

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

Звіт
до лабораторної роботи №1
з дисципліни «Екстримальне програмування»
на тему «Створення бази даних і заповнення тестовими даними»

Виконала
студентка групи КН-311
Мельничук М.П.

Прийняв
Щербак С.С.

Львів-2020

Мета: створення бази даних та заповнення її тестовими даними.

Теоретичні відомості

База даних (БД) — це організована структура, яка призначена для зберігання, зміни та обробки взаємозалежної інформації, переважно великих обсягів. БД використовують для динамічних сайтів з великими обсягами (інтернет-магазин, портал, корпоративний сайт).

Бази даних для сайтів дають змогу зберігати інформацію, що виглядає як зв'язані між собою таблиці. Саме в БД зберігаються вся необхідна та корисна інформація для функціонування сайту (клієнтські дані, прайс-лист, список товарів).

Щоб створити запит до бази даних часто використовують Structured Query Language. SQL дає змогу додавати, редагувати та видаляти інформацію, що міститься у таблицях.

Під час програмування сайтів використовують різні системи управління БД. До основних СУБД, відносять:

- об'єктно-реляційна система управління базами даних Oracle Database;
- вільна система управління базами даних PostgreSQL;
- система керування базами даних Microsoft SQL Сервер; вільна система управління базами даних MySQL;

Такі системи управління відрізняються централізованою обробкою запитів, забезпечують надійність, доступність та безпеку БД. Найбільш популярною системою управління є MySQL, вона дає зручний доступ для управління БД та підтримує велику кількість таблиць різних типів.

Розподілені бази даних (РБД) — це сукупність логічно пов'язаних між собою БД, які є розподіленими у комп'ютерній мережі.

Переваги РБД:

- ця модель відображає інформацію в найбільш простій для користувача формі;
- заснована на розвиненому математичному апараті, який дозволяє досить лаконічно описати основні операції з даними;
- дозволяє створювати мови маніпулювання даними не процедурного типу;
- маніпулювання даними на рівні вихідний БД і можливість зміни.

Недоліки РБД:

- найповільніший доступ до даних;

- трудомісткість розробки.

СУБД — це програмні засоби, які виступають посередником між БД та її користувачами. Завдяки сукупності мовних та програмних засобів, СУБД сприяють створенню, веденню та спільного використання БД різними користувачами. Сучасна програма СУБД складаються з ядра, процесору мови БД, підсистеми підтримки часу виконання та сервісних програм, які надають додаткові можливості для обслуговування інформаційних систем.

Хід роботи

Для розгортання бази даних було обрано MySQL, також для роботи додатку потрібно розробити скрипт який під час запуску проекту буде створювати базу даних і відповідні таблиці.

Вміст файлу CREATE_DB.sql:

```
DROP DATABASE placement_of_goods;
CREATE DATABASE placement_of_goods;
USE placement_of_goods;
CREATE TABLE categories(
id int primary key auto_increment,
name varchar(100) unique
);
CREATE TABLE goods(
id int primary key auto_increment,
name varchar(100),
category_id int,
FOREIGN KEY (category_id) REFERENCES categories(id)
);
CREATE TABLE customers(
id int primary key auto_increment,
name varchar(100),
surname varchar(100),
date date
);
CREATE TABLE users (
id int primary key auto_increment,
name varchar(100),
surname varchar(100),
email varchar(100),
password varchar(20)
);
CREATE TABLE shops(
id int primary key auto_increment,
name varchar(100),
country varchar(100),
city varchar(100),
owner int,
```

```

FOREIGN KEY (owner) REFERENCES users(id)
);
CREATE TABLE shops_categories(
id int primary key auto_increment,
shop_id int,
category_id int,
FOREIGN KEY (shop_id) REFERENCES shops(id),
FOREIGN KEY (category_id) REFERENCES categories(id)
);
CREATE TABLE locations(
id int primary key auto_increment,
name varchar(100),
distance float,
shop_id int,
side varchar(100),
FOREIGN KEY (shop_id) REFERENCES shops(id)
);
CREATE TABLE categories_locations(
id int key auto_increment,
global_location_id int,
category_id int,
location_id int,
date date,
FOREIGN KEY (category_id) REFERENCES shops_categories(id),
FOREIGN KEY (location_id) REFERENCES locations(id)
);
CREATE TABLE purchase (
id int primary key auto_increment,
purchase_id int,
shop_id int,
customer_id int,
good_id int,
location_id int,
date DATETIME DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (shop_id) REFERENCES shops(id),
FOREIGN KEY (good_id) REFERENCES goods(id),
FOREIGN KEY (customer_id) REFERENCES customers(id),
FOREIGN KEY (location_id) REFERENCES categories_locations(id)
);

```

Після виконання скрипту можемо побачити створену схему бази даних placement_of_goods:

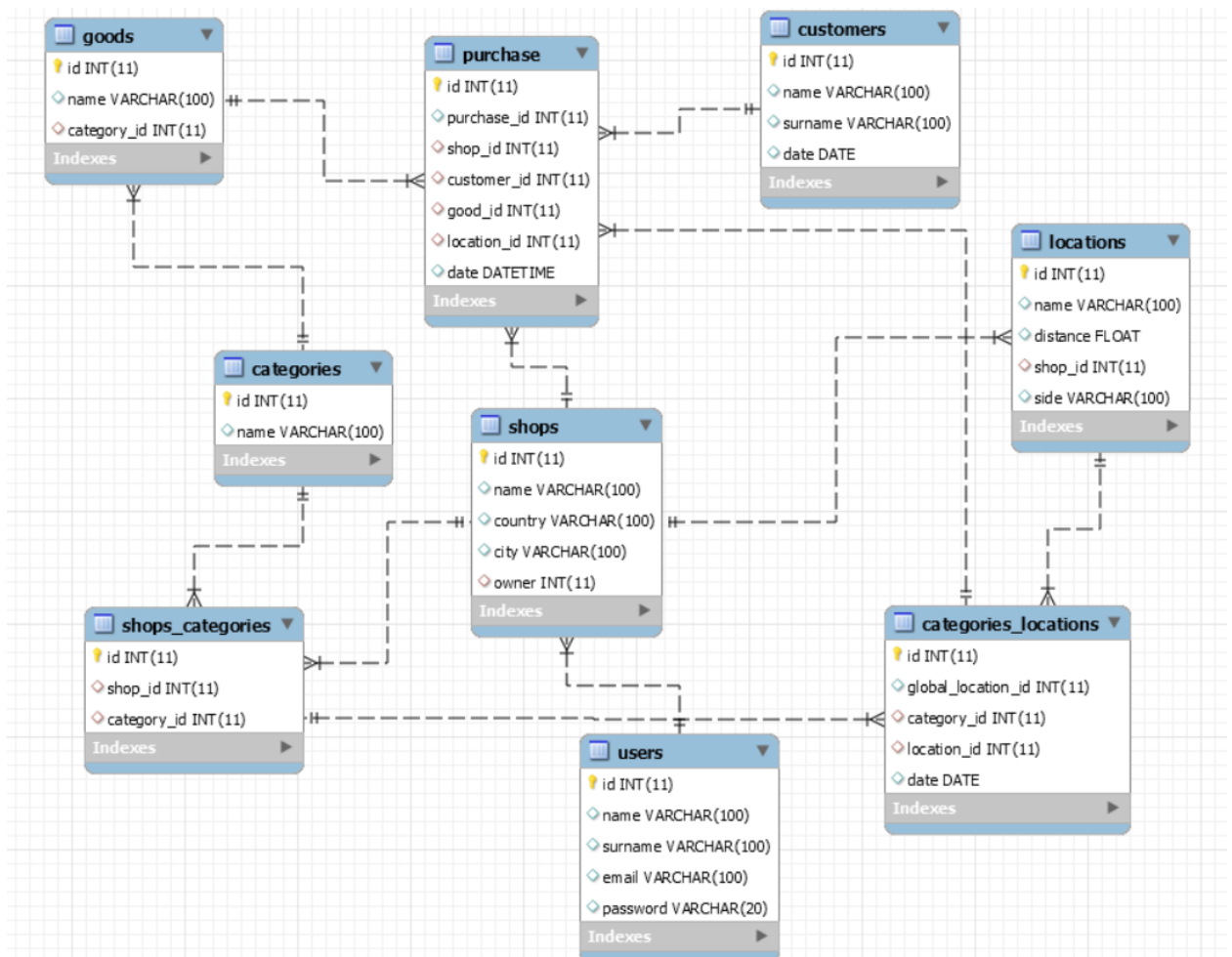


Рис. 1.1 Схема бази даних *placement_of_goods*

Також оскільки системі потрібна велика кількість даних для аналізу, при запуску проекту варто заповнити базу даних тестовими даними для зручності подальшої розробки і тестування.

Вміст файлу *PUBLISH_TEST_DATA.sql*:

```

USE placement_of_goods;
INSERT INTO users(name, surname, email, password)
VALUES('Maryna', 'Melnichuk', 'marynamelnichuk859@gmail.com', 'mary67123'),
('Olena', 'Fedorchuk', 'ofedorchuk@gmail.com', 'olena67123');
INSERT INTO shops(name, country, city, owner)
VALUES('ATB', 'USA', 'Ostin', 1),
('Metro', 'Ukraine', 'Lviv', 2);
INSERT INTO categories(name) VALUES('dairy products'), ('baked goods'), ('fruits'),
('vegetables'), ('frozen foods'), ('alcohol'), ('meat'), ('fish'), ('cereals'), ('confectionery');
INSERT INTO shops_categories(shop_id, category_id) VALUES(1,1), (1,2),
(1,3), (1,4), (1,5), (1,6), (1,7), (2,1), (2,2),
(2,3), (2,4), (2,5), (2,8), (2,9);
INSERT INTO goods(name, category_id) VALUES('apples', 3), ('cucumber', 4),
('bread', 2), ('milk', 1), ('yogurt', 1), ('kefir', 1), ('cheese', 1), ('vodka', 6),
('pork', 7), ('beef', 7), ('dumplings', 5), ('salmon', 8), ('rise', 9), ('buckwheat', 9),
('candy roshen', 10), ('cookies', 10), ('oranges', 3), ('potato', 4), ('pancakes', 5),
  
```

```

('beer',6), ('herring',8), ('buns',2), ('bulgur', 9), ('bananas', 3), ('tomatoes', 4);
INSERT INTO customers(name, surname, date) VALUES('Irina', 'Rishko', '2000-09-07'),
('Elizaveta', 'Koziurman', '2002-10-12'), ('Vasilina','Skorohoda','1999-01-02'),
('Eduard', 'Yarsyk', '1989-11-09'), ('Oleksandr', 'Kondrat', '1996-05-01');
INSERT INTO locations(name, distance, shop_id, side) VALUES('posithion_1', 1.5, 1, 'left'),
('posithion_2', 2.0, 1, 'right'), ('posithion_3', 2.5, 1, 'center'), ('posithion_4', 1.7, 1, 'left'),
('posithion_5', 2.0, 1, 'right'), ('posithion_6', 1.4, 2, 'center'), ('posithion_7', 1.5, 2, 'left'),
('posithion_8', 2.5, 2, 'left'), ('posithion_9', 2.0, 2, 'right'), ('posithion_10', 1.0, 2, 'right'),
('posithion_11', 1.5, 1, 'right'), ('posithion_12', 2.5, 2, 'center');
INSERT INTO categories_locations(global_location_id, category_id, location_id, date) VALUES(1,
1, 1, '2019-07-12'),
(1, 2, 2, '2019-07-12'),(1, 3, 3, '2019-07-12'),(1, 4, 4, '2019-07-12'),(1, 5, 5, '2019-07-12'),
(1, 6, 6, '2019-07-12'),(2, 7, 7, '2020-01-23'),(2, 8, 8, '2020-01-23'),(2, 9, 9, '2020-01-23'),
(2, 10, 10, '2020-01-23'),(2, 11, 11, '2020-01-23'),(2,12,12, '2020-01-23'), (2,1,1, '2020-01-23'),
(2,2,2, '2020-01-23');
INSERT INTO purchase(purchase_id, shop_id, customer_id, good_id, location_id) VALUES(1, 1,
1, 1, 1),
(2, 1, 2, 9, 1), (3, 2, 3, 11, 2), (4, 2, 4, 10, 2), (5, 2, 1, 2, 1), (6, 1, 3, 6, 1), (4, 2, 4, 5, 2),
(5, 2, 1, 7, 2), (1, 1, 1, 8, 1), (6, 1, 3, 9, 1), (7, 2, 2, 3, 2), (3, 2, 3, 8, 2), (5, 2, 1, 5, 2),
(6, 1, 3, 1, 1), (8, 1, 5, 3, 1), (9, 2, 5, 4, 2), (9, 2, 5, 12, 2), (10, 1, 3, 13, 1), (9, 2, 5, 14, 2);
Також після заповнення тестовими даними можемо побачити записи в
таблицях бази даних:

```

```
SELECT * FROM placement_of_goods.categories;
```

	id	name
▶	6	alcohol
	2	baked goods
	9	cereals
	10	confectionery
	1	dairy products
	8	fish
	5	frozen foods
	3	fruits
	7	meat
	4	vegetables

Рис. 1.2 Записи в таблиці categories

```
SELECT * FROM placement_of_goods.purchase;
```

	id	purchase_id	shop_id	customer_id	good_id	location_id	date
►	1	1	1	1	1	1	2020-05-11 00:12:39
	2	2	1	2	9	1	2020-05-11 00:12:39
	3	3	2	3	11	2	2020-05-11 00:12:39
	4	4	2	4	10	2	2020-05-11 00:12:39
	5	5	2	1	2	1	2020-05-11 00:12:39
	6	6	1	3	6	1	2020-05-11 00:12:39
	7	4	2	4	5	2	2020-05-11 00:12:39
	8	5	2	1	7	2	2020-05-11 00:12:39
	9	1	1	1	8	1	2020-05-11 00:12:39
	10	6	1	3	9	1	2020-05-11 00:12:39
	11	7	2	2	3	2	2020-05-11 00:12:39
	12	3	2	3	8	2	2020-05-11 00:12:39

Рис. 1.3 Записи в таблиці purchase

Висновок: в результаті лабораторної роботи було створено базу даних і заповнено її тестовими даними.