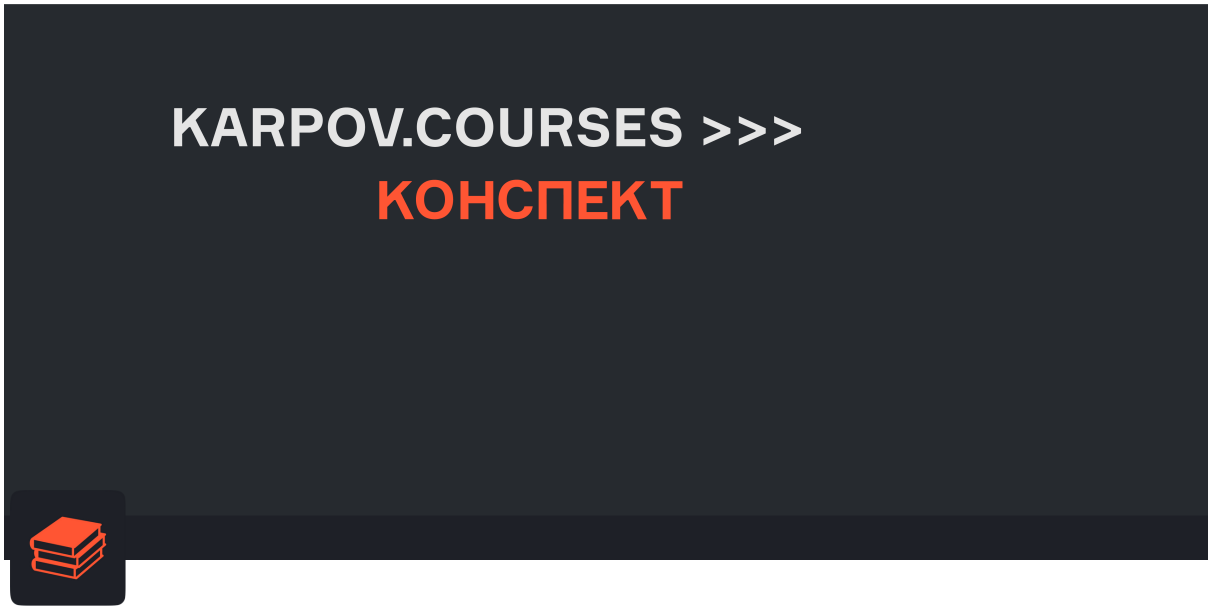


> Конспект > 3 урок > Dimensional modeling: Практикум

> Оглавление

> Draw.io

Draw.io – бесплатное приложение, предназначенное для моделирования диаграмм и блок-схем бизнес-процессов. Присутствует возможность интеграции с Google Документами, Dropbox, OneDrive, JIRA, Confluence, Chrome и GitHub.

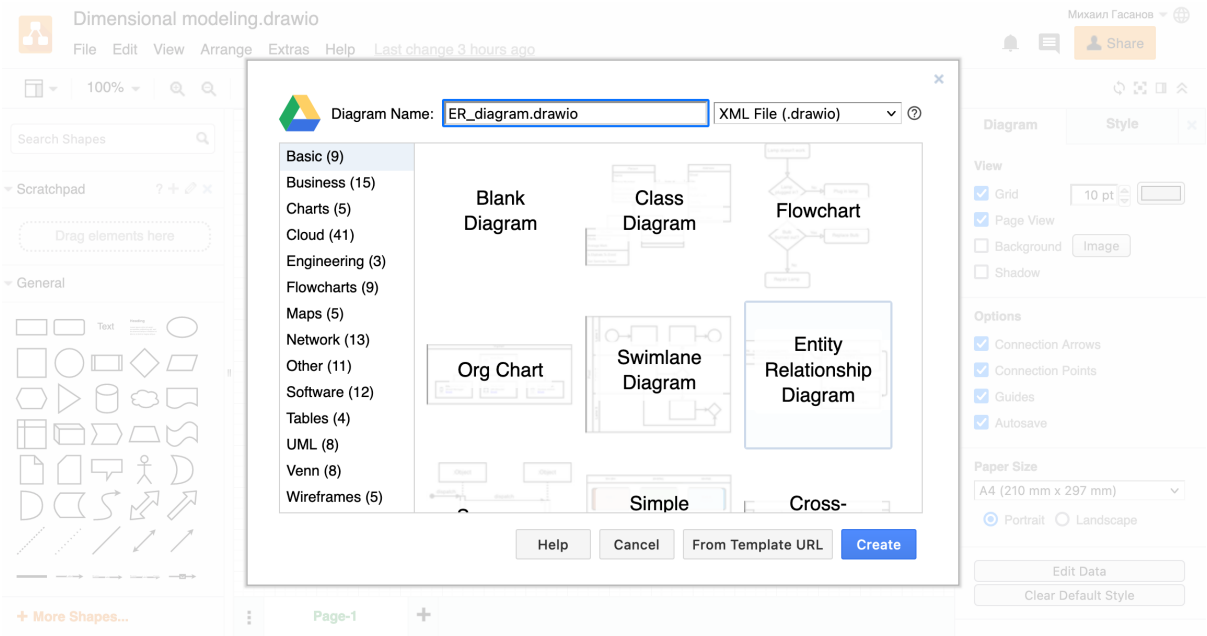
Помимо веб-версии существует программа для установки на ПК, которая поддерживает ОС Windows, MacOS и Linux.

Приложение будет полезно для специалистов, которым требуется инструмент для построения схем бизнес-процессов. Сервис подходит для специалистов, чья деятельность
связана с созданием презентаций и баз данных, построением инженерных и сетевых схем, проектированием программного обеспечения.

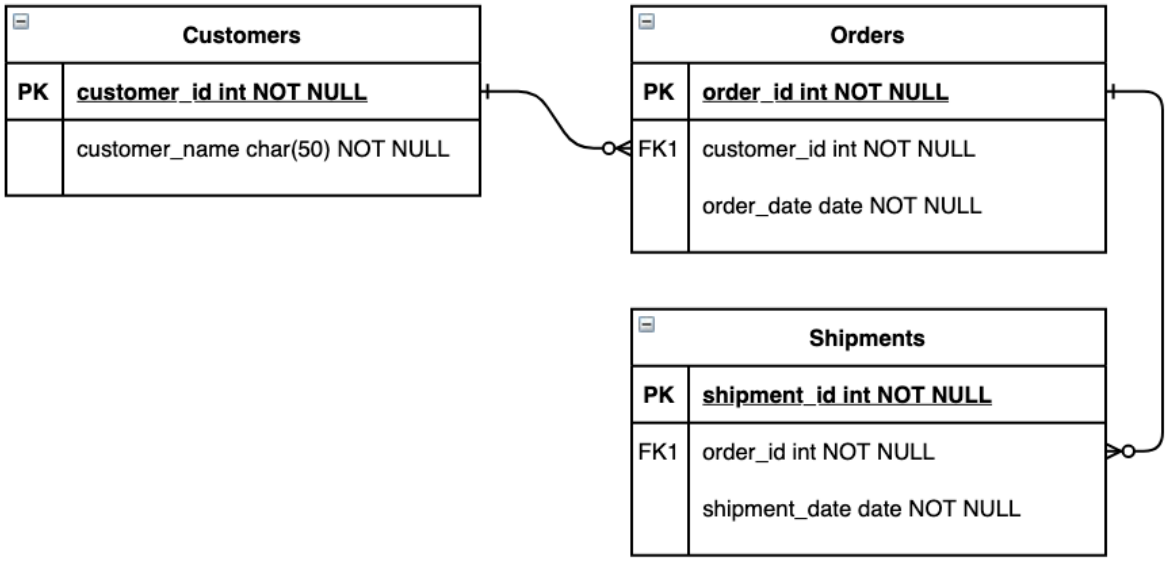
Draw.io позволяет визуализировать ER-диаграммы, которые мы обсудили на предыдущих лекциях. Существует множество редакторов для создания ER-диаграмм, основное отличие платных (проприетарных) версий от бесплатных заключается в возможности проведения reverse- или forward-engineering. Reverse-engineering позволяет получить ER-диаграмму на основе существующей БД. Forward-engineering, позволяет создать БД из построенных ER-диаграмм.

> Создание ER-диаграммы

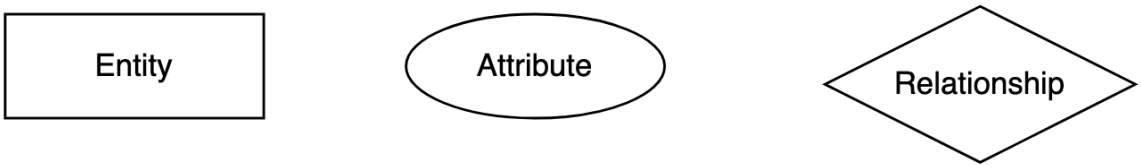
Для создания ER-диаграммы выбираем: `File > New...` , в окне выбора диаграмм выбираем `Entity Relationship Diagram`



Draw.io автоматически создаст ER-диаграмму с покупателями, заказами и отгрузками.

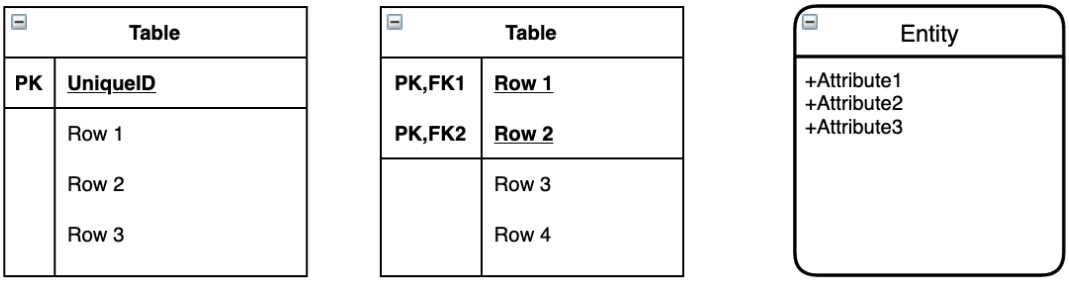


В разделе Entity Relationship можно выбрать различные представления сущностей, таблиц и атрибутов.



Нотация Питера Чена

Нотация Питера Чена. Эта нотация была представлена Питером Ченом, являющимся
одним из основоположников реляционных баз данных, и долгое время применялась для графической интерпретации предметной области в терминах сущностей и связей, иллюстрирующих ее абстрактное представление на логическом и концептуальном уровнях.



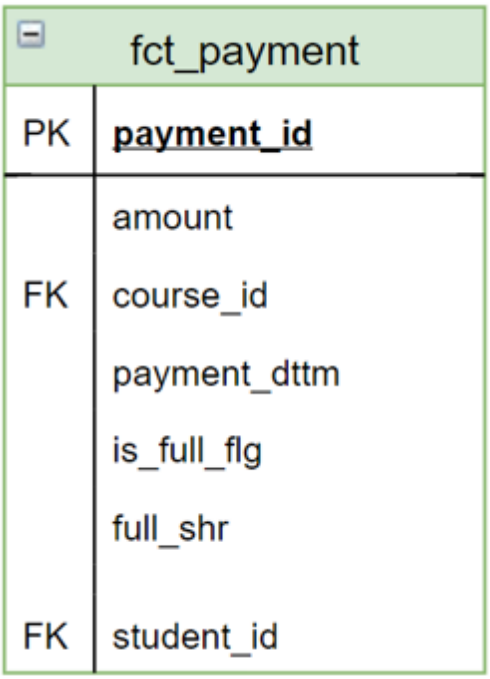
Нотация Crow's foot.

Нотация Crow's foot является одной из наиболее известных в разработке баз данных, отражающей уровень логического представления базы данных с обозначением некоторых компонентов модели базы данных в графическом виде, облегчая, тем самым, отображение диаграммы в рабочем пространстве. Модели такого типа менее громоздки по сравнению с моделями в нотации Питера Чена.

> Создание ER-диаграммы по Кимбаллу

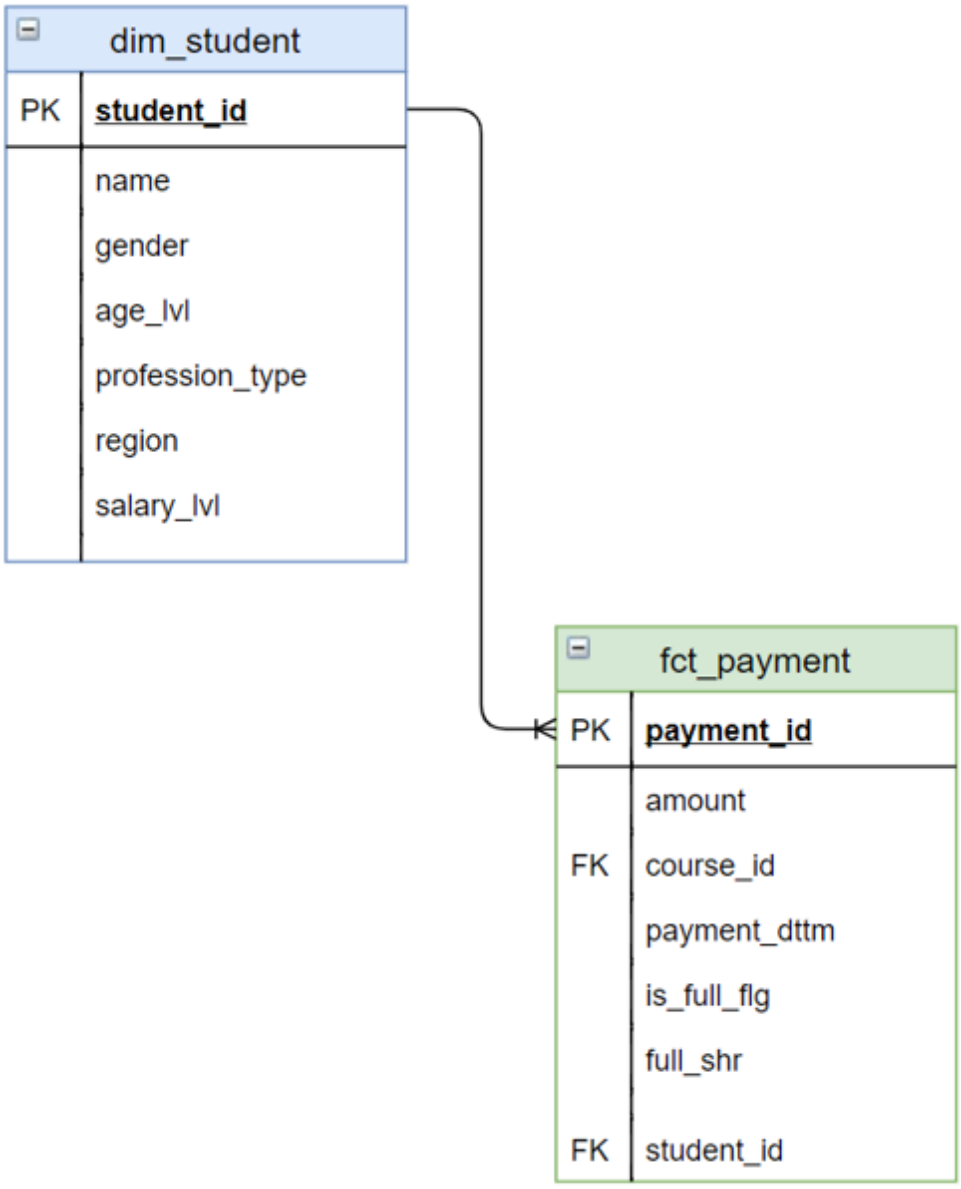
Предметной областью в практическом занятие является модель студентов и преподавателей в рамках проекта KarpovCourses. Проектировать будем по Кимбаллу, исходя из потребностей аналитиков.

Перед началом работы по созданию хранилища данных мы должна определится с тем, какие данные попадут на витрину данных для аналитиков. В модели с KarpovCourses в первую очередь аналитику будут интересны финансы, например, как студенты оплачивают курсы, сколько всего продаж и так далее. Для начала создадим таблицу фактов. Перед началом проектирования хранилища данных необходимо договориться о единообразии названий во всем хранилище. Для таблицы фактов мы будем использовать префикс `fct`.



В таблицу фактов платежей добавим поля: сумма платежа (amount), курс (course_id), дата платежа (payment_dttm), полностью или частично была совершена оплата (is_full_flg), доля платежа (full_shr), студент (student_id).

Добавим таблицу измерений dim_student. Для таблиц измерений будем использовать префикс `dim`. Первичным ключом будет являться идентификатор студента. У нас есть ФИО студента, но для анализа нам также будут полезны пол, возрастная группа, профессия, регион и класс зарплаты. Мы может не запрашивать все данные у студента, часть данных мы можем получить из сторонних источников.



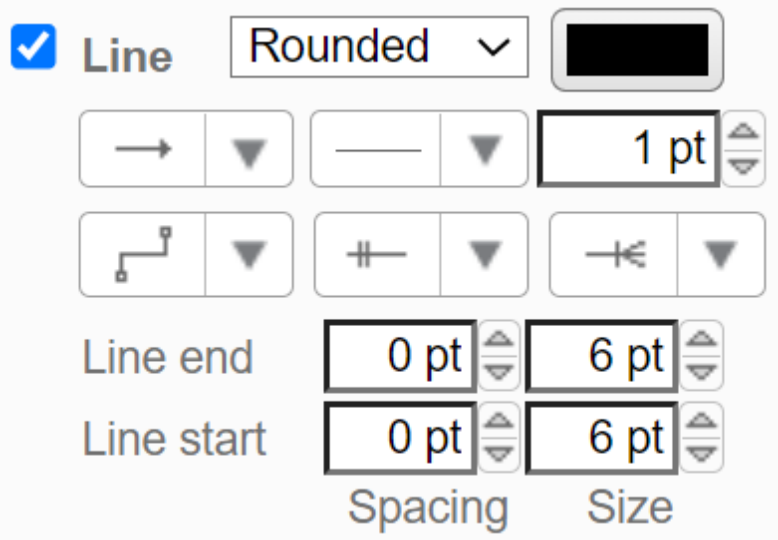
В таблицу измерений dim_student добавим поля: ФИО (name), пол (gender), возрастная группа (age_lvl), профессия студента (profession_type), регион студента (region), класс зарплаты (salary_lvl).

> Связь фактов и измерений в ER-диаграмме

Свяжем таблицу фактов и таблицы измерений. Связь у нас один ко многим. Нам необходимо правильно указать окончания связей. У нас может быть много записей в таблице фактов с минимальным количеством связей - 1, поэтому мы выбираем тип *один и лапка*, как на изображении ниже. У нас не может присутствовать запись в таблице фактов без записи в таблице измерений. Для

таблицы измерений мы выбираем тип окончания *один*, его символ представлен на изображении ниже. *(если забыли обозначения типов связей – тут)*

В меню STYLE мы можем настроить отображение и тип связи.

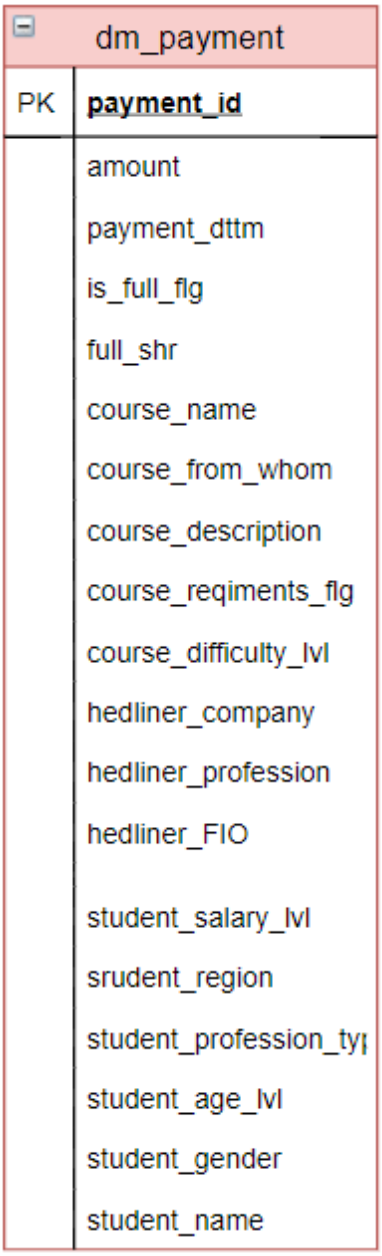


меню STYLE в draw.io

> Подготовка плоской таблицы для аналитиков

Аналитикам может потребоваться делать меньше join-операций и нам может потребоваться плоская витрина, например, с платежами. Для работы с плоской таблицей аналитикам нужно присоединить таблицы измерений к таблице фактов, в нашем случае к таблице `fct_payment` таблицы `dim_student`, `dim_course` и `dim_lector`. Для этого надо скопировать все поля из таблиц измерений в таблицу фактов. Назовем новую таблицу фактов `dm_payment` (dm – Data mart). Когда мы присоединяем таблицу по SCD-2 у нас есть два сценария. В SCD-2 у нас есть даты начала и окончания, если при join'е мы укажем дату окончания равной плюс бесконечность, то мы получим актуальное состояния, а если мы при join'е сделаем between начала и окончания, то мы получим состояния на дату платежа.

В результате мы получили плоскую денормализованную таблицу. Для аналитика такая таблица будет удобна, но с точки зрения управления такая таблица не очень удобна.



Плоская таблица данных для аналитиков

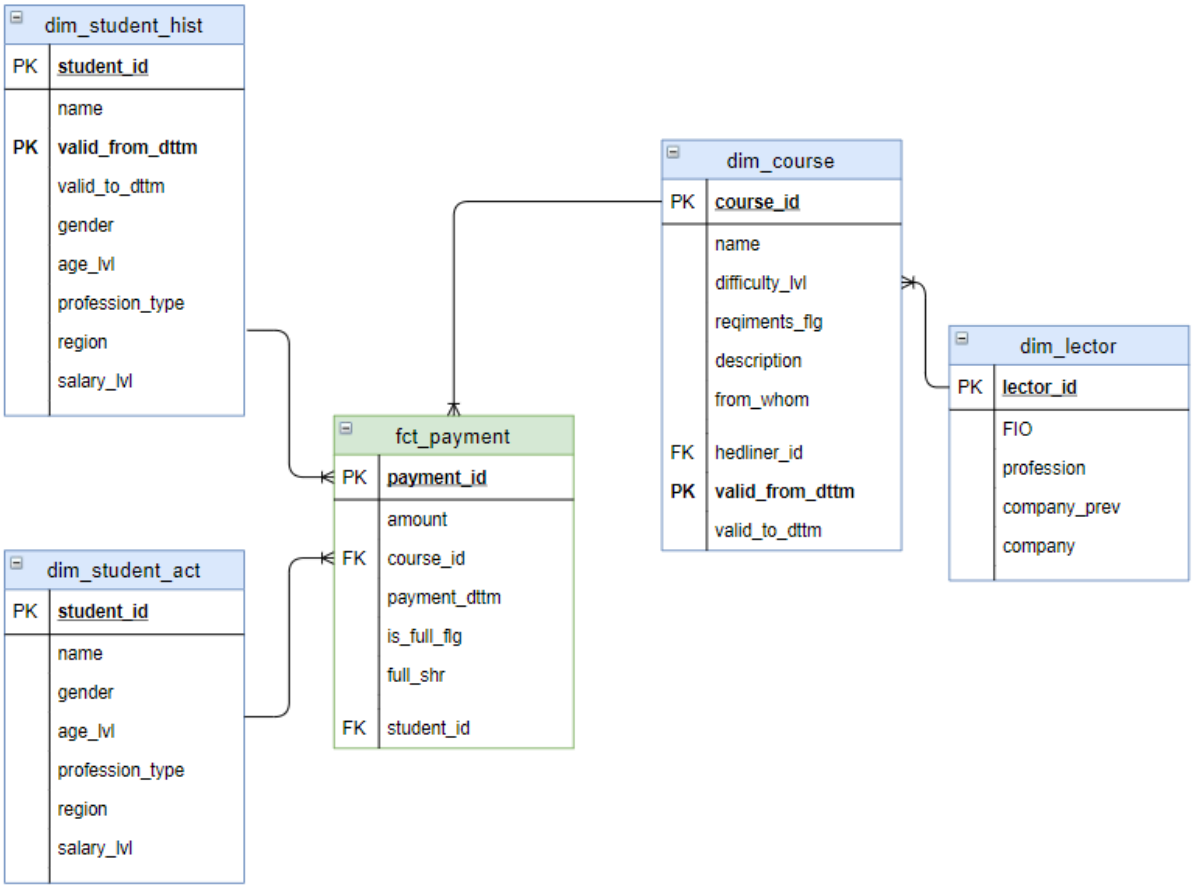
> Добавление SCD в таблицы измерений

Мы подготовили небольшую витрину данных для аналитиков. Нам осталось выбрать и добавить типы измерений (SCD) в наши таблицы измерений. Допустим, мы хотим смотреть историю изменений по курсу. Мы можем это делать по SCD-2, в таком случае нам необходимо добавить два дополнительных поля: `valid_from_dttm` и `valid_to_dttm`. После добавлениях таких

полей уникальный id курса уже не будет валидным ключом, поэтому теперь нам надо использовать пару `course_id` + `valid_from_dttm` или `course_id` + `valid_to_dttm`.

Также нам интересно посмотреть аналогичную информацию по лекторам. Допустим, что лектор редко меняет свою компанию. Будем хранить по другому типу с хранением предыдущего значения: SCD-1 (на практике такой тип используется редко). Добавим поле `company_prev`, но у нас нет даты, с которой произошло изменение, поэтому этими данными будет сложно пользоваться на практике.

Для студентов мы сделаем хранение по типу SCD-4. Получится 2 таблицы по студентам: одна в актуальном состоянии `dim_student_act`, другая с историческими значениями `dim_student_hist`. В `dim_student_hist` нужно добавить даты действия, что и в SCD-2.

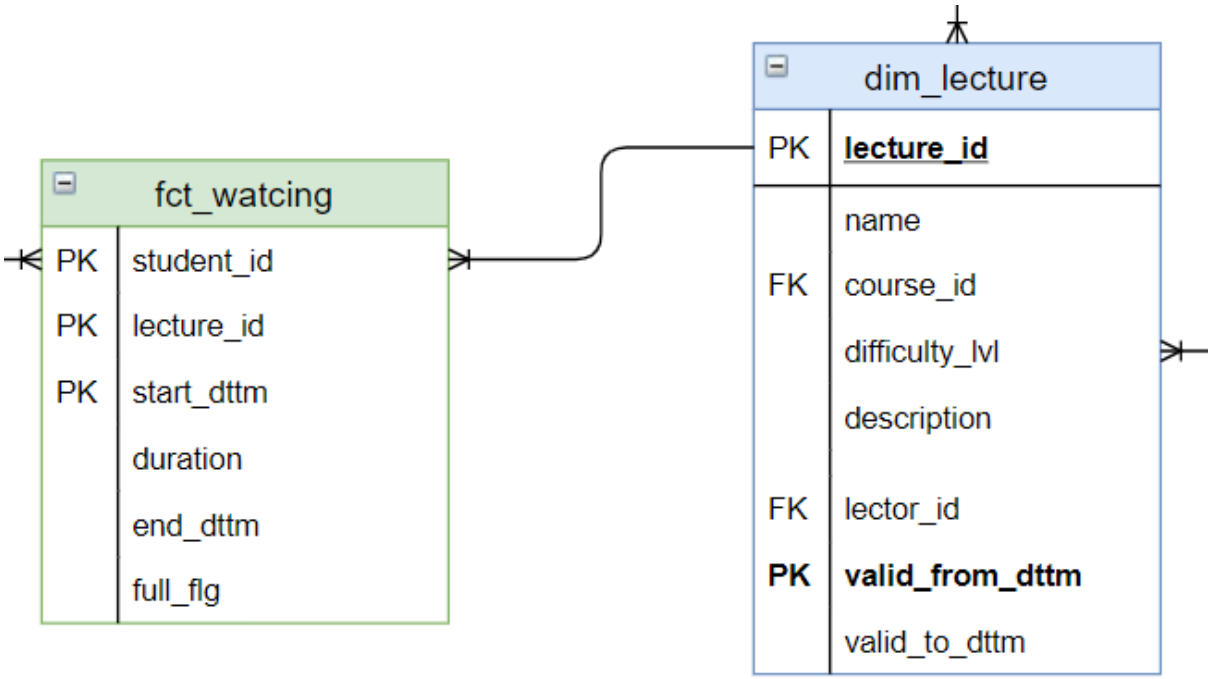


> Таблица фактов с просмотрами и общая схема

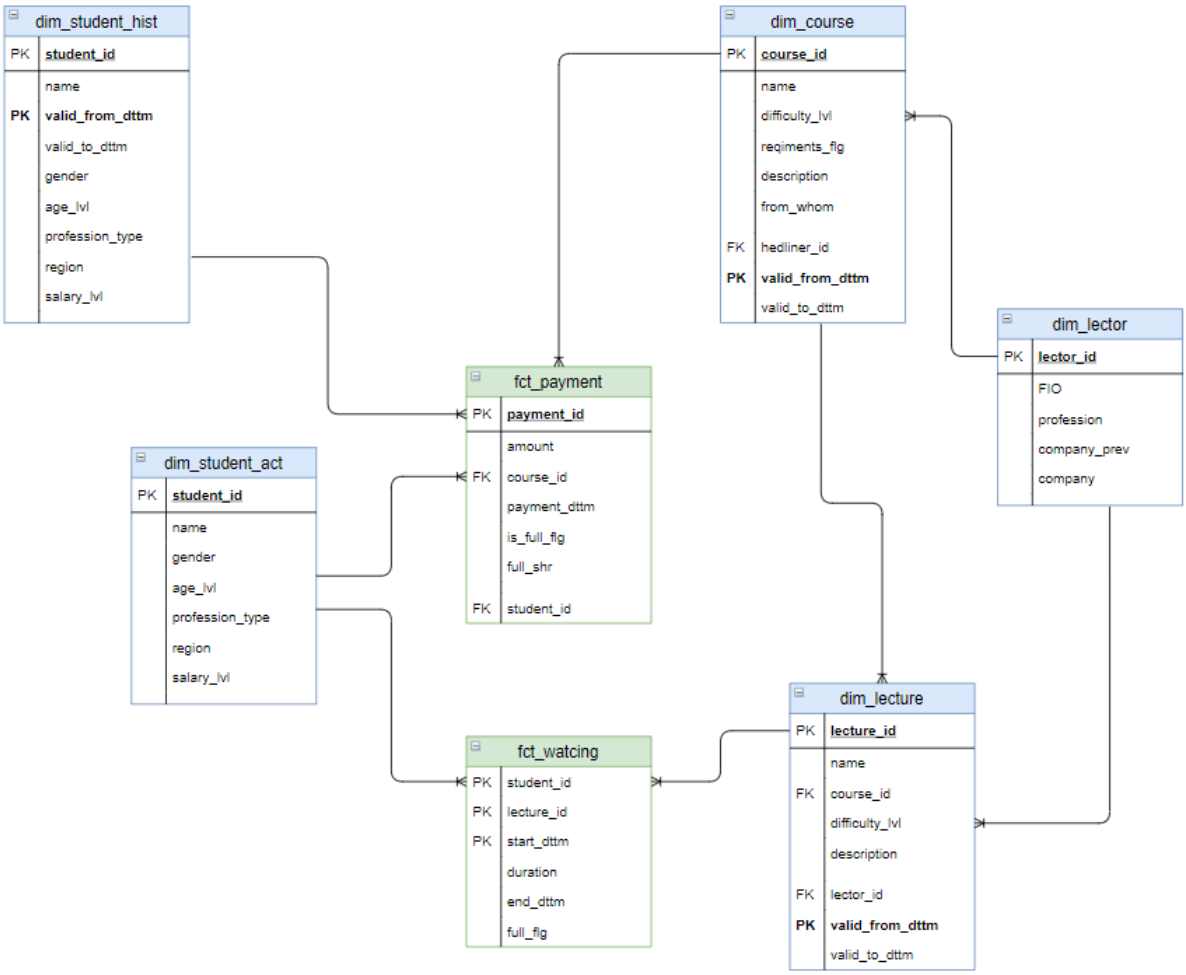
Помимо платежей аналитикам может быть интересно, насколько студенты заинтересованы материалом и досматривают лекции до конца, какие преподаватели лучше готовят материалы. Мы создаем таблицу фактов с просмотрами. В таблице фактов мы укажем поля первичного ключа `student_id`, `lecture_id`, `start_dttm`, а также продолжительность `duration`, `end_dttm`, флаг просмотра лекции до конца `full_flg`. Мы выбрали такой первичный ключ исходя из предположения, что в нашей системе студент может запустить только одно видео.

Дополнительно создаем таблицу измерений с информацией о лекциях. Первичным ключом будет являться `lecture_id`. Добавим поля с названием лекции name, сложностью `difficulty_lvl`, описанием `description`, лектором `lector_id`, а также поля для хранения по SCD-2 `valid_from_dttm` и `valid_to_dttm`.

Добавим связь между курсом и лекцией, лектором и лекцией, а также внешний ключ `course_id` в лекцию.



Получилась следующая общая схема



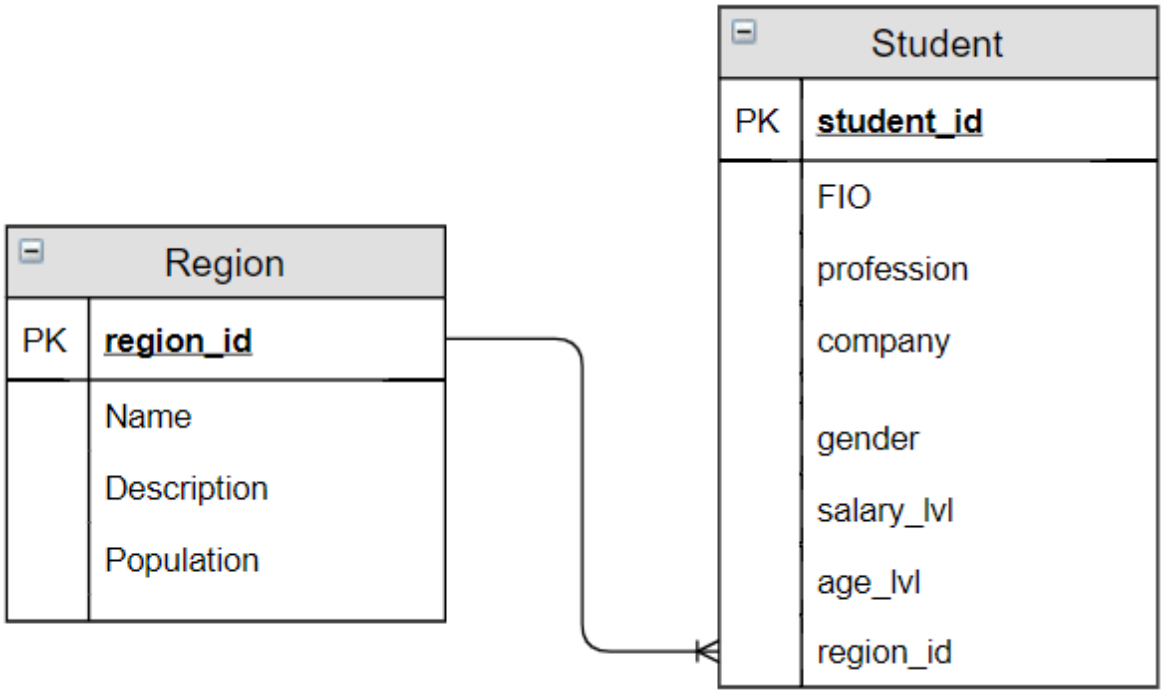
> Создание ER-диаграммы по Инмону

Выделим ключевые бизнес-сущности, их характеристики (атрибуты) и связи между ними.

У нас есть сущность Студенты, у которых есть ФИО `FIO`, профессия `profession`, компания `company`, пол `gender`, возраст `age_lvl`, регион `region_id` и уровень зарплат `salary_lvl`.

Добавим таблицу с Регионами, у которых есть наименование `Name`, описание `Description` и количество населения `Population`.

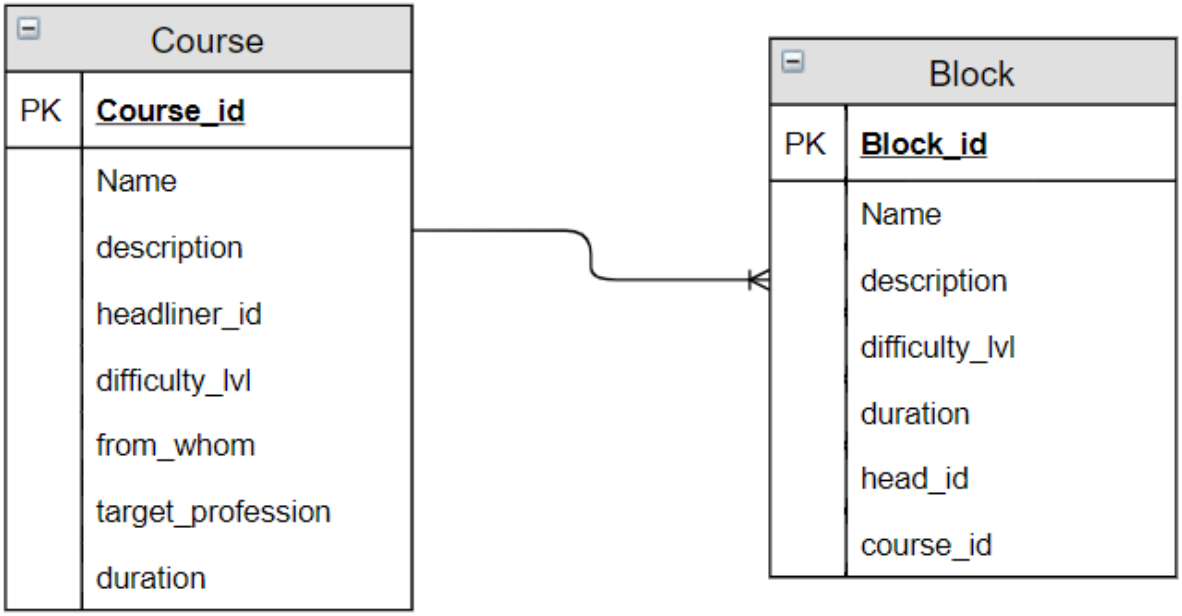
Добавим связь между этими сущностями.



Таблицы с сущностями Студенты и Регионы

Также у нас есть Курсы, с атрибутами наименование `Name`, описание `description`, уровень сложности `difficulty_lvl`, для кого предназначен курс `from_whom`, для какой профессии `target_profession`, продолжительность `duration`.

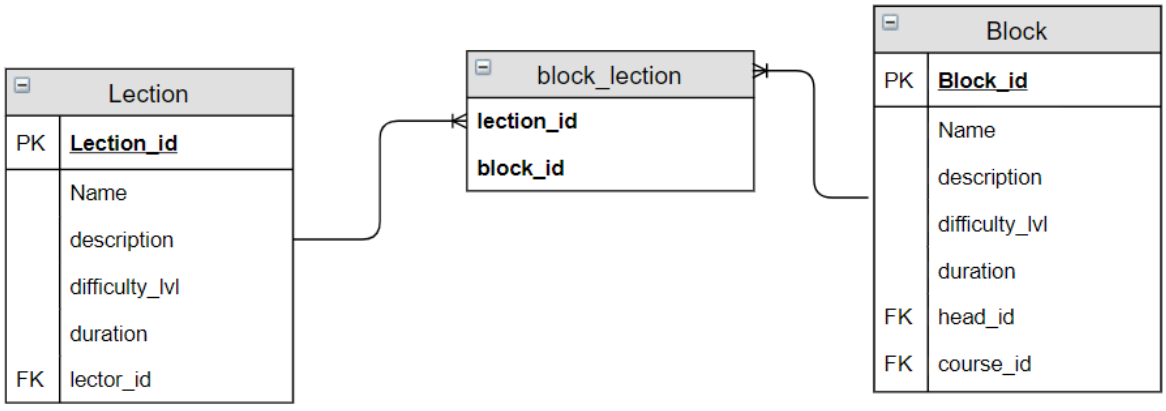
Курс состоит из Блоков, у которых есть наименование `Name`, описание `description`, уровень сложности `difficulty_lvl`, продолжительность `duration`.



Таблицы с сущностями Курсы и Блоки

Добавим сущность с Лекциями, у которых есть наименование `Name`, описание `description`, уровень сложности `difficulty_lvl`, продолжительность `duration`, принадлежность к блоку `block_id`.

Допустим, что одна лекция может быть в нескольких блоках. Создадим таблицу для соединения блоков и лекций `block_lection`. Фактически это реализация соединения многие ко многим, но через отдельную таблицу.



Таблицы с сущностями Лекции, Блоки и их связь

Добавим сведения о Лекторах: ФИО `FIO`, компания `company`, профессия `profession`, навыки `skills`, и соединим с другими сущностями, добавив в них поля внешних ключей.

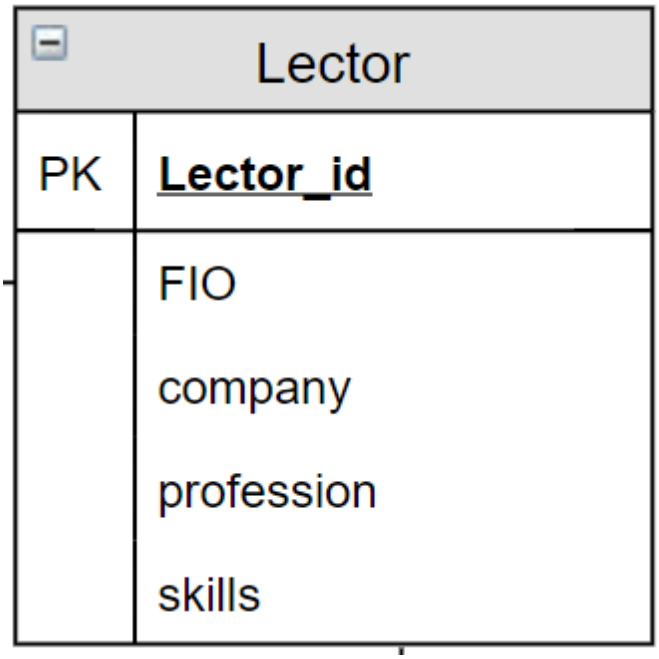


Таблица с сущностью Лектор

Добавим сущность с Просмотрами видео: студент `student_id`, лекция `lection_id`, начало просмотра `start_dttm`, продолжительность `duration`, досмотрено ли до конца `full_flg`.

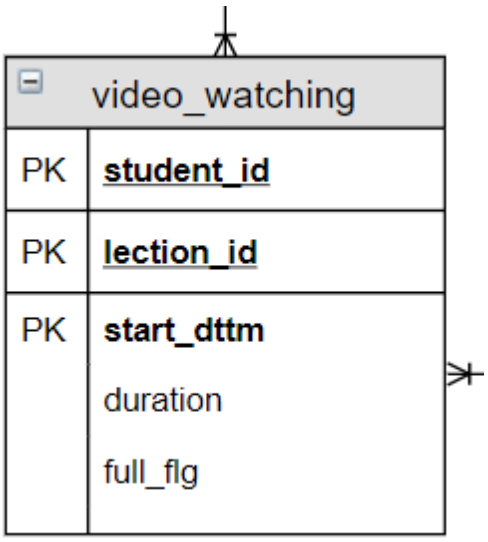
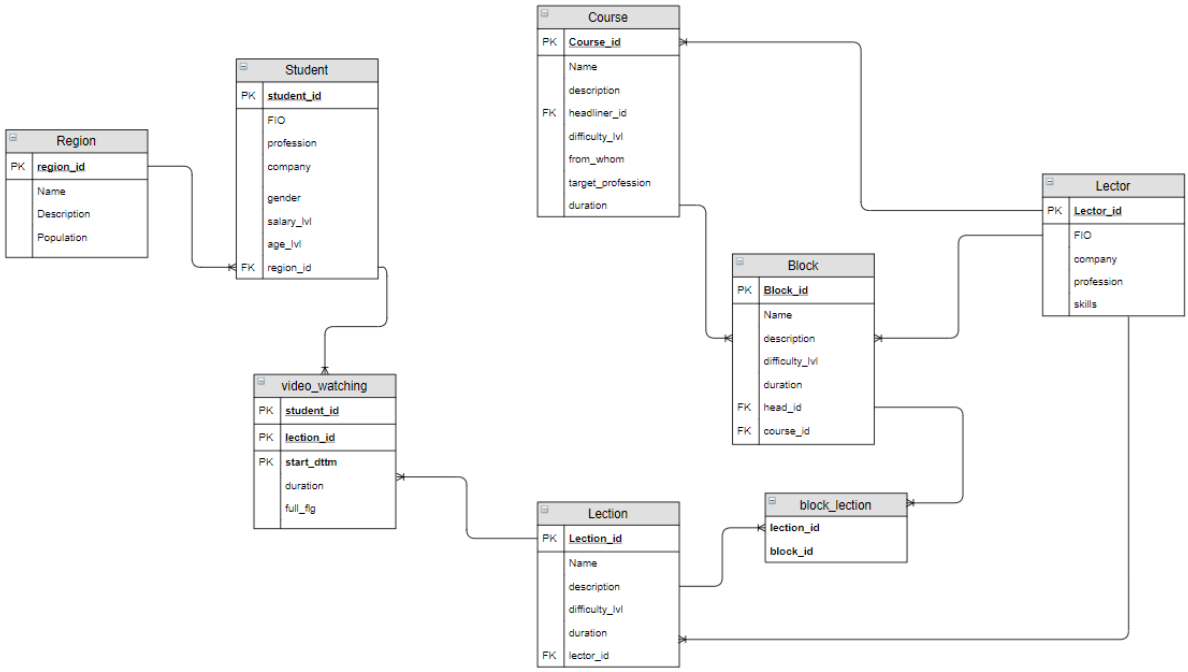


Таблица с сущностью Просмотра видео

Можно добавить ещё много таблиц, описывающих те или иные сущности, а также историзмы по SCD.



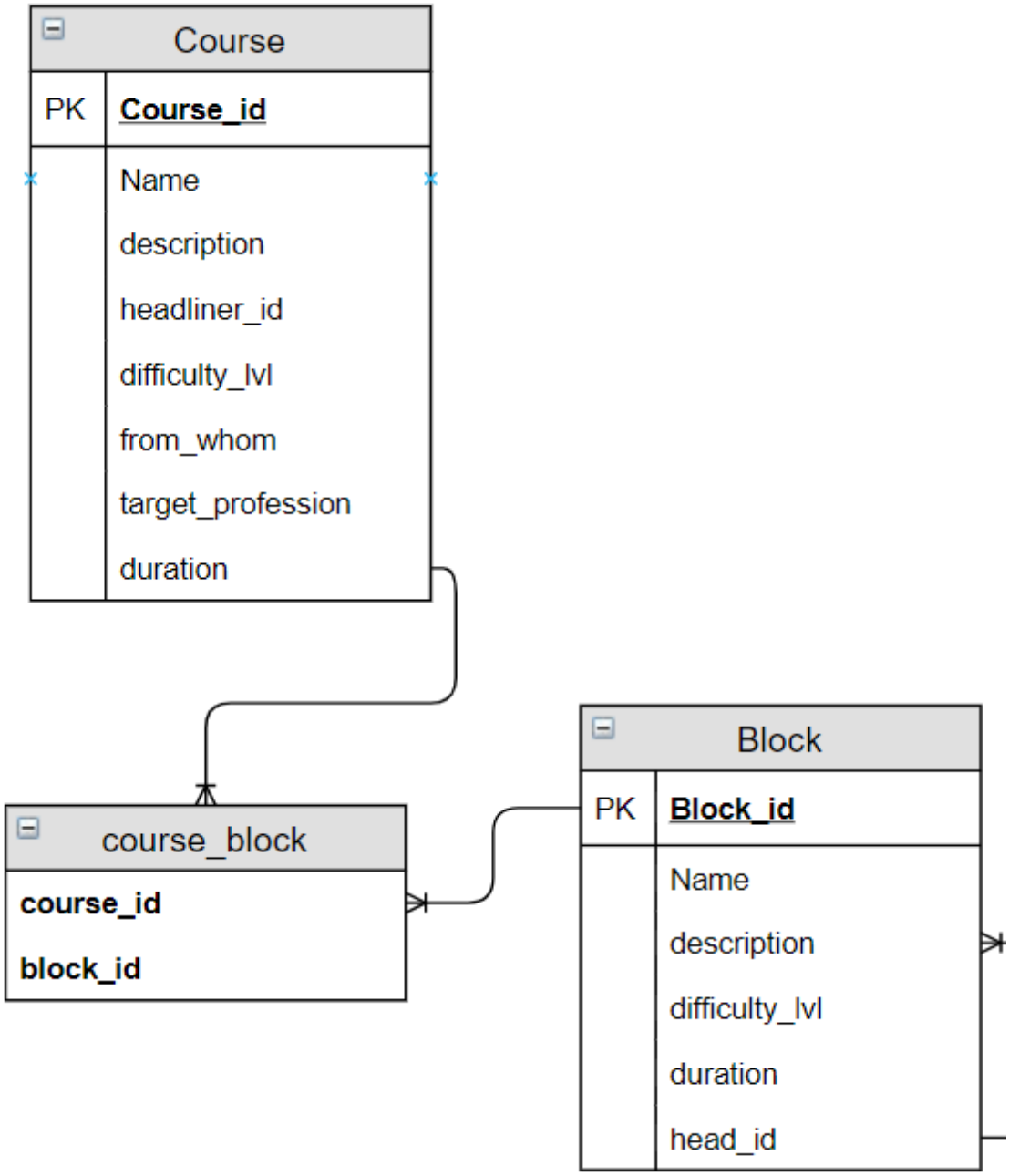
Общая схема

Данная схема не отрицает таблицу фактов и измерений, это наш центральный слой, на базе него мы можем стоить таблицы фактов и измерений.

> Неустойчивость ER-диаграммы по Инмону

Такая схема не является устойчивой. Допустим, что между курсом и блоком поменялась связь: теперь блок может принадлежать нескольким курсам.

Убираем поле `course_id` из блока и добавляем таблицу связи между курсом и блоком.



Изменение связи между Курсом и Блоком

Если мы захотим добавить регион к лектору, то это тоже повлечёт за собой изменение полей таблицы, и необходимо будет прогрузить данные по регионам, таким образом на время изменений таблица будет заблокирована.

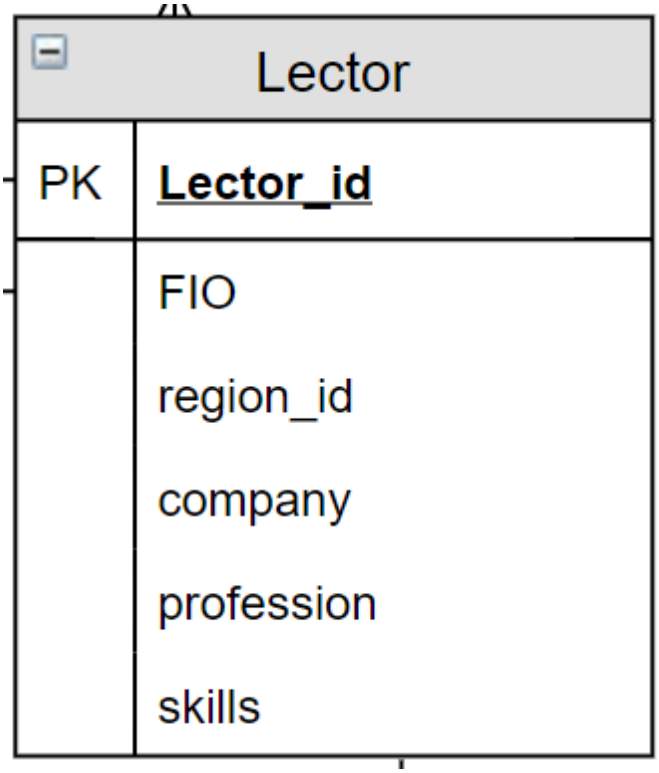


Таблица с сущностью Лектор с добавленным полем region_id