

資料庫管理期末專案 - 筆知

Group 24 嚴邦華、余沛殷

系統簡介與分析

- ▶ 你是否也曾在創作文章時覺得缺少共鳴？「筆知」是一個為文章創作者提供交流和互相學習的線上平台，讓擁有共同興趣和經歷的人們能夠在這裡分享想法、撰寫文章、閱讀別人的作品，並進行互動留言，打造充滿活力的文章社群。不論你是希望尋找志同道合的朋友，還是需要不同觀點來激發創意，「筆知」都能幫助你找到最適合的夥伴！

系統簡介與分析

使用者可以：

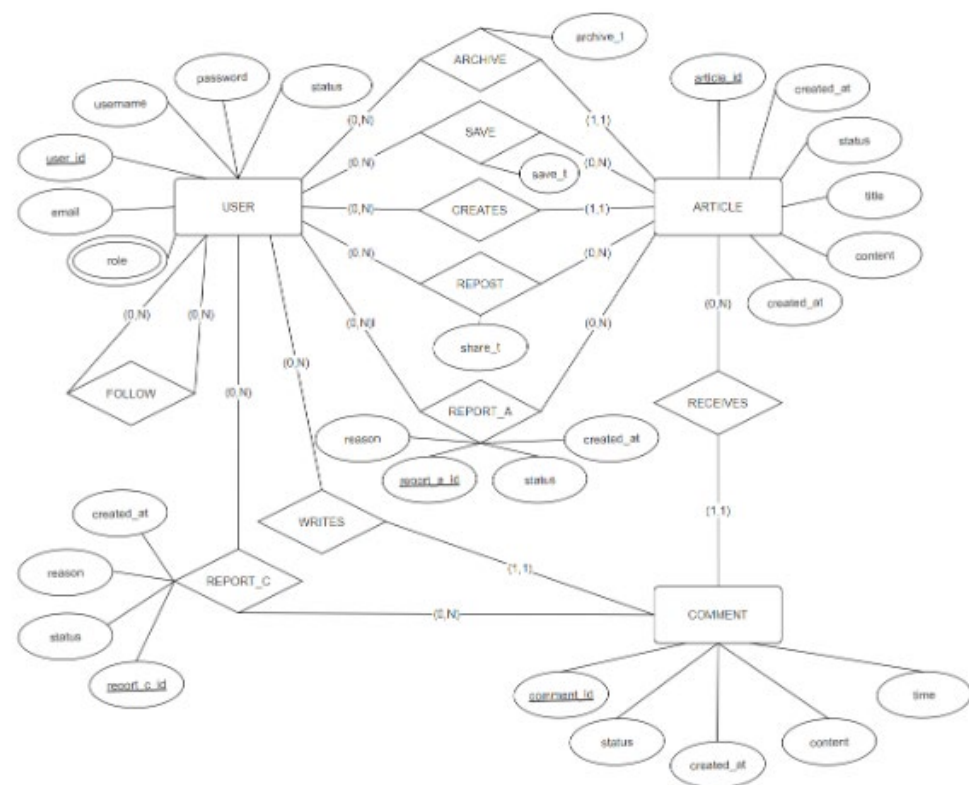
- ▶ 瀏覽文章與留言
- ▶ 新增文章
- ▶ 儲存文章
- ▶ 分享文章
- ▶ 查詢文章
- ▶ 封存文章
- ▶ 瀏覽個人頁面
- ▶ 新增留言
- ▶ 追蹤他人
- ▶ 舉報他人
- ▶ 更改密碼與電子郵件

管理者可以：

- ▶ 管理文章
- ▶ 管理留言
- ▶ 管理帳號
- ▶ 查詢舉報
- ▶ 管理舉報

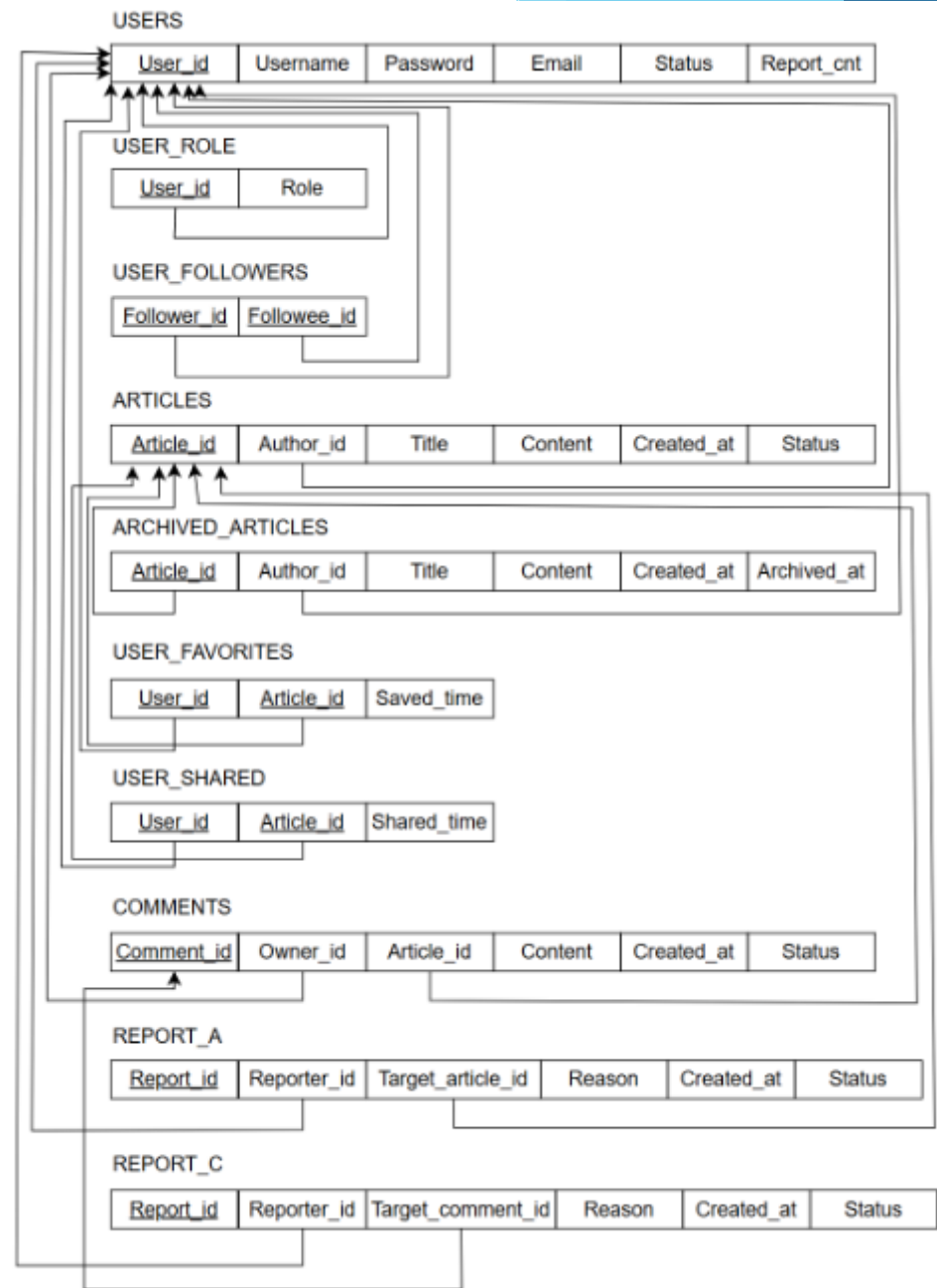
系統簡介與分析

- 右圖是「筆知」的 ERD。在這個 ERD 中，共有三個主要實體，分別是 USER、ARTICLE 和 COMMENT，以及多個關係，包括 ARCHIVE、SAVE、CREATES、REPOST、REPORT_A、REPORT_C、FOLLOW 和 RECEIVES，共同描繪了「筆知」平台內各類功能與交互行為。



系統簡介與分析

- 我們可以將前頁的 ER Diagram 轉換成右圖的 Database Schema Diagram，一共有多個表，分別是 USERS、USER_ROLE、包括 USER_FOLLOWS、ARTICLES、ARCHIVED_ARTICLES、USER_FAVORITES、USER_SHARED、COMMENTS、REPORT_A和REPORT_C。



系統簡介與分析

- ▶ 在 1NF 方面，我們將 multi-valued 屬性 Role 從原關聯中獨立出來，以獨立的關聯描述，我們的 schema 因此滿足 1NF。
- ▶ 在 2NF 方面，如果關聯中的所有非鍵屬性 (non-prime attribute) 都完全功能相依 (fully functional dependency) 於任一候選鍵 (candidate key)，也就是沒有出現部分功能相依性 (partial functional dependency)，則滿足 2NF，而我們的 schema 也符合 2NF。
- ▶ 在 3NF 方面，如果一個關聯中的非鍵屬性都沒有遞移相依 (transitively dependency) 於主鍵，則滿足 3NF。檢視設計的關聯，沒有存在任何遞移相依，因此符合 3NF。在比 3NF 更嚴格的 BCNF 方面，要求關聯中的每一個功能相依的箭頭左方都要是超級鍵 (super key)，也就是要確保 $X \rightarrow Y$ 的 X 一定是超級鍵，而我們的 schema 也符合 BCNF。
- ▶ 最後是 4NF，由於「筆知」的所有關聯都不存在多值相依 (multi-valued dependency)，因此滿足 4NF。

系統展示

指令效能優化與索引建立分析

- ▶ 我們認為「筆知」系統中最常使用到的功能為「顯示於頁面的文章」，因此我們決定針對該指令進行效能優化。文章是否可顯示於主頁面需要判斷「文章是否被作者封存」。由於文章狀態 **Status** 原本是雜亂地分佈於資料庫之中，理論上透過建立索引，我們應該能更有效率地找到活躍中的文章。為了能有效優化效能，我們為該資料表中的「文章狀態 **Status**」欄位建立索引，語法如下。
- ▶ 從平均0.2356 秒變快至 0.1768 秒。

```
|| CREATE INDEX idx_article_status  
|| ON ARTICLES(status)
```


指令效能優化與索引建立分析

- ▶ 由於「筆知」系統中 **ARTICLES** 資料表中的資料量大，但也因為平台提供搜尋文章之功能，因此使用量很大，而在搜尋中除了搜尋標題外也需確認該文章的狀態 **Status** 是否為「活躍中」，因此在實現功能時將兩者逐一確認會較為費時，因此透過建立索引有望優化資料庫搜尋效率。為了能有效優化效能，我們為該資料表中的「文章標題 **Title**」及「文章狀態 **Status**」欄位建立索引，語法如下。
- ▶ 從平均0.1840 秒變快至 0.1456 秒。

```
|| CREATE INDEX idx_article_title_status  
|| ON ARTICLES(Title, Status)
```

交易管理及併行控制

- ▶ 在給 **User** 功能中新增文章與新增留言的部分，在這個過程中如果出現違反資料庫限制的情況，例如標題與內容不可為空值等，新增的動作會立即停止。此時，系統將使用 `db.rollback()` 方法撤回該次交易（其中 `db` 為 `psycopg2` 中的 `connection` 類別，作為資料庫的連線），取消之前的所有資料庫異動。系統將在最後使用 `db.commit()` 提交交易，確保文章與留言成功儲存至資料庫。
- ▶ 在我們的資料庫設計中，並未需要進行併行控制的部分。

未來展望

在這次的資料庫管理期末專題中，我們著眼於打造一個基礎的系統，未來想要將以此為基礎持續進行擴展與優化。我們希望能夠：

- ▶ 新增更多功能：未來可以引入進階的功能模組，例如自動化數據分析、使用者行為分析，以及推薦系統等。這些功能將提升系統的實用性與價值，滿足更多使用場景的需求。
- ▶ 擴充資料庫結構：隨著功能增加，資料庫的設計將進一步模組化與規範化，以便支持更大的數據規模及多樣化的數據類型。我們將優化資料表之間的關聯性，並在需要時引入分區表或分散式資料庫技術以應對高效能需求。
- ▶ 安全性與資料隱私：強化加強系統的身份驗證與授權管理，並導入更多加密技術，確保敏感資料和連線能得到妥善保護。