



Факультет программной инженерии и компьютерной техники
Информационные системы и базы данных

Лабораторная работа № 1
Вариант № 1133

Преподаватель: Харитоновна Анастасия Евгеньевна
Выполнил: Геллер Леонид Александрович
Группа: Р33301

Задание

Для выполнения лабораторной работы №1 необходимо:

1. На основе предложенной предметной области (текста) составить ее описание. Из полученного описания выделить сущности, их атрибуты и связи.
2. Составить инфологическую модель.
3. Составить даталогическую модель. При описании типов данных для атрибутов должны использоваться типы из СУБД PostgreSQL.
4. Реализовать даталогическую модель в PostgreSQL. При описании и реализации даталогической модели должны учитываться ограничения целостности, которые характерны для полученной предметной области.
5. Заполнить созданные таблицы тестовыми данными.

Для создания объектов базы данных у каждого студента есть своя схема. Название схемы соответствует имени пользователя в базе studs (sXXXXXX). Команда для подключения к базе studs:

```
psql -h pg -d studs
```

Каждый студент должен использовать свою схему при работе над лабораторной работой №1 (а также в рамках выполнения 2, 3 и 4 этапа курсовой работы).

Отчёт по лабораторной работе должен содержать:

1. Текст задания.
2. Описание предметной области.
3. Список сущностей и их классификацию (стержневая, ассоциация, характеристика).
4. Инфологическая модель (ER-диаграмма в расширенном виде - с атрибутами, ключами...).
5. Даталогическая модель (должна содержать типы атрибутов, вспомогательные таблицы для отображения связей "многие-ко-многим").
6. Реализация даталогической модели на SQL.
7. Выводы по работе.

Темы для подготовки к защите лабораторной работы:

1. Архитектура ANSI-SPARC
2. Модель "Сущность-Связь". Классификация сущностей. Виды связей. Ограничения целостности.
3. DDL
4. DML

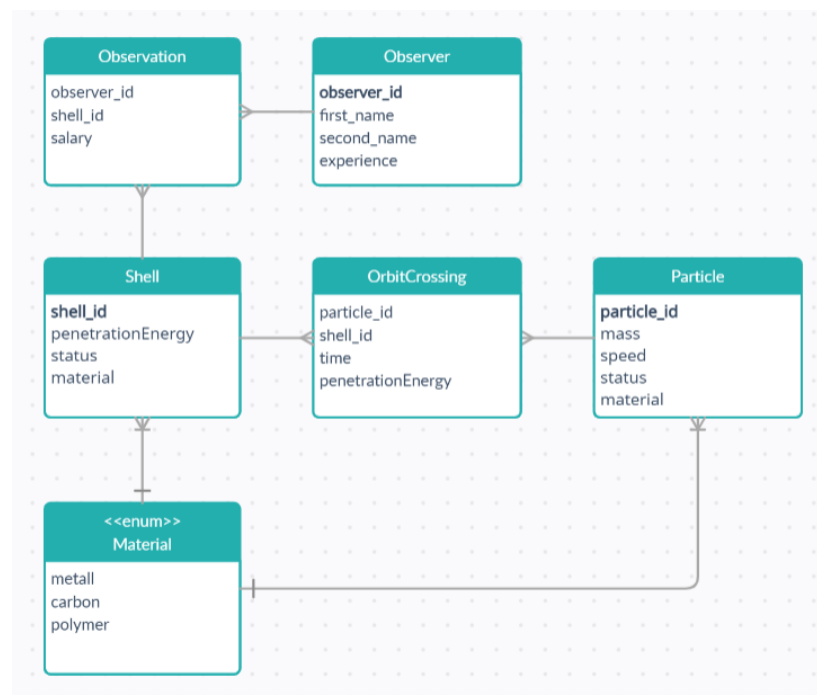
Предметная область

Прежде всего он заметил небольшое оплавление металла оболочки - пятнышко чуть более сантиметра диаметром, с крошечным кратером посредине. Частица космической пыли, ударившая туда на скорости более полутора тысяч километров в секунду, была, наверно, меньше булавочной головки, и огромная кинетическая энергия, несомая ею, мгновенно превратила ее в пар. Как это часто бывает, кратер выглядел так, будто он образован взрывом внутри оболочки: на таких скоростях материалы ведут себя странно и привычные законы ньютоновой механики редко применимы.

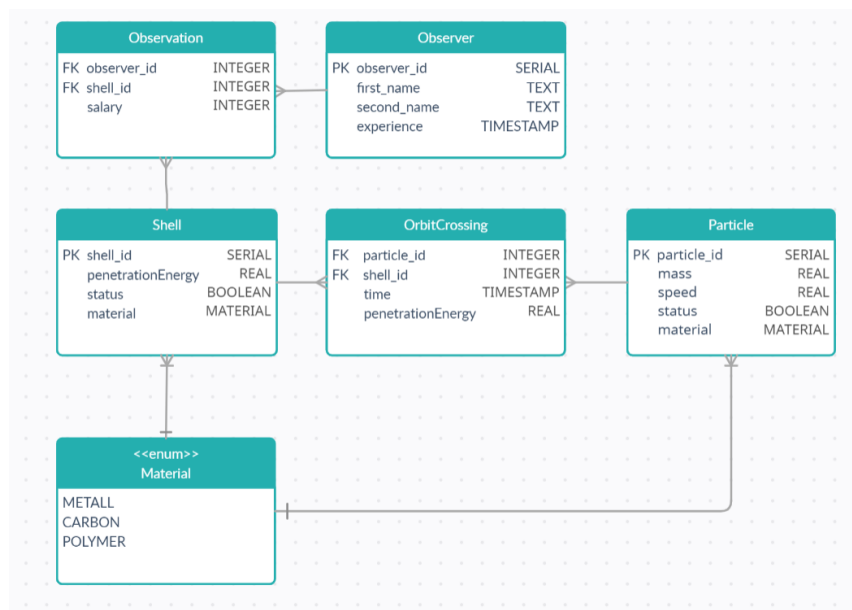
Описание предметной области

Первой сущностью, которую очевидно можно выделить, является **ЧАСТИЦА** космической пыли (id, масса, скорость, материал, и статус (в свободном полете, застряла в оболочке при попадании)), которая оставляет на **ОБОЛОЧКЕ** (id, энергия (необходимая для пробития), статус (f – герметична(по умолчанию), t - пробита), материал) некоторое повреждение при **ПЕРЕСЕЧЕНИИ_ОРБИТЫ** частицы с оболочкой (частица, оболочка, время столкновение, энергия столкновения). На характер повреждения влияет соответствие **МАТЕРИАЛОВ** (металл, углеродное покрытие, полимер) частицы и оболочки. Повреждение может быть ликвидировано **НАБЛЮДАТЕЛЕМ** (id, имя, фамилия, опыт (время принятия на работу)), контролирующего состояние оболочки. Наблюдатель связан с оболочкой посредством **НАБЛЮДЕНИЯ** (наблюдатель, оболочка, зарплата за обслуживание).

Инфологическая модель



Даталогическая модель



Реализация

- характеристическая

```
CREATE TYPE material AS ENUM ('metall', 'carbon', 'polymer');
```

- стержневая

```
CREATE TABLE particle (
    particleId SERIAL PRIMARY KEY,
    mass REAL NOT NULL,
    speed REAL NOT NULL,
    status BOOLEAN NOT NULL DEFAULT FALSE,
    material MATERIAL NOT NULL
);
```

- ассоциативная

```
CREATE TABLE orbitCrossing (
    penetrationEnergy REAL NOT NULL,
    time TIMESTAMP NOT NULL,
    particleId INTEGER REFERENCES particle ON DELETE CASCADE,
    shellId INTEGER REFERENCES shell ON DELETE CASCADE,
    PRIMARY KEY(particleId, shellId)
```

);

- стержневая

```
CREATE TABLE shell (  
    shellId SERIAL PRIMARY KEY,  
    penetrationEnergy REAL NOT NULL,  
    isHoled BOOLEAN DEFAULT FALSE,  
    material MATERIAL NOT NULL,  
);
```

- стержневая

```
CREATE TABLE observer (  
    observerId SERIAL PRIMARY KEY,  
    firstName TEXT,  
    lastName TEXT NOT NULL,  
    experience TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

- ассоциативная

```
CREATE TABLE observation (  
    observerId INTEGER REFERENCES observer ON DELETE CASCADE,  
    shellId INTEGER REFERENCES shell ON DELETE CASCADE,  
    PRIMARY KEY(observerId, shellId),  
    salary INTEGER NOT NULL  
);
```

Заполнение данными

```
INSERT INTO particle (mass, speed, material) VALUES (75.1, 12, 'carbon');
```

```
INSERT INTO shell (penetrationEnergy, material) VALUES (444, 'carbon');
```

```
INSERT INTO orbitCrossing (penetrationEnergy, time, particleId, shellId) VALUES (500, TIMESTAMP '2031-05-16 15:36:38', 1, 1);
```

```
INSERT INTO observer (firstName, lastName) VALUES ('Evgeniy', 'Tsopa');
```

```
INSERT INTO observer (firstName, lastName, experience) VALUES ('Evgeniy', 'Tsopa', TIMESTAMP '2011-05-16 15:36:38');
```

```
INSERT INTO observation(observerId, shellId, salary) VALUES (2, 1, 100000);
```

Вывод

Архитектура ANSI-SPARC определяет принцип, согласно которому рекомендуется строить системы управления базами данных (СУБД). Выделяют 3 уровня системы:

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

В основе архитектуры ANSI-SPARC лежит концептуальный уровень. В современных СУБД он может быть реализован при помощи представления. Концептуальный уровень описывает данные и их взаимосвязи с наиболее общей точки зрения, — концепции архитекторов базы, используя реляционную или другую модель.

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

На внешнем уровне описываются различные подмножества элементов концептуального уровня для представлений данных различным пользовательским программам. Каждый пользователь получает в своё распоряжение часть представлений о данных, но полная концепция скрыта. Отделение внешнего уровня от концептуального обеспечивает логическую независимость данных.

ER-модель (от англ. Entity-Relationship model, модель «сущность — связь») — модель данных, позволяющая описывать концептуальные схемы предметной области. ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Во время проектирования баз данных происходит преобразование схемы, созданной на основе ER-модели, в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.).

ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена диаграмма «сущность-связь» (англ. Entity-Relationship diagram, ERD, ER-диаграмма).

Понятия «ER-модель» и «ER-диаграмма» часто не различают, хотя для визуализации ER-моделей могут быть использованы и другие графические нотации, либо визуализация может вообще не применяться (например, использоваться текстовое описание).

Классификация сущностей

Стержневая сущность – это независимая сущность

Ассоциативная сущность (ассоциация) – это связь вида "многие-ко-многим" между сущностями или экземплярами сущности. Ассоциации рассматриваются как полноправные сущности

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.

Обозначающая сущность или обозначение – это связь вида "многие-к-одной" или "одна-к-одной" между двумя сущностями и отличается от характеристики тем, что не зависит от обозначаемой сущности.

Data Definition Language (DDL) (язык описания данных) — это семейство компьютерных языков, используемых в компьютерных программах для описания структуры баз данных. Наиболее популярным языком DDL является SQL, используемый для получения и манипулирования данными в РСУБД, и сочетающий в себе элементы DDL и DML.

Функции языков DDL определяются первым словом в предложении (часто называемом запросом), которое почти всегда является глаголом. В случае с SQL это глаголы — «create» («создать»), «alter» («изменить»), «drop» («удалить»). Эти запросы или команды часто смешиваются с другими командами SQL, в связи с чем DDL не является отдельным компьютерным языком.

Запрос «create» используется для создания базы данных, таблицы, индекса, представления или хранимой процедуры. Запрос «alter» используется для изменения существующего объекта базы данных (таблицы, индекса, представления или хранимой процедуры) или самой базы данных. Запрос «drop» используется для удаления существующего объекта базы данных (таблицы, индекса, представления или хранимой процедуры) или самой базы данных. И наконец, в DDL существуют понятия первичного и внешнего ключа, которые осуществляют соблюдение целостности данных.

Data Manipulation Language (DML) (язык управления (манипулирования) данными) — это семейство компьютерных языков, используемых в компьютерных программах или пользователями баз данных для получения, вставки, удаления или изменения данных в базах данных. Наиболее популярным языком DML является SQL, используемый для получения и манипулирования данными в РСУБД. Языки DML изначально использовались только компьютерными программами, но с появлением SQL стали также использоваться и людьми.

Функции языков DML определяются первым словом в предложении (часто называемом запросом), которое почти всегда является глаголом. В случае с SQL эти глаголы — «select» («выбрать»), «insert» («вставить»), «update» («обновить»), и «delete» («удалить»). Это превращает природу языка в ряд обязательных утверждений (команд) к базе данных.