**РОЗРОБКА EAV БАЗИ ДАНИХ ДЛЯ ЗБЕРІГАННЯ ІНФОРМАЦІЇ**

**DEVELOPMENT OF EAV DATABASE FOR STORAGE INFORMATION**

Науковий керівник: к.т.н., доцент кафедра інформаційних технологій

Вороной Сергій Михайлович

Здобувач бакалаврату Батура Максим Юрійович

Supervisor: (Ph.D), assistant professor Department of Information Technology

Voronoi Sergiy Mikhailovich

Bachelor's candidate Batura Maxim Yuriyovich

***Анотація:*** *розглянуто проблеми, особливості призначення та функціональність EAV типу бази даних. Їх структуру, та основні поля таблиць.*

***Ключові слова****: бази даних.*

***Abstract****: problems, features of purpose and functionality of EAV type of database are considered. Their structure, and the main soldering tables.*

***Keywords****: databases.*

Модель сутності – атрибута – значення (EAV) – це модель даних для ефективного використання сутностей, відносно скромна. Такі сутності відповідають математичному поняттю розрідженої матриці [1].

Термін «база даних EAV» відноситься до структури бази даних, де значна частина даних моделюється як EAV. Проте навіть у базі даних, описаної як «заснована на EAV», деякі таблиці у системі традиційними реляційними таблицями [2].

У прості конструкції EAV значення атрибута є простими або примітивними типами даних, до бази даних двигун вирівняний [3]. Системи EAV, що використовуються для представлення, можуть мати об'єкт (примірник) можуть мати підструктуру: деякі з його атрибутів можуть мати інші види об'єктів класу, які, у свою чергу, можуть мати підструктуру, щоб довільний рівень складності. У автомобіля, наприклад, є двигун, трансмісія і т. д., а у двигуна є такі компоненти, як циліндри [4]. Допустима підструктура для даного класу визначена в метаданих атрибутів, як обговорюється пізніше. Таким чином, наприклад, атрибут «пам'ять з довільним доступом» може використовувати клас «комп'ютер», але не до класу «двигун» [5].

Для представлення підструктури включають спеціальну таблицю EAV, у якій стовпець значень містить посилання інші сутності у системі (т. е. значення зовнішнього ключа у таблиці об'єктів) [6]. Щоб отримати всю інформацію про цей об'єкт, потрібно рекурсивний обхід метаданих, за яким слідує рекурсивний обхід даних, який зупиняється, коли кожен отриманий атракціон є основним (основним) [7]. Рекурсивний обхід необхідний незалежно від того, які деталі окремого класу у традиційній формі або у формі EAV; такий обхід виконується, наприклад, у стандартних об'єктно-реляційних системах [8]. На практиці кількість рівнів рекурсії, як правило, відносно невелика для міжнародних класів, тому втрати продуктивності через рекурсію скромні, особливо при індексуванні використання об'єктів [9].

EAV/CR (EAV з класами та відносинами) відноситься до структури, яка підтримує складну підструктуру [10]. Система EAV, яка використовується в стандартній реляційній системі, залежить від того, чи є атрибутиженними або щільними (рис.1). . EAV / CR дійсно дуже докладними метаданими, які досить багаті, щоб підтримувати автоматичне створення інтерфейсу для окремих класів без необхідності писати код інтерфейсу користувача для кожного класу.



Рисунок 1 – Структура бази даних

В тому, що можна згенерувати пакет динамічних SQL-запитів, який не залежить від класу вихідних даних, заснованих на його метаданих та використовуючи інформацію для генерації запитів до таблиць, і деякі з цих запитів можуть бути рекурсивними. Цей підхід добре працює для запитів «об'єкт за раз», і в веб-інтерфейсі перегляду, де при цьому на ім'я об'єкта всі деталі об'єкта надані окремі: метадані, пов'язані з класом об'єкта, також представлені уявлення деталей об'єкта, тому що воно містить заголовки окремих атрибутів, порядок, у яких вони мають бути представлені, і навіть спосіб їх угруповання.

У таблиці OBJECTS зберігається основна інформація про кожен з об’єктів, а саме:

* ідентифікаційний номер об’єкту — обов’язкове поле;
* ідентифікаційний номер батьківського об’єкту — не обов’язкове поле;
* ідентифікаційний номер об’єктоного типу — обов’язкове поле;
* ім’я об’єкту — не обов’язкове поле;
* опис об’єкту — не обов’язкове поле.

Таблиця OBJTYPE зберігає наступну інформацію:

* ідентифікаційний номер об’єктоного типу — обов’язкове поле;
* код об’єктного типу — обов’язкове поле;
* ім’я об’єктного типу — не обов’язкове поле;
* опис об’єктного типу — не обов’язкове поле.

Наступна таблиця OBJREFERENCE:

* ідентифікаційний номер атрибуту — обов’язкове поле;
* ідентифікаційний номер об’єкту на котрий посилаються — обов’язкове поле;
* ідентифікаційний номер об’єкту який посилається — обов’язкове поле.

У таблиці ATTRTYPE:

* ідентифікаційний номер атрибуту — обов’язкове поле;
* ідентифікаційний номер об’єкту до якого відноситься атрибут — обов’язкове поле;
* ідентифікаційний номер атрибуту для побудови посилання — не обов’язкове поле;
* код атрибуту — не обов’язкове поле;
* ім’я атрибуту — не обов’язкове поле.

Таблиця ATTRIBUTES:

* ідентифікаційний номер атрибуту — обов’язкове поле;
* ідентифікаційний номер об’єкту — обов’язкове поле;
* значення атрибуту — не обов’язкове поле;
* значення дати атрибуту — не обов’язкове поле;
* ідентифікаційний номер листового значення — не обов’язкове поле.

Наступна таблиця Lists:

* ідентифікаційний номер листового значення — обов’язкове поле;
* значення листового запису — обов’язкове поле.

література

1. Балик Н. Р., Мандзюк В. І. Бази даних MySQL: Навчальний посібник.  — Тернопіль: Навчальна книга – Богдан, 2010.— 160  с
2. Авторський колектив: канд. техн. наук, доцент М. Ю. Лосєв – розділи 2, 4; канд. фіз.-мат. наук, доцент В. В. Федько – розділи 1, 3.
3. Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Keir Thomas, Matt Wade - Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013.
4. Дейт, К. Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильяме", 2005. — 1328 с.: ил. — Парал. тит. англ
5. Реляционные базы данных в примерах, EPAM Systems, RD Dep, 2021
6. Эффективные методы работы с вертикальной моделью данных / Д. В. Гмарь, Ю. А. Игнатова, Э. В. Цуранов, К. И. Шахгельдян // Информационные технологии и вычислительные системы 2/2015
7. ENTITY-ATTRIBUTE-VALUE (EAV) DESIGN IN POSTGRESQL – DON’T DO IT! [Електронний ресурс] URL: <https://www.cybertec-postgresql.com/en/entity-attribute-value-eav-design-in-postgresql-dont-do-it/> (дата звернення 15.05.2022)
8. Entity Attribute Value in Magento [Електронний ресурс] URL: <https://blog.magestore.com/entity-attribute-value-in-magento/> (дата звернення 15.05.2022)
9. EAV Database: General Architecture Overview - How It Works [Електронний ресурс] URL: <https://www.gomage.com/nl/blog/eav-database/> (дата звернення 15.05.2022)
10. Проектування ORM системи для EAV бази даних [Електронний ресурс] URL:[http://apeps.kpi.ua/downloads/Хорошун%20-%20Проектування%20ORM%20системи%20для%20EAV%20бази%20даних.pdf](http://apeps.kpi.ua/downloads/%D0%A5%D0%BE%D1%80%D0%BE%D1%88%D1%83%D0%BD%20-%20%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20ORM%20%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B8%20%D0%B4%D0%BB%D1%8F%20EAV%20%D0%B1%D0%B0%D0%B7%D0%B8%20%D0%B4%D0%B0%D0%BD%D0%B8%D1%85.pdf) (дата звернення 15.05.2022)