

Лабораторная работа №1

Проектирование реляционной базы данных

Описание работы

Согласно выбранному варианту вам необходимо спроектировать реляционную БД под управлением современной полноценной СУБД (например PostgreSQL, MySQL, Oracle). *Использование решений по типу H2 и SQLite, разворачивающих БД в памяти в виде файлика - запрещается.*

Ваша БД должна содержать таблицы отображающие сущности выбранной предметной области. БД должна содержать несколько таблиц (хотя бы больше 4-5).

Замечание

Ваши варианты написаны довольно абстрактно, что дает вам свободу в проявлении фантазии и смекалки. Перечень таблиц и сущностей не указан явно – подумайте, что там может быть необходимо в поставленной задаче. Таблиц не должно быть много или слишком много, ведь как минимум за каждую из них необходимо отчитаться и реализовать CRUD :) **В среднем всё везде укладывается в 6-10 таблиц, это немного...**

В худшем случае вы можете спросить помощь у преподавателя.

Выполнение

Какой подход вы будете использовать: code-first (написать код и на его основе сгенерировать БД) или sql-first (написать собственноручно БД и под неё уже делать код) не имеет значения.

В любом случае выходом на данной ЛР должна быть спроектированная БД, которая соответствует нормальным формулам, содержит адекватные таблицы и связи. Также при связывании таблиц необходимо добавить ограничения и каскады.

Всё это выполнимо средствами SQLAlchemy.

В качестве ключей разрешается использовать INT или UUID.

Содержание отчета

1. Выбранный вариант
2. Описание таблиц – что за таблица, какие поля, связи и каскады.
3. ERD-диаграмма
4. SQL-скрипт для генерации данной БД

Про диаграмму и скрипт

В среднем любая система работы с БД, будь БД сгенерирована или написана вручную может сгенерировать вам ERD-диаграмму и SQL-скрипт (pgAdmin точно может).

Пример описания таблиц

building – строение

Таблица building содержит информацию о строениях.



Поле	Тип данных	Описание	Ограничения
<u>id</u>	UUID (PK)	Уникальный идентификатор записи	PRIMARY KEY, NOT NULL
<u>kks_code</u>	VARCHAR	Код KKS строения	NOT NULL
<u>pbs_code_id</u>	UUID (FK → <u>pbs_code.id</u>)	PBS-код строения	NOT NULL, FOREIGN KEY, ON DELETE DENY, ON DELETE UNLINK
<u>description</u>	TEXT	Описание строения	
<u>created_by</u>	VARCHAR (255)	Пользователь, создавший запись	
<u>created_date</u>	TIMESTAMP WITH TIME ZONE	Дата и время создания записи	
<u>deleted_by</u>	VARCHAR (255)	Пользователь, удаливший запись	
<u>deleted_date</u>	TIMESTAMP WITH TIME ZONE	Дата и время удаления записи	

связи building

- pbs_code_id (FK → pbs_code.id) - PBS-код строения. Поведение при удалении: ON DELETE UNLINK, ON DELETE DENY.

Пример описания таблиц со связью n:n

tag - проектные позиции

Таблица **tag** описывает проектные позиции, включая их код, описание, связи с другими объектами. Имеет связи многие-ко-многим с проектными условиями (**design_condition**) и исходными событиями (**initial_event**).



Поле	Тип данных	Описание	Ограничения
<u>id</u>	UUID (PK)	Уникальный идентификатор	PRIMARY KEY, NOT NULL
<u>kks_code</u>	VARCHAR (255)	KKS код позиции проекта	NOT NULL
<u>description</u>	TEXT	Описание позиции	
<u>building_id</u>	UUID (FK → <u>building_id</u>)	Строение к которому относится данная проектная позиция	FOREIGN KEY, ON DELETE UNLINK
<u>system_id</u>	UUID (FK → <u>system_id</u>)	Система к которой относится данная проектная позиция	NOT NULL, FOREIGN KEY, ON DELETE UNLINK
<u>pbs_code_id</u>	UUID (FK → <u>pbs_code_id</u>)	PBS код данной проектной позиции	NOT NULL, FOREIGN KEY, ON DELETE UNLINK
<u>revision</u>	VARCHAR (255)	Номер ревизии	NOT NULL
<u>created_by</u>	VARCHAR (255)	Пользователь, создавший запись	
<u>created_date</u>	TIMESTAMP WITH TIME ZONE	Дата и время создания записи	
<u>deleted_by</u>	VARCHAR (255)	Пользователь, удаливший запись	
<u>deleted_date</u>	TIMESTAMP WITH TIME ZONE	Дата и время удаления записи	

- building_id (FK → building_id) - строение к которому относится данная проектная позиция. Поведение при удалении: UNLINK.
- system_id (FK → system_id) - система к которой относится данная проектная позиция. Поведение при удалении: UNLINK.
- pbs_code_id (FK → pbs_code_id) - PBS код данной проектной позиции. Поведение при удалении: UNLINK.

1

Таблица связи: **tag_design_condition_link**

Поле	Тип данных	Описание	Ограничения
<u>tag_id</u>	UUID (FK → <u>tag_id</u>)	Ссылка на таблицу проектных позиций <u>tag</u>	NOT NULL, FOREIGN KEY, ON DELETE UNLINK
<u>design_condition_id</u>	UUID (FK → <u>design_condition_id</u>)	Ссылка на таблицу проектных состояний <u>design_condition</u>	NOT NULL, FOREIGN KEY, ON DELETE UNLINK