ключевые атрибуты КОДМУЖЧИНЫ и КОДЖЕНЩИНЫ, а также уточняющие атрибуты ДАТАРЕГИСТРАЦИИ, МЕСТОРЕГИСТРАЦИИ, НОМЕР_СВИДЕТЕЛЬСТВА и т. д.

Характеристическая сущность (характеристика) — это связь вида "многие-к-одной" или "одна-к-одной" между двумя сущностями (частный случай ассоциации). Единственная цель характеристики в рамках рассматриваемой предметной области состоит в описании или уточнении некоторой другой сущности. Необходимость в них возникает в связи с тем, что сущности реального мира имеют иногда многозначные свойства.

Целостность (от англ. integrity — нетронутость, неприкосновенность, сохранность, целостность) — понимается как правильность данных в любой момент времени. Но эта цель может быть достигнута лишь в определенных пределах: СУБД не может контролировать правильность каждого отдельного значения, вводимого в базу данных (хотя каждое значение можно проверить на правдоподобность). Например, нельзя обнаружить, что вводимое значение 5 (представляющее номер дня недели) в действительности должно быть равно 3. С другой стороны, значение 9 явно будет ошибочным и СУБД должна его отвергнуть. Однако для этого ей следует сообщить, что номера дней недели должны принадлежать набору (1,2,3,4,5,6,7).

Поддержание целостности базы данных может рассматриваться как защита данных от неверных изменений или разрушений (не путать с незаконными изменениями и разрушениями, являющимися проблемой безопасности). Современные СУБД имеют ряд средств для обеспечения поддержания целостности (так же, как и средств обеспечения поддержания безопасности).

Выделяют три группы правил целостности.

- Целостность по сущностям. Не допускается, чтобы какой-либо атрибут, участвующий в первичном ключе, принимал неопределенное значение.
- Целостность по ссылкам. Значение внешнего ключа должно либо: быть равным значению первичного ключа ассоциируемой (характеризуемой) сущности; быть полностью неопределенным, т. е. каждое значение атрибута, участвующего во внешнем ключе, должно быть неопределенным.
- Целостность, определяемая пользователем. Для любой конкретной базы данных существует ряд дополнительных специфических правил, которые относятся к

ней одной и определяются разработчиком. Чаще всего контролируется: уникальность тех или иных атрибутов; диапазон значений (экзаменационная оценка от 2 до 5); принадлежность набору значений (пол "М" или "Ж").

DDL (Data Definition Language) и **DML** (Data Manipulation Language) — это два основных типа языковых конструкций, используемых в реляционных базах данных для работы с данными и их структурой.

DDL используется для определения и изменения структуры базы данных, таких как создание, изменение или удаление объектов (таблиц, индексов, представлений и т.д.). Основные команды DDL:

- CREATE
- ALTER
- DROP
- TRUNCATE
- RENAME

DDL команды являются **автокоммитными** — они сразу применяют изменения в базе данных, и их нельзя откатить (в большинстве СУБД).

DML (Data Manipulation Language)

DML используется для работы с данными внутри таблиц: добавление, изменение, удаление и выборка данных. Основные команды DML:

- SELECT
- INSERT
- UPDATE
- DELETE

DML команды не являются автокоммитными — их можно откатить с помощью команды **ROLLBACK** (если транзакция не завершена).

вкратце:

- **DDL** используется для работы со структурой базы данных (таблицы, индексы и т.д.).
- **DML** используется для работы с данными (добавление, изменение, удаление и выборка).

References