

Міністерство освіти і науки України  
Національний технічний університет України «КПІ ім. Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки  
інформації та управління

OLAP та сховища даних

## **ЗВІТ**

з лабораторної роботи № 1

**Виконав**  
**студент**

*Групи ПІ-71 Амброс*  
*Всеволод Володимирович*

---

(№ групи, прізвище, ім'я, по батькові )

**Прийняв**

*Ст. викладач Олійник Юрій*  
*Олександрович*

---

(посада, прізвище, ім'я, по батькові )

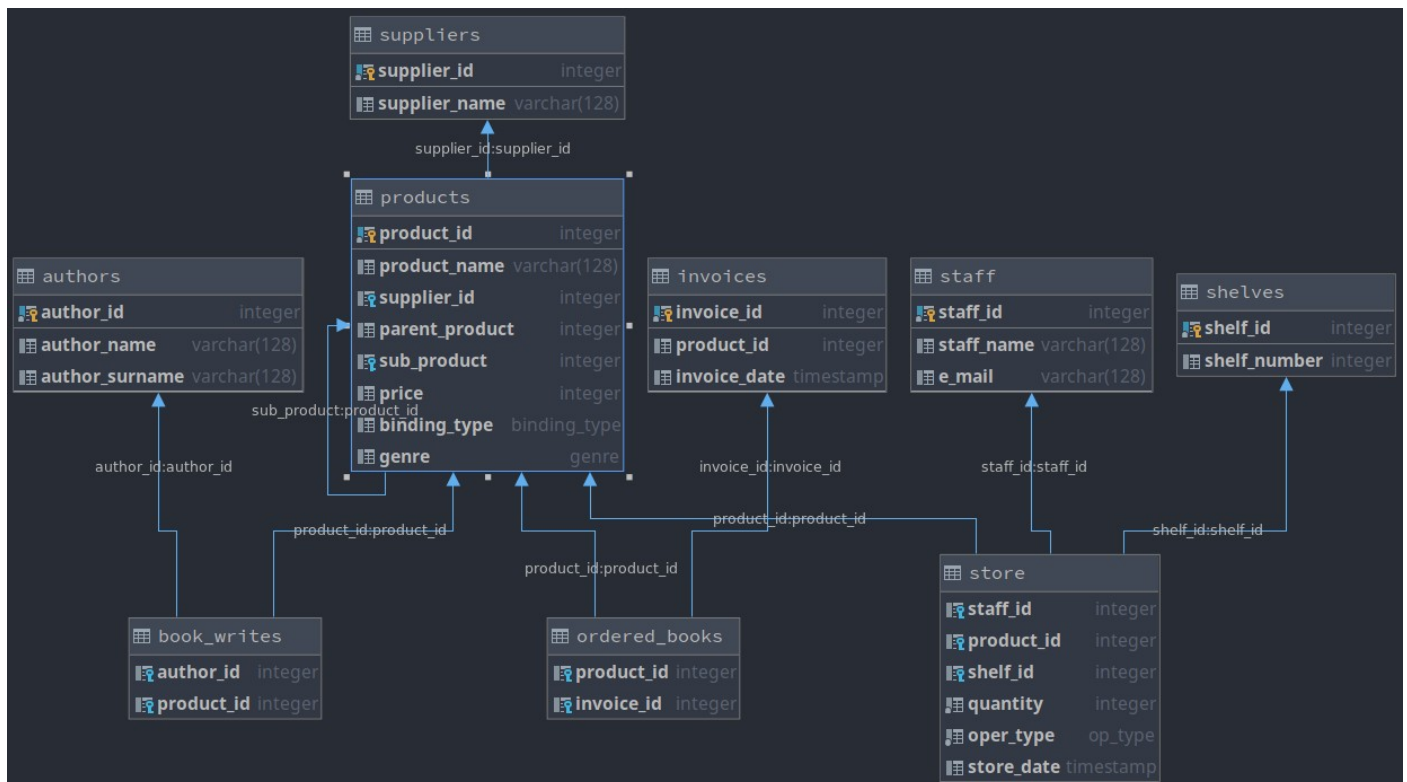
Київ 2019

## 1.Завдання

- 1) Спроекувати схему БД у 3НФ для супермаркету будівельних матеріалів на основі існуючої моделі. Для категорій товарів використайте ієрархічну структуру (для можливості збереження структури Категорія\Підкатегорія рівня1\ Підкатегорія рівня2\... без обмежень по рівням).
- 2) За допомогою SQL-запитів вивести дані, що ВІДПОВІДАЮТЬ наступним обмеженням. Для ряду обмежень використайте регулярні вирази :
  - a) Назва поставщика повинна бути унікальна у рамках назв товарів. Тобто, наприклад поле SUPPLIER = 'Lg' не може бути у різних Product (TV-10).
  - b) Максимальна кількість товарів на полиці STORE.SHELF – 30. З урахуванням того, що STORE.QUANTITY – кількість товарів на полиці STORE.SHELF.
  - c) Діапазон дат: 01.01.2011 - 31.05.2014. Використайте регулярні вирази.
  - d) Одному і тому ж значенню поля ID\_STUFF повинні відповідати одні й ті ж значення полів STUFF\_NAME, E\_MAIL таблиці INVOICE
  - e) Типи операцій на складі – лише IN, OUT (незалежно від регістра) . Використайте регулярні вирази.?
  - f) Виведіть дерево категорій товарів з кількістю товарів на всіх рівнях.

## 2.Хід роботи

- 1) Діаграма предметної області «Книжковий супермаркет».



## 2) SQL запити

```
-- Назва поставщика повинна бути унікальна у рамках назв товарів.  
-- Тобто, наприклад поле SUPPLIER = 'Lg' не може бути у різних  
-- Product (TV-10).  
  
WITH supp_join AS (  
    SELECT st.product_id, supp.supplier_id  
    FROM store st  
    JOIN products pr ON st.product_id = pr.product_id  
    JOIN suppliers supp ON pr.supplier_id = supp.supplier_id  
)  
SELECT b.supplier_id, b.product_id  
FROM (  
    SELECT product_id  
    FROM supp_join  
    GROUP BY product_id  
    HAVING count(DISTINCT supplier_id) = 1  
) a  
JOIN supp_join b ON a.product_id = b.product_id;
```

```
-- Максимальна кількість товарів на полиці STORE.SHELF - 30.  
-- З урахуванням того, що STORE.QUANTITY - кількість товарів  
-- на полиці STORE.SHELF.  
  
SELECT shelf_id, sum(quantity) AS shelf_sum  
FROM store  
GROUP BY shelf_id  
HAVING sum(quantity) <= 30  
ORDER BY shelf_sum;
```

```
-- Діапазон дат: 01.01.2011 - 31.05.2014. Використайте  
-- регулярні вирази.  
  
SELECT *  
FROM invoices  
WHERE to_char(invoice_date, 'dd.mm.yyyy') ~  
'([0-9][0-9]\.[0-1][0-9]\.201[1-3])|([0-9]{2}\.[0-1][0-9]\.2014)';
```

--. Одному і тому ж значенню поля ID\_STUFF повинні відповідати  
--💡одні й ті ж значення полів STUFF\_NAME, E\_MAIL таблиці INVOICE.

```
WITH staff_inv AS (  
    SELECT staff.staff_id, staff_name, e_mail  
    FROM staff  
        JOIN store st ON st.staff_id = staff.staff_id  
        JOIN products pr ON pr.product_id = st.product_id  
        JOIN invoices inv ON inv.product_id = pr.product_id  
)  
SELECT staff_id  
FROM staff_inv  
GROUP BY staff_id  
HAVING count(DISTINCT staff_name) = 1  
AND count(DISTINCT e_mail) = 1;
```

-- Типи операцій на складі – лише IN, OUT (незалежно від регістра).  
-- Використайте регулярні вирази.

```
SELECT *  
FROM store  
WHERE oper_type::VARCHAR ~* '^(in|out)$';
```

```
-- Дерево каталогів

WITH RECURSIVE rec (id, path, path_name, name, count, level) AS (
    SELECT c.product_id,
           ARRAY [c.product_id],
           ARRAY [product_name::TEXT],
           product_name,
           (
               SELECT max(st.quantity)
               FROM products
               JOIN store st ON products.product_id = st.product_id
               GROUP BY c.product_id
           )::BIGINT,
           0
    FROM products c
    WHERE parent_product IS NULL
    UNION ALL
    SELECT c.product_id,
           array_append(r.path, c.product_id),
           array_append(r.path_name, c.product_name::TEXT),
           c.product_name,
           (SELECT count(*)
            FROM products
            WHERE c.parent_product = c.product_id),
           level + 1
    FROM products c
    JOIN rec r ON r.id = c.parent_product
)
SELECT lr.level,
       array_to_string(lr.path_name, ' -> ') AS path,
       (SELECT sum(sr.count) FILTER (WHERE lr.path <@ sr.path)
        FROM rec sr) AS all_sum
FROM rec lr
ORDER BY lr.level;
```

4) Додаткові таблиці ті типи, створені при нормалізації БД в 3НФ:

- `binddng_type`: ENUM («hard», «soft», «glossy») — тип палітурки книжки
- `genre`: ENUM ('adventure', 'novel', 'thriller', 'stories', 'documentation', 'sci-fi', 'children"s') — жанр книги
- `ordered_books` — замовлені книги, відноситься до `invoices` як «один до одного»
- `authors` — автори книжок, відноситься до `book_writes` як «один до багатьох»

- *book\_writes* — проміжна таблиця для того, щоб можна було реалізувати збереження інформації про написання одним автором багатьох книжок і кількома авторами однієї книги