

ЗАДАНИЕ

Технические требования к заданию:

1. Выберите область, которая вам действительно интересна.
2. Сначала набросайте сущности (таблицы) на бумаге, продумайте связи (один-ко-многим, многие-ко-многим).
3. Нормализуйте БД: избегайте повторяющихся данных.
4. Используйте внешние ключи (FOREIGN KEY) для связей.
5. Начните с простого: 4-5 таблиц вполне достаточно для проекта.
6. SQLAlchemy ORM поможет сделать код чище, но можно начать и с чистого sqlite3.

Области на выбор представлены ниже:

1. Система управления космической экспедицией

Описание: БД для планирования миссии на Марс (или другую планету). Нужно учитывать экипаж, ресурсы, научные задачи и оборудование.

Примерные таблицы (идеи):

- Crew (Экипаж) - астронавты, их должность, специализация, дата рождения.
- Spaceship_Modules (Модули корабля/базы) - жилой модуль, лаборатория, оранжерея, их состояние.
- Resources (Ресурсы) - вода, еда, кислород, топливо. Учет расходов и остатков.
- Scientific_Tasks (Научные задачи) - описание задачи, приоритет, статус выполнения.
- Equipment (Оборудование) - марсоходы, скафандры, инструменты. Привязка к модулю и состоянию.
- Task_Assignments (Назначения задач) - кто какую задачу выполняет и когда.

Связи: Астронавты назначаются на задачи. Оборудование используется для задач. Ресурсы потребляются экипажем и оборудованием.

Примеры запросов:

1. Показать всех астронавтов, кто не назначен на задачи высокого приоритета на следующей неделе.
2. Рассчитать суммарный расход воды на предстоящий месяц для планирования запасов.
3. Найти всё оборудование, находящееся в неисправном состоянии, и ответственного за него астронавта.

2. База данных киберспортивной команды

Описание: Управление профессиональной киберспортивной организацией: игроки, тренеры, турниры, стратегии.

Примерные таблицы (идеи):

- Players (Игроки) - никнейм, роль в игре, дата вступления в команду, зарплата.
- Coaches (Тренеры) - специализация (тактика, психология и т.д.).
- Tournaments (Турниры) - название, дата, призовой фонд, занятое место.
- Heroes (Персонажи игры) - название, роль, сила/слабость (если игра, напр., Dota 2).
- Matches (Матчи) - ссылка на турнир, противник, результат, длительность.
- Training_Schedules (Расписание тренировок) - дата, цель, участники.

Связи: Игроки участвуют в матчах и турнирах. Тренеры проводят тренировки. У каждого игрока есть "пул" героев, которых он лучше всего играет.

Примеры запросов:

1. Определить самого результативного игрока (по Win Rate) за последний сезон.
2. Вывести расписание тренировок на неделю с указанием ответственного тренера.
3. Найти всех героев, которых команда никогда не выбирала на крупных турнирах.

3. Платформа для организации фестиваля (музыкального, фуд-феста)

Описание: Система для учета участников, локаций, расписания и логистики мероприятия.

Примерные таблицы (идеи):

- Participants (Участники) - музыканты, стенды с едой, мастер-классы. Тип, контактное лицо.
- Stages_Locations (Сцены/Локации) - название, вместимость, тип (сцена, палатка, площадка).
- Schedule (Расписание) - время начала и окончания, участник, локация.
- Staff (Персонал) - волонтеры, охрана, техники. Смена, роль.
- Equipment_Rental (Аренда оборудования) - колонки, стулья, гримерки. Стоимость, сроки, поставщик.

Связи: Участники привязаны к событиям в расписании. События происходят в локациях. Персонал назначается на смены и локации.

Примеры запросов:

1. Показать "расписание дня" для конкретной сцены в хронологическом порядке.
2. Найти все временные интервалы, когда конкретный волонтер не занят.
3. Посчитать общую стоимость аренды всего звукового оборудования за время фестиваля.

4. Сервис проката электромобилей / самокатов

Описание: Система учета транспорта, его местоположения, аренды и клиентов.

Примерные таблицы (идеи для студентов):

- Vehicles (Транспорт) - уникальный ID, тип (электросамокат, электровелосипед), модель, уровень заряда, текущий статус (свободен/арендован/на обслуживании).
- Parking_Zones (Зоны парковки) - адрес, вместимость, количество свободных мест.
- Clients (Клиенты) - данные, рейтинг, баланс.
- Rentals (Аренды) - клиент, транспорт, время начала, время окончания, цена, стартовая и конечная зоны парковки.
- Service_Log (Журнал обслуживания) - транспорт, дата, тип неисправности, статус.

Связи: Транспорт припаркован в зоне. Аренда связывает клиента и транспорт между двумя зонами.

Примеры запросов:

1. Найти все свободные электросамокаты с зарядом >70% в указанной зоне парковки.
2. Определить топ-3 самых популярных зон парковки (куда чаще всего оставляют транспорт).
3. Рассчитать среднюю длительность поездки клиентов с рейтингом выше 4.5.

5. База знаний для алхимика / зачарователя (в стиле фэнтези-игр)

Описание: создать "научную" базу рецептов, ингредиентов и эффектов.

Примерные таблицы (идеи):

- Ingredients (Ингредиенты) - название, тип (растение, минерал, часть существа), редкость, базовые свойства.
- Recipes (Рецепты) - название зелья/заклинания, тип (эликсир, яд, усиление), сложность.
- Recipe_Components (Компоненты рецепта) - связь рецепта и ингредиента + необходимое количество.
- Effects (Эффекты) - что дает рецепт (лечение, невидимость, урон огнем).
- Experiments (Эксперименты) - журнал попыток создания: дата, рецепт, успех/неудача, заметки.

Связи: Рецепт состоит из нескольких ингредиентов. Рецепт приводит к одному или нескольким эффектам.

Примеры запросов:

1. Найти все рецепты зелий лечения, для которых нужен ингредиент "Корень мандрагоры".
2. Показать все неудачные эксперименты за последний месяц и использованные в них рецепты.
3. Определить самый "дорогой" (по сумме редкости ингредиентов) рецепт.

ПО: VS Code (по желанию другое)

Формат подачи файлов: архив с файлами с расширением .py /файл с расширением .py