Выделенное является ключом. Описаны основные отношения чернового варианта проектирования базы данных.

User

1. **Id**
2. Nickname
3. Password
4. Role – либо пользователь, либо модератор, либо редактор. Соответственно: User, Moderator, Editor.

Task

1. **Id**
2. Rating – рейтинг задачи.
3. Name
4. DataLink – относительная ссылка на область памяти на серваке, где хранится архив всех необходимых для задачи данных (проверяющая программа; набор тестов; условие; авторское решение; разбор).

Section

1. **Id**
2. Name

TaskInSection – задача принадлежит соответствующему разделу

1. **TaskId**
2. **SectionId**

RatingInSection – рейтинг пользователя в соответствующей секции.

1. **UserId**
2. **SectionId**
3. Rating – рейтинг может принимать некоторое значение null, означающее, что пользователь еще не прошел начальное тестирование по этому разделу.

Ограничение целостности на это отношение: user может вступить в отношение тогда и только тогда, когда user.Role == User.

SystemMessage – системное сообщение от модератора пользователю.

1. **Id**
2. UserId
3. ModeratorId
4. Text
5. SendingTime – время посылки.
6. WasRead – булево значение, прочитал ли пользователь уже это сообщение.

Ограничение целостности на это отношение: User с UserId должен иметь Role == User, User с ModeratorId должен иметь Role == Moderator.

FeedbackMessage – фидбэк пользователя.

1. **Id**
2. UserId
3. Text
4. SendingTime – время посылки.

ModeratorReadFeedback – информация о том, что модератор прочитал соответствующий фидбэк.

1. **FeedbackId**
2. **ModeratorId**

Submission – посылка пользователем решения по задаче.

1. **TaskId**
2. **UserId**
3. CodeLink – ссылка на код пользователя.
4. ProgrammingLanguage – выбранный язык программирования пользователем.
5. Status – статус посылки (тут нужно подумать, какие статусы можно выделить. Примерами могут быть «в посылке», «компилируется», «проверяется на таком-то тесте». В целом, если задача уже проверена, в статусе можно хранить вердикт проверки (над тем, какой формат у вердикта тоже можно подумать), а можно вердикт и выделить отдельно).
6. SendingTime – время, когда решение отослано.

Тут еще надо подумать, мб нужно сохранять рейтинг пользователя и задачи на момент решения, как он изменился для задачи и системы после решения. Да и в целом, вопрос с посылками довольно сырой. Потому что сейчас непонятно, то ли каждая посылка является значимой, то ли пользователь делает много — много посылок нахаляву, а потом сдает задачу, либо пропускает ее (получая и теряя рейтинг, соответственно). Во втором случае стоит ввести еще одно отношение, хранящее инфу о том, что пользователь какие-то задачки «подтвердил».

ProgrammingLanguage – инфа о доступных в системе ЯП.

1. **Id**
2. Name
3. … ссылка на директорию на серваке, где хранится компилятор / интерпретатор (не описал адекватно, потому что для компилируемых ЯП (с++, java) необходим компилятор, а возможность запуска отдельна от компилятора; для интерпретируeмых (python) интерпретатора достаточно, чтобы запустить прогу, пока не придумал, как эту инфу в базе хранить).
4. Options -- информация о том, с какими параметрами производится компиляция / интерпретация.

Также мы хотели для каждого раздела давать пользователю не больше, чем одну задачу, пока он ее не пропустил или решил, перейдя, тем самым, к следующей. Это тоже надо хранить.

Что-то нужно делать с начальным тестированием по каждому разделу. Данных, которые можно и нужно хранить много, единолично принимать такие решения не считаю возможным.

Короче говоря, вариант довольно сырой, многое может еще измениться, добавиться.