



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Системи та засоби інтерактивної аналітики

Лабораторна робота №2
Варіант №6

Виконав:
Поночевний Назар ФІ-92
Перевірив:
Тітков Д. В.

ЛАБОРАТОРНА РОБОТА №2

Нормальні форми у БД

Тема: Нормалізація БД.

Завдання: Навчитися створювати БД, які задовольняють вимогам нормальних форм.

Робоче завдання

1. Створити базу даних, назву вибрати самостійно.
2. Варіанти у Табл 2.1
 - 1) Задати для всіх таблиць типи полів.
 - 2) Структуру таблиць вибрати керуючись логікою і правилами реляційних БД у 3-й нормальній формі.
 - 3) Варіанти за нашим списком студентів, 11 варіант = 1. 12 = 2 далі, аналогічно.
 - 4) Крапка з комою (;) розділяє поля, кома (,) розділяє декілька значень в одному полі.
3. Створити звіт. Приєднати до класу.
 - 1) У звіті навести все необхідне для повторення і перевірки ваших дій (SQL запити для створення БД і таблиць, структуру БД, і т.д). Навести знімки екрана, які підтверджують виконані дії.
 - 2) Зробити висновки по роботі та занести їх у звіт.
4. Підготувати відповіді на контрольні питання (для офлайн захисту навести їх у протоколі).
5. Захистити роботу.

Варіант № 6

Магазин рослин, облік

№; Назва; Склад; Постачальник; Особливості; Відповідальні

1; Фікус Каріка; №1; UGT, AVDtrade; Садові, Закритого ґрунту; Луцик М. В., Ступак Я. К.

2; Аденіум огрядний; №2; AVDtrade; Кімнатні;Падик В. О.

3; Аглаонема; №1, №3; КвітиУкраїни; Кімнатні; Ступак Я. К.

...

341; Азалія; №1; Волошка; Садові; Луцик М. В.

1) Код з коментарями:

```
-- Create database and tables
```

```
CREATE DATABASE plant_store;
```

```
USE plant_store;
```

```
DROP TABLE IF EXISTS catalog, employees_sets_employees,  
features_sets_features, suppliers_sets_suppliers, plants, warehouses,  
suppliers, suppliers_sets, features, features_sets, employees,  
employees_sets;
```

```
CREATE TABLE plants (
```

```
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL UNIQUE  
);  
  
CREATE TABLE warehouses (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL UNIQUE  
);  
  
CREATE TABLE suppliers (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL UNIQUE  
);  
  
CREATE TABLE suppliers_sets (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NULL UNIQUE  
);  
  
CREATE TABLE suppliers_sets_suppliers (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    suppliers_set_id INT UNSIGNED NOT NULL,  
    supplier_id INT UNSIGNED NOT NULL,  
    FOREIGN KEY (suppliers_set_id) REFERENCES suppliers_sets(id) ON  
UPDATE CASCADE,  
    FOREIGN KEY (supplier_id) REFERENCES suppliers(id) ON UPDATE  
CASCADE  
);  
  
CREATE TABLE features (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL UNIQUE  
);  
  
CREATE TABLE features_sets (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NULL UNIQUE  
);  
  
CREATE TABLE features_sets_features (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    features_set_id INT UNSIGNED NOT NULL,  
    feature_id INT UNSIGNED NOT NULL,
```

```

        value DOUBLE NULL,
        FOREIGN KEY (features_set_id) REFERENCES features_sets(id) ON
UPDATE CASCADE,
        FOREIGN KEY (feature_id) REFERENCES features(id) ON UPDATE
CASCADE
    );

CREATE TABLE employees (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(255) NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    middle_name VARCHAR(255) NULL
);

CREATE TABLE employees_sets (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NULL UNIQUE
);

CREATE TABLE employees_sets_employees (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    employees_set_id INT UNSIGNED NOT NULL,
    employee_id INT UNSIGNED NOT NULL,
    FOREIGN KEY (employees_set_id) REFERENCES employees_sets(id) ON
UPDATE CASCADE,
    FOREIGN KEY (employee_id) REFERENCES employees(id) ON UPDATE
CASCADE
);

CREATE TABLE catalog (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    plant_id INT UNSIGNED NOT NULL,
    warehouse_id INT UNSIGNED NOT NULL,
    suppliers_set_id INT UNSIGNED NOT NULL,
    features_set_id INT UNSIGNED NOT NULL,
    employees_set_id INT UNSIGNED NULL,
    FOREIGN KEY (plant_id) REFERENCES plants(id) ON UPDATE CASCADE,
    FOREIGN KEY (warehouse_id) REFERENCES warehouses(id) ON UPDATE
CASCADE,
    FOREIGN KEY (suppliers_set_id) REFERENCES suppliers_sets(id) ON
UPDATE CASCADE,
    FOREIGN KEY (features_set_id) REFERENCES features_sets(id) ON
UPDATE CASCADE,

```

```

    FOREIGN KEY (employees_set_id) REFERENCES employees_sets(id) ON
UPDATE CASCADE
);

-- Insert data into them

START TRANSACTION;

INSERT INTO plants (name) VALUES
    ('Ficus Carica'),
    ('Adenium fat'),
    ('Aglaonema'),
    ('Azalea');
INSERT INTO warehouses (name) VALUES ('No1'), ('No2'), ('No3');
INSERT INTO suppliers (name) VALUES ('UGT'), ('AVDtrade'), ('Flowers
of Ukraine'), ('Voloshka');
INSERT INTO features (name) VALUES ('Garden'), ('Closed ground'),
('Indoor'), ('Height');
INSERT INTO employees (first_name, last_name, middle_name) VALUES
    ('M.', 'Lucyk', 'V.'),
    ('Y.', 'Stupak', 'K.'),
    ('V.', 'Padik', 'O.');

INSERT INTO suppliers_sets (name) VALUES ('UGT+ACDt Group'),
('AVDt'), ('FoU'), ('Vol');
INSERT INTO suppliers_sets_suppliers (suppliers_set_id, supplier_id)
VALUES
    (1, 1), (1, 2),
    (2, 2),
    (3, 3),
    (4, 4);

INSERT INTO features_sets (name) VALUES ('Gard'), ('Gard Closed'),
('Ind');
INSERT INTO features_sets_features (features_set_id, feature_id,
value) VALUES
    (1, 1, NULL),
    (2, 1, NULL), (2, 2, NULL), (2, 4, 100),
    (3, 3, NULL), (3, 4, 80);

INSERT INTO employees_sets (name) VALUES ('Group 1'), ('Group 2');
INSERT INTO employees_sets_employees (employees_set_id, employee_id)
VALUES

```

```
(1, 1), (1, 2),  
(2, 3);
```

```
INSERT INTO catalog (plant_id, warehouse_id, suppliers_set_id,  
features_set_id, employees_set_id) VALUES
```

```
(1, 1, 1, 2, 1),  
(2, 2, 2, 3, 2),  
(3, 1, 3, 3, NULL),  
(3, 3, 3, 3, NULL),  
(4, 1, 4, 1, 1);
```

```
COMMIT;
```

```
-- Show the data to test if everything is correct
```

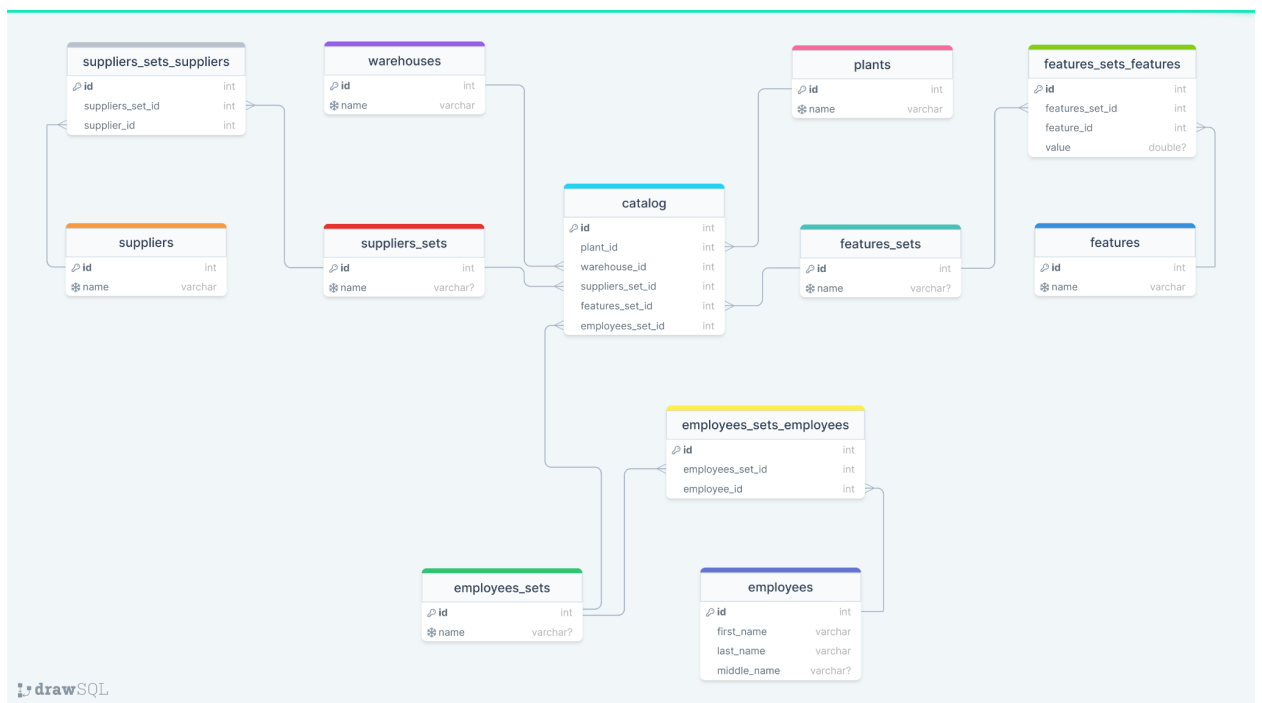
```
SELECT plants.name AS Plant,  
       warehouses.name AS Warehouse,  
       (SELECT GROUP_CONCAT(suppliers.name SEPARATOR ', ')  
        FROM plant_store.suppliers  
         INNER JOIN plant_store.suppliers_sets_suppliers  
           ON suppliers.id = suppliers_sets_suppliers.supplier_id  
         INNER JOIN plant_store.suppliers_sets  
           ON suppliers_sets_suppliers.suppliers_set_id =  
suppliers_sets.id  
        WHERE suppliers_sets.id = catalog.suppliers_set_id) AS  
Suppliers,  
       (SELECT GROUP_CONCAT(CASE  
                             WHEN features_sets_features.value IS  
NULL THEN features.name  
                             ELSE CONCAT(features.name, ' < ',  
features_sets_features.value)  
                             END SEPARATOR ', ')  
        FROM plant_store.features  
         INNER JOIN plant_store.features_sets_features  
           ON features.id = features_sets_features.feature_id  
         INNER JOIN plant_store.features_sets  
           ON features_sets_features.features_set_id =  
features_sets.id  
        WHERE features_sets.id = catalog.features_set_id) AS  
Features,  
       (SELECT GROUP_CONCAT(CONCAT(employees.first_name, ' ',  
employees.last_name) SEPARATOR ', ')  
        FROM plant_store.employees
```

```

INNER JOIN plant_store.employees_sets_employees
  ON employees.id = employees_sets_employees.employee_id
INNER JOIN plant_store.employees_sets
  ON employees_sets_employees.employees_set_id =
employees_sets.id
WHERE employees_sets.id = catalog.employees_set_id) AS
Responsible
FROM plant_store.catalog
  INNER JOIN plant_store.plants ON catalog.plant_id = plants.id
  INNER JOIN plant_store.warehouses ON catalog.warehouse_id =
warehouses.id
  LEFT JOIN plant_store.suppliers_sets ON catalog.suppliers_set_id =
suppliers_sets.id
  LEFT JOIN plant_store.features_sets ON catalog.features_set_id =
features_sets.id
  LEFT JOIN plant_store.employees_sets ON catalog.employees_set_id =
employees_sets.id;

```

2) ER-діаграма:



3) Знімки екрана, які підтверджують виконані дії:

Database: MySQL v5.7
 Run Save Load Example Collaborate

Sign in Have any feedback?

Results

Query #1 Execution time: 2ms
 Copy as Markdown

Plant	Warehouse	Suppliers	Features	Responsible
Ficus Carica	No1	UGT, AVDtrade	Garden, Closed ground, Height < 100	M. Lucyk, Y. Stupak
Adenium fat	No2	AVDtrade	Indoor, Height < 80	V. Padik
Aglaonema	No1	Flowers of Ukraine	Indoor, Height < 80	null
Aglaonema	No3	Flowers of Ukraine	Indoor, Height < 80	null
Azalea	No1	Voloshka	Garden	M. Lucyk, Y. Stupak

Висновки

В цій лабораторній роботі я створив базу даних під назвою "магазин рослин" та визначив таблиці "рослини", "склади", "каталог", "постачальники", "постачальники_набори", "постачальники_набори_постачальників", "характеристики", "характеристики_набори", "характеристики_набори_характеристик", "співробітники", "співробітники_набори", "співробітники_набори_співробітників". Я встановив типи полів для кожної таблиці та встановив зв'язки між ними за допомогою зовнішніх ключів. Я також нормалізував базу даних до 3-ї нормальної форми, щоб мінімізувати надмірність і залежність. Наостанок я вставив дані в таблиці за допомогою команди INSERT. Ця лабораторна робота продемонструвала моє вміння структурувати та створювати базу даних в MySQL, визначати зв'язки між таблицями за допомогою зовнішніх ключів та нормалізувати структуру бази даних.