**Пользователи**:

**Сущности**: ID, Имя, Email, Может\_быть\_автором

Хранит информацию о пользователях. Указывает, может ли пользователь быть автором.

**Разделы**

**Сущности**: ID, Название, Родительский\_раздел\_ID

Организует публикации по разделам и подразделам.

**Публикации**

**Сущности**: ID, Заголовок, Анонс, Описание, Дата\_публикации, Автор\_ID

Содержит данные о статьях. Связана с автором через Автор\_ID.

**Публикации\_Разделы**

**Сущности**: Публикация\_ID, Раздел\_ID

Реализует связь многие-ко-многим между публикациями и разделами.

**Теги**

**Сущности**: ID, Название

Хранит теги для публикаций.

**Публикации\_Теги**

**Сущности**: Публикация\_ID, Тег\_ID

Связывает публикации и теги (многие-ко-многим).

**Комментарии**

**Сущности**: ID, Публикация\_ID, Пользователь\_ID, Текст, Дата

Хранит комментарии пользователей к публикациям.

**Оценки**

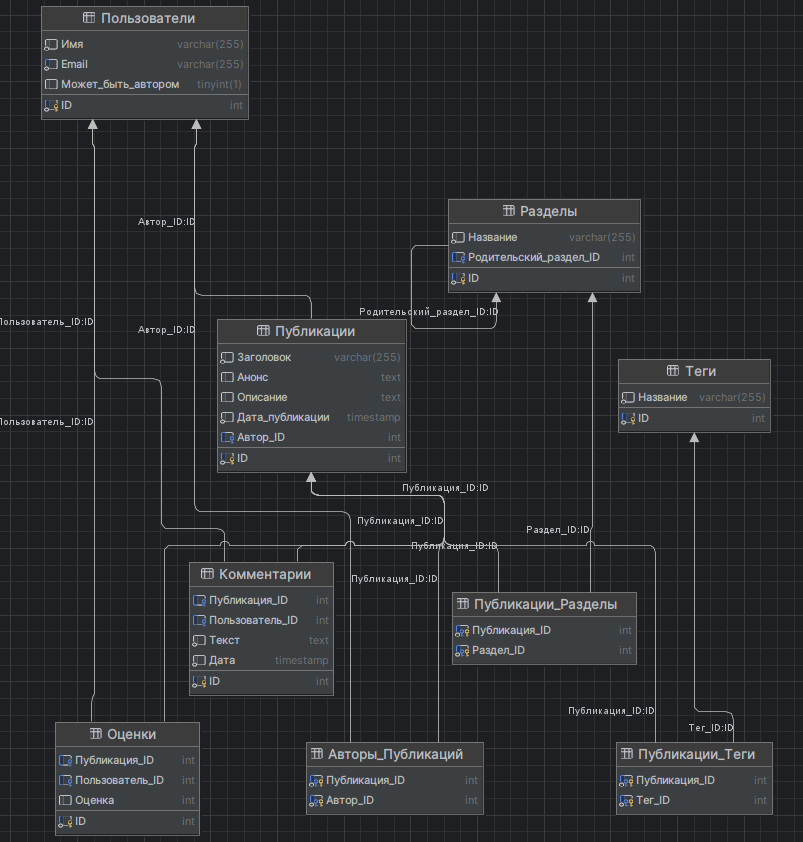
**Сущности**: ID, Публикация\_ID, Пользователь\_ID, Оценка

Содержит оценки публикаций от пользователей.

**Авторы\_Публикаций**

**Сущности**: Публикация\_ID, Автор\_ID

Реализует связь многие-ко-многим между авторами и публикациями.



написатьSQL-запроснаполучениевсехоценокикомментариевк публикациям заданного автора:

SELECT

Оценки.Оценка,

Комментарии.Текст AS Комментарий,

Комментарии.Дата AS Дата\_комментария,

Публикации.Заголовок AS Заголовок\_публикации

FROM

Публикации

JOIN

Авторы\_Публикаций ON Публикации.ID = Авторы\_Публикаций.Публикация\_ID

JOIN

Пользователи AS Авторы ON Авторы\_Публикаций.Автор\_ID = Авторы.ID

LEFT JOIN

Оценки ON Публикации.ID = Оценки.Публикация\_ID

LEFT JOIN

Комментарии ON Публикации.ID = Комментарии.Публикация\_ID

WHERE

Авторы.ID = :author\_id; -- :author\_id заменить на ID автора