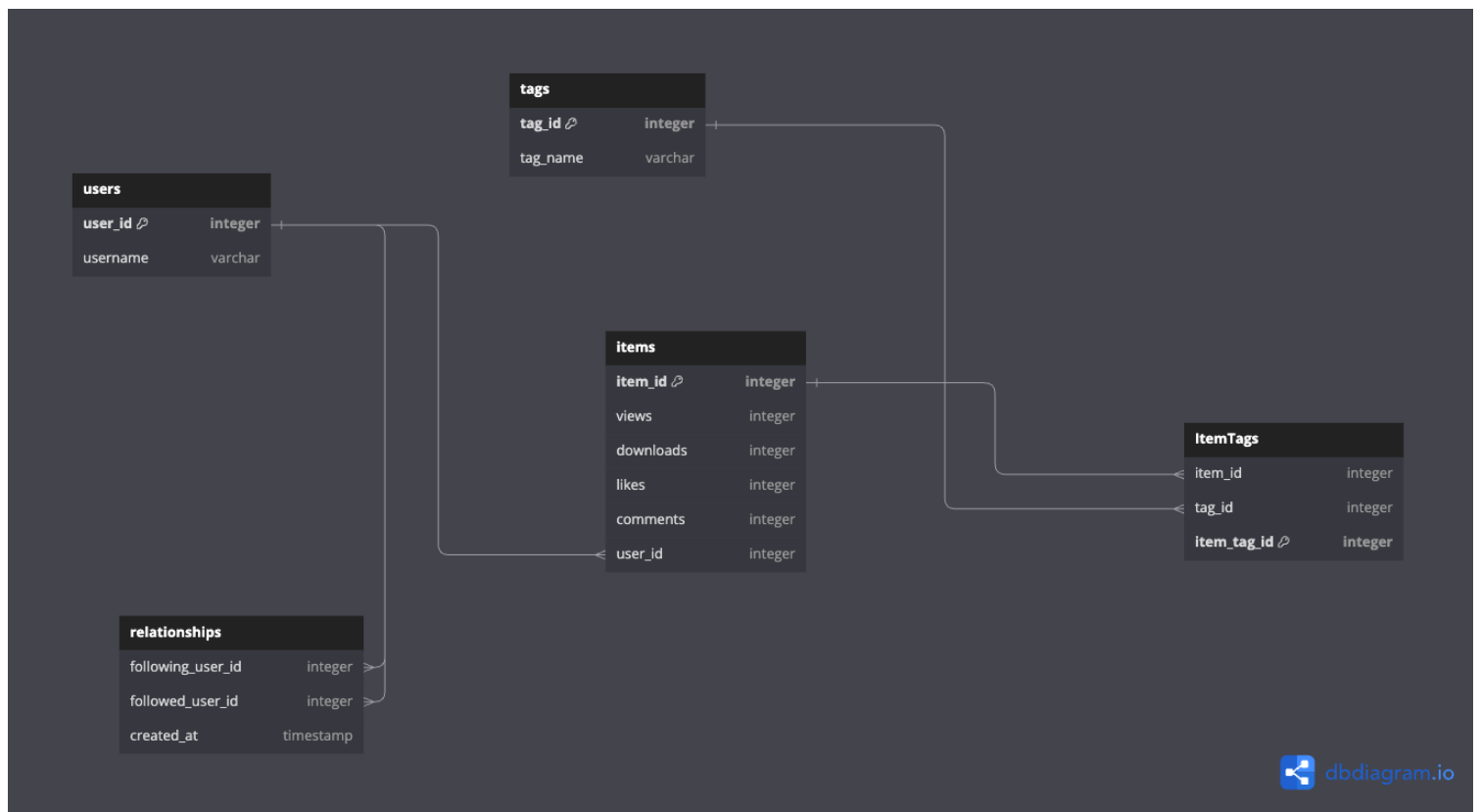


- Скласти модель даних для подальшого збереження в головному Data Warehouse.
- - Описати особливості джерела даних в контексті вивантаження великих об'ємів даних.



1. Взаємозв'язки таблиць:

- Таблиця users пов'язана з таблицею items через user_id, що вказує на те, що кожен об'єкт асоційований з користувачем. Вивантаження даних тут може включати великі об'єми контенту, створеного користувачами.
- Таблиця items пов'язана з таблицею tags через таблицю сполучень ItemTags, що вказує на відносини багато-до-багатьох. Це може ускладнити вивантаження даних, оскільки кожен об'єкт може мати багато тегів, і один тег може бути пов'язаний з багатьма об'єктами.

2. Індиксація та оптимізація:

- Великі об'єми даних часто вимагають належної індиксації для підвищення швидкості запитів, особливо для оперативного вивантаження даних. Індиксація зовнішніх ключів, таких як user_id, item_id, і tag_id, може значно поліпшити продуктивність.
- Для оптимізації запитів можуть бути використані складні запити SQL з JOIN-операціями, особливо при роботі з великими наборами даних з багатьох таблиць.

3. Проблеми з проєктуванням:

- Якщо items має високий обсяг views, downloads, likes, і comments, робота з цією таблицею може бути важкою при високому обсязі трафіку, особливо при наявності реальних оновлень даних.
- Таблиця relationships містить час створення (created_at), який може бути важливим для вивчення тенденцій або динаміки взаємин між користувачами. Обробка часових міток у великих наборах даних може вимагати спеціальних підходів, таких як часові вікна або індекси на основі часу.

4. Розширення масштабу:

- Вивантаження та обробка великих об'ємів даних можуть вимагати розширеного програмного забезпечення для роботи з базами даних, такого як розподілені бази даних або хмарні рішення, щоб управляти великими об'ємами та високим навантаженням.

5. Запити та транзакції:

- При вивантаженні великих даних важливо врахувати транзакційність системи, особливо коли дані часто

оновлюються. Використання транзакцій допоможе забезпечити консистентність даних і зменшити можливість даних, які можуть бути пошкоджені або втрачені під час процесів оновлення.