

* <https://www.ca.com/us/register/forms/ca-erwin-data-modeler-community-edition-evaluation-software.aspx> - ссылка на ERwin Data modeler (Community Edition)
* <http://erwin.com/products/data-modeler/community-edition/> - ссылка на описание

// Данный документ **НЕ ЯВЛЯЕТСЯ ФИНАЛЬНОЙ ВЕРСИЕЙ** //

ЗАМЕЧАНИЯ:

Связь (Parent - children) является *идентифицирующей*, если любой экземпляр сущности children не может существовать без связи с экземпляром сущности children (иными словами, у любого ребенка существует родитель).

Связь (Parent - children) является *неидентифицирующей*, если она не является идентифицирующей.

Различают зависимые и независимые сущности. На ER – диаграмме зависимая сущность изображается прямоугольником со скругленными углами.

Различают три уровня логической модели, отличающихся по глубине представления информации о данных:

* **Диаграмма сущность-связь (Entity Relationship Diagram (ERD));**
* Модель данных, основанная на ключах (Key Based model (KB));
* Полная атрибутивная модель (Fully Attributed model (FA)).

**Диаграмма сущность - связь представляет собой модель данных верхнего уровня. Она включает сущности и взаимосвязи, отражающие основные бизнес-правила предметной области.**

Согласно синтаксису **IDEF1X** имя атрибута должно быть уникально в рамках модели (а не только в рамках сущности!).

// Определиться с доменами атрибутов. Домен атрибута будет использоваться при определении типа колонки на уровне физической модели.

// Дать текстовой описание каждой сущности в закладке *Definition*

// Дать текстовое определение каждой связи в закладке *Definition*

// Считаю, что целесообразно отделить модераторов от пользователей на логическом уровне в целях производительности запросов и упрощении схемы (пусть нам нужно найти всех модераторов; чтобы их найти нужно пройти по всем USER-ам и посмотреть их статус (атрибут Role), а теперь представьте сколько пользователей в системе развития АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ)

// Возможно подискутировать о выборе первичного ключа (возможно он будет сложным)

// Привести модель к нормальной форме

1. Связь User – SystemMessage *идентифицирующая*, т.е. экземпляр сущности SystemMessage не может существовать без экземпляра User (системное сообщение отправляет один из модераторов, само по себе оно существовать не может).
2. Связь SystemMessage – User *неидентифицирующая*, т.к. любой User существует вне зависимости от того, создано ли SystemMessage или нет.
3. Аналогичная пунктам 1 и 2 ситуация со связью User - FeedbackMessage.
4. Связь Section - Task *идентифицирующая*, т.к. любая задача (любой экземпляр сущности Task) принадлежит какой-нибудь секции (экземпляр сущности Section), и при том только одной.
5. Связь Task – Section *неидентифицирующая*, т.к. любая секция (экземпляр сущности Section) может существовать вне зависимости от того, принадлежит ей хотя бы одна задача (экземпляр сущности Task) или нет.
6. Связи Moderator/FeedbackMes и FeedbackMes/Moderator являются идентифицирующими для сущности ModeratorReadFM (имеется в виду, что экземпляр сущности ModeratorReadFM не может существовать отдельно без связи с Moderator и Feedback; экземпляр сущности ModeratorReadFM создается лишь тогда, когда один из модераторов прочитывает фидбэк, и содержит информацию о фидбэке и модераторе, который прочитал этот фидбэк).
7. Связи User/Section и Section/User являются идентифицирующими для сущности RatingInSection (имеется в виду, что экземпляр сущности RatingInSection не может существовать без связи с User и Section; экземпляр сущности RatingInSection содержит информацию о рейтинге пользователя в секции).
8. Связи User/Task и Task/User являются идентифицирующими для сущности Submission (экземпляр сущности Submission содержит информацию о результате решения задачи Y пользователем X).