

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

**Лабораторна робота № 2**

з дисципліни «Професійне використання SQL та PL/SQL для  
СУБД Oracle»

Тема: «Створення даталогічної моделі за варіантом»

**ВАРІАНТ 11**

Виконали:

студент групи

**ІМ-13 Первєєв Євгеній**

**Олексійович**

Перевірила:

Ксенія

Ульяницька

Київ 2024

**Тема:** Створення даталогічної моделі за варіантом.

**Мета:** За отриманим варіантом розробити власну даталогічну модель; для власної структури створити окрему схему даних та відпрацювати перелік простіших запитів SQL.

### **Хід роботи**

Варіант:

#### **11. АІС аптеки**

Аптека продає медикаменти і виготовляє їх за рецептами.

Ліки можуть бути різних типів: 1) готові ліки: таблетки, мазі, настоянки; 2) виготовлені аптекою: мікстури, мазі, розчини, настойки, порошки. Різниця в типах ліків відбивається в різному наборі атрибутів, що їх характеризують. Мікстури і порошки виготовляються тільки для внутрішнього застосування, розчини для зовнішнього, внутрішнього застосування і для змішування з іншими ліками та мазі тільки для зовнішнього застосування. Ліки різні також за способом приготування і за часом приготування. Порошки і мазі виготовляються змішуванням різних компонент. При виготовленні розчинів і мікстур інгредієнти не тільки змішують, але і відстоюють з подальшою фільтрацією лік, що збільшує час виготовлення.

В аптеці існує довідник технологій приготування різних ліків. У ньому зазначаються: ідентифікаційний номер технології, назву ліків і сам спосіб приготування.

## 1) Аналіз наданої предметної області та створення ER моделі.

- “Ліки можуть бути різних типів: 1) готові ліки: таблетки, мазі, настоянки; 2) виготовлені аптекою: мікстури, мазі, розчини, настойки, порошки. Різниця в типах ліків відбивається в різному наборі атрибутів, що їх характеризують.”

Unfinished_Medicines	
ID	number
Name	varchar2
A_Method_ID	number
P_Method_ID	number
Recipe_ID	number
P_Time	number

Finished_Medicines	
ID	number
Name	varchar2
A_Method_ID	number

Створено дві таблиці з різними атрибутами для двох типів ліків

- “. Мікстури і порошки виготовляються тільки для внутрішнього застосування, розчини для зовнішнього, внутрішнього застосування і для змішування з іншими ліками та мазі тільки для зовнішнього застосування.”

Application_Methods	
ID	number
Name	varchar2

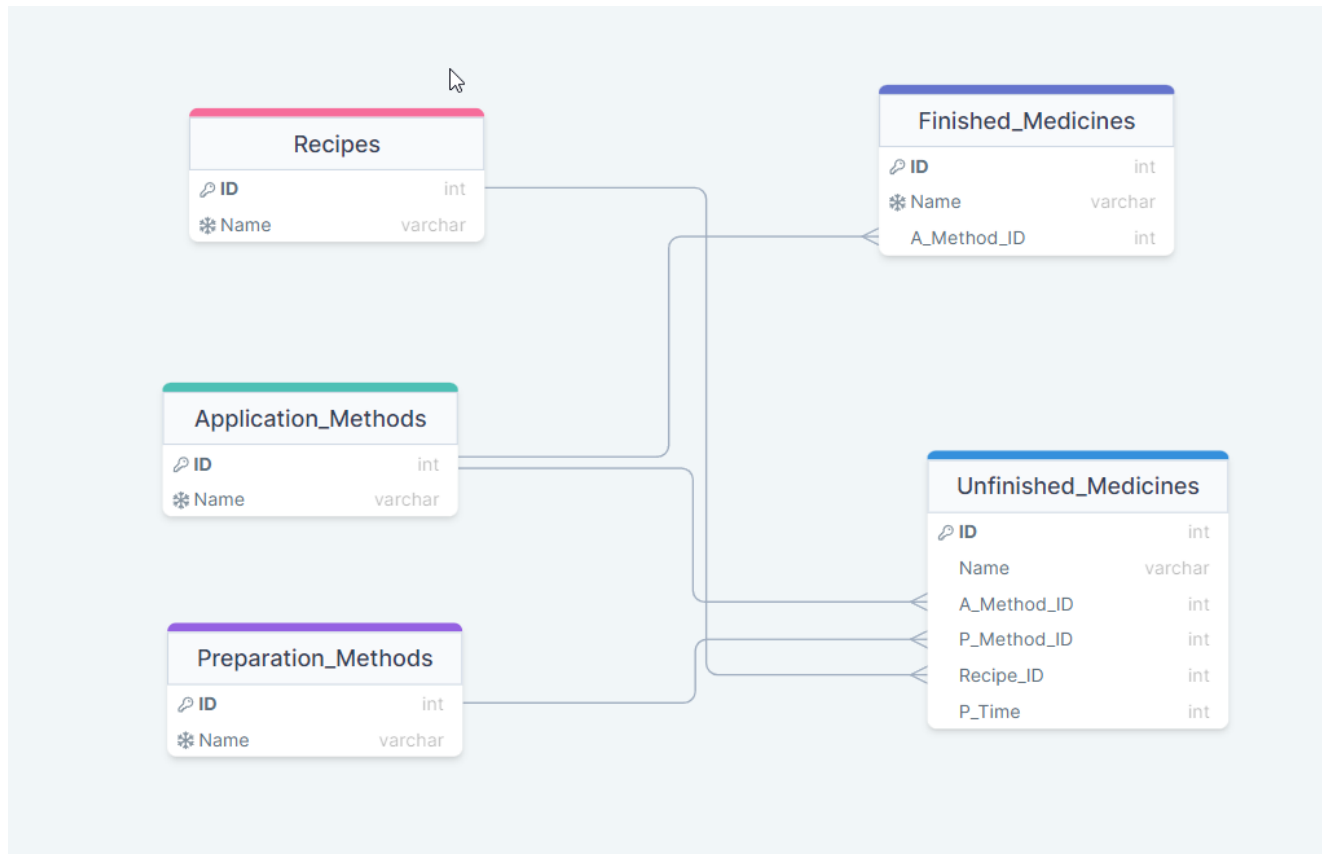
Створено відповідну таблицю в якій будуть міститись 3 значення(для зовнішнього застосування, для внутрішнього застосування та для змішування).

- “Ліки різні також за способом приготування і за часом приготування. Порошки і мазі виготовляються змішуванням різних компонент. При виготовленні розчинів і мікстур інгредієнти не тільки змішують, але і відстоюють з подальшою фільтрацією лік, що збільшує час виготовлення.”

Preparation_Methods	
ID	number
Name	varchar2

Судячи з тексту є два способи приготування це змішування для мазі і порошків, та змішування з відстоюванням та фільтрацією(іде як один метод) для мікстур розчинів і тд.

- “В аптеці існує довідник технологій приготування різних ліків. У ньому зазначаються: ідентифікаційний номер технології, назву ліків і сам спосіб приготування.”



## **2) Створення та наповнення БД.**

```
CREATE TABLE Recipes (  
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    Name VARCHAR2(20),  
    Ingredients VARCHAR2(20)  
);  
  
CREATE TABLE Application_Methods (  
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    Name VARCHAR2(20)  
);  
  
CREATE TABLE Preparation_Methods (  
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    Name VARCHAR2(20)  
);  
  
CREATE TABLE Finished_Medicines (  
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    Name VARCHAR2(20),  
    A_Method_ID NUMBER,  
    FOREIGN KEY (A_Method_ID) REFERENCES Application_Methods(ID)  
);  
  
CREATE TABLE Unfinished_Medicines (  
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    Name VARCHAR2(20),  
    A_Method_ID NUMBER,  
    P_Method_ID NUMBER,  
    Recipe_ID NUMBER,  
    P_Time NUMBER,  
    FOREIGN KEY (A_Method_ID) REFERENCES Application_Methods(ID),  
    FOREIGN KEY (P_Method_ID) REFERENCES Preparation_Methods(ID),  
    FOREIGN KEY (Recipe_ID) REFERENCES Recipes(ID)  
);
```

**ДОДАНО ОБМЕЖЕННЯ ЦІЛІСНОСТІ**

```

ALTER TABLE Recipes
ADD CONSTRAINT unique_name_recipes UNIQUE (Name);

ALTER TABLE Recipes
MODIFY Name NOT NULL;

ALTER TABLE Application_Methods
ADD CONSTRAINT unique_name_application_methods UNIQUE (Name);

ALTER TABLE Application_Methods
MODIFY Name NOT NULL;

ALTER TABLE Preparation_Methods
ADD CONSTRAINT unique_name_preparation_methods UNIQUE (Name);

ALTER TABLE Preparation_Methods
MODIFY Name NOT NULL;

```

### **Скрипти для наповнення**

```

ALTER TABLE Recipes MODIFY (Name VARCHAR2(50));
ALTER TABLE Recipes MODIFY (Ingredients VARCHAR2(50));
ALTER TABLE Application_Methods MODIFY (Name VARCHAR2(50));
ALTER TABLE Preparation_Methods MODIFY (Name VARCHAR2(50));
ALTER TABLE Finished_Medicines MODIFY (Name VARCHAR2(50));
ALTER TABLE Unfinished_Medicines MODIFY (Name VARCHAR2(50));

INSERT INTO Recipes (Name, Ingredients) VALUES ('Recipe1', 'Ingredient1');
INSERT INTO Recipes (Name, Ingredients) VALUES ('Recipe2', 'Ingredient2');
INSERT INTO Recipes (Name, Ingredients) VALUES ('Recipe3', 'Ingredient3');

UPDATE Recipes SET Ingredients = 'Інгредієнт1' , Name = 'Рецепт1' where ID = 1;
UPDATE Recipes SET Ingredients = 'Інгредієнт2' , Name = 'Рецепт2' where ID = 2;
UPDATE Recipes SET Ingredients = 'Інгредієнт3' , Name = 'Рецепт3' where ID = 3;

INSERT INTO Application_Methods (Name) VALUES ('Зовнішнє заст. ');
INSERT INTO Application_Methods (Name) VALUES ('Внутрішнє заст. ');
INSERT INTO Application_Methods (Name) VALUES ('Для змішування');

INSERT INTO Preparation_Methods (Name) VALUES ('Змішування');
INSERT INTO Preparation_Methods (Name) VALUES ('Зміш-відст-фільтр');

INSERT INTO Finished_Medicines (Name, A_Method_ID) VALUES ('Ношпа', 1);
INSERT INTO Finished_Medicines (Name, A_Method_ID) VALUES ('Мазь Вишневського', 2);
INSERT INTO Finished_Medicines (Name, A_Method_ID) VALUES ('Аспірин', 2);

```

```
UPDATE Finished_Medicines SET A_Method_ID = '1' WHERE name ='Аспірин';
```

```
INSERT INTO Finished_Medicines (Name, A_Method_ID) VALUES ('Вазелін', 2);
```

```
INSERT INTO Finished_Medicines (Name, A_Method_ID) VALUES ('Аспаркам', 1);
```

```
INSERT INTO Finished_Medicines (Name, A_Method_ID) VALUES ('Детралекс', 1);
```

```
INSERT INTO Unfinished_Medicines (Name, A_Method_ID, P_Method_ID, Recipe_ID, P_Time)  
VALUES ('Мазь за рец1', 2, 1, 1, 15);
```

```
INSERT INTO Unfinished_Medicines (Name, A_Method_ID, P_Method_ID, Recipe_ID, P_Time)  
VALUES ('Розчин за рец2', 3, 2, 2, 10);
```

```
INSERT INTO Unfinished_Medicines (Name, A_Method_ID, P_Method_ID, Recipe_ID, P_Time)  
VALUES ('Мікстура за рец1', 1, 2, 3, 30);
```

### **3) Базові запити до бд, тестування.**

**--Запит на створення репорту по препаратам які потребують приготування**  
**SELECT**

```
    unfinished_medicines.name AS Medicine_Name,  
    recipes.name AS Recipe,  
    preparation_methods.name AS Preparation_Method,  
    application_methods.name AS Application_Method,  
    unfinished_medicines.p_time AS Preparation_Time
```

**FROM**

```
    unfinished_medicines
```

**JOIN**

```
    application_methods ON unfinished_medicines.a_method_id = application_methods.id
```

**JOIN**

```
    preparation_methods ON unfinished_medicines.p_method_id = preparation_methods.id
```

**JOIN**

```
    recipes ON unfinished_medicines.recipe_id = recipes.id;
```

**-- Запит з використанням псевдонімів таблиць**

**SELECT**

```
    UM.name AS Medicine_Name,  
    R.name AS Recipe_Name,  
    PM.name AS Preparation_Method,  
    AM.name AS Application_Method,  
    UM.p_time AS Preparation_Time
```

**FROM**

```
    Unfinished_Medicines UM
```

JOIN

Application\_Methods AM ON UM.a\_method\_id = AM.id

JOIN

Preparation\_Methods PM ON UM.p\_method\_id = PM.id

JOIN

Recipes R ON UM.recipe\_id = R.id;

-- Запит на отримання готових препаратів та методу їх застосування

SELECT

FM.name AS Medicine\_Name,  
AM.name AS Application\_Method

FROM

finished\_medicines FM

JOIN

application\_methods AM ON FM.a\_method\_id = AM.id;

The screenshot shows a database query editor with multiple tabs. The active tab is 'APPLICATION\_METHODS'. The query editor displays a SQL query that joins 'unfinished\_medicines' with 'application\_methods', 'preparation\_methods', and 'recipes'. Below the query editor, the 'Query Result' tab is active, showing a table with 5 columns: MEDICINE\_NAME, RECIPE, PREPARATION\_METHOD, APPLICATION\_METHOD, and PREPARATION\_TIME. The table contains 3 rows of data.

```
SELECT
unfinished_medicines.name AS Medicine_Name,
recipes.name AS Recipe,
preparation_methods.name AS Preparation_Method,
application_methods.name AS Application_Method,
unfinished_medicines.p_time AS Preparation_Time
FROM
unfinished_medicines
JOIN
application_methods ON unfinished_medicines.a_method_id = application_methods.id
JOIN
preparation_methods ON unfinished_medicines.p_method_id = preparation_methods.id
JOIN
recipes ON unfinished_medicines.recipe_id = recipes.id;
```

```
SELECT
UM.name AS Medicine_Name,
R.name AS Recipe_Name,
PM.name AS Preparation_Method,
AM.name AS Application_Method,
```

MEDICINE_NAME	RECIPE	PREPARATION_METHOD	APPLICATION_METHOD	PREPARATION_TIME
1 Мазь за рец1	Рецепт1	Змішування	Внутрішнє заст.	15
2 Розчин за рец2	Рецепт2	Зміш-відст-фільтр	Для змішування	10
3 Мікстура за рец1	Рецепт3	Зміш-відст-фільтр	Зовнішнє заст.	30



```
SELECT
    FM.name AS Medicine_Name,
    AM.name AS Application_Method
FROM
    finished_medicines FM
JOIN
    application_methods AM ON FM.a_method_id = AM.id;
```

Query Result x

SQL | All Rows Fetched: 6 in 0,003 seconds

	MEDICINE_NAME	APPLICATION_METHOD
1	Ношпа	Зовнішнє заст.
2	Мазь Вишневського	Внутрішнє заст.
3	Аспірин	Зовнішнє заст.
4	Вазелін	Внутрішнє заст.
5	Аспаркам	Зовнішнє заст.
6	Детралекс	Зовнішнє заст.

## Висновок

Під час виконання лабораторної роботи було розроблену власну даталогічну модель, створена схема даних та відпрацьовано скрипти для її перевірки.

## Захист

```

CREATE TABLE Students (
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY NOT NULL,
    Name VARCHAR2(20) NOT NULL,
    Surname VARCHAR2(20) NOT NULL
);

```

```

CREATE TABLE Courses (
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY NOT NULL,
    Name VARCHAR2(20) NOT NULL UNIQUE,
    Description VARCHAR2(100)
);

```

```

CREATE TABLE Students_Courses (
    ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    student_id NUMBER NOT NULL,
    course_id NUMBER NOT NULL,
    FOREIGN KEY (student_id) REFERENCES Students(ID),

```

```
FOREIGN KEY (course_id) REFERENCES Courses(ID)
);
```

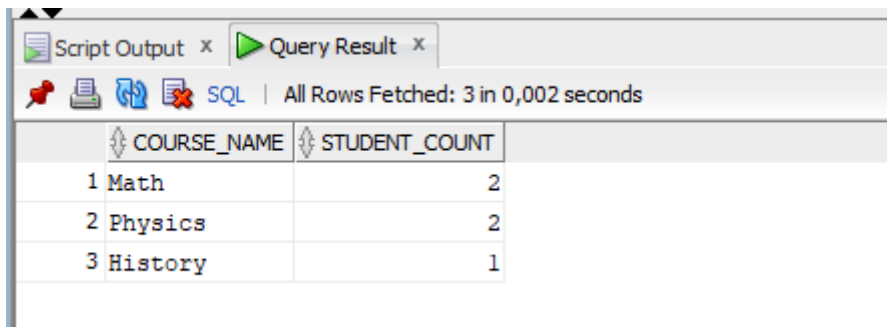
```
INSERT INTO Students (Name, Surname) VALUES ('John', 'Doe');
INSERT INTO Students (Name, Surname) VALUES ('Alice', 'Smith');
INSERT INTO Students (Name, Surname) VALUES ('Bob', 'Johnson');
```

```
INSERT INTO Courses (Name, Description) VALUES ('Math', 'Basic Mathematics Course');
INSERT INTO Courses (Name, Description) VALUES ('Physics', 'Introduction to Physics');
INSERT INTO Courses (Name, Description) VALUES ('History', 'World History Overview');
```

```
INSERT INTO Students_Courses (student_id, course_id) VALUES (1, 1); -- John на курс математики
INSERT INTO Students_Courses (student_id, course_id) VALUES (1, 2); -- John на курс фізики
INSERT INTO Students_Courses (student_id, course_id) VALUES (2, 1); -- Alice на курс математики
INSERT INTO Students_Courses (student_id, course_id) VALUES (3, 3); -- Bob на курс історії
INSERT INTO Students_Courses (student_id, course_id) VALUES (3, 2); -- Bob на курс фізики
```


```
--Отримання кількості студентів на кожному курсі
SELECT c.Name AS Course_Name, COUNT(sc.student_id) AS Student_Count
FROM Courses c
LEFT JOIN Students_Courses sc ON c.ID = sc.course_id
GROUP BY c.Name;
```

```
--Отримання студентів та їх курсів
SELECT s.Name, s.Surname, c.Name AS Course_Name, c.Description AS Course_Description
FROM Students s
JOIN Students_Courses sc ON s.ID = sc.student_id
JOIN Courses c ON sc.course_id = c.ID;
```



	COURSE_NAME	STUDENT_COUNT
1	Math	2
2	Physics	2
3	History	1

Script Output xQuery Result x

 | All Rows Fetched: 5 in 0,014 seconds

	NAME	SURNAME	COURSE_NAME	COURSE_DESCRIPTION
1	John	Doe	Math	Basic Mathematics Course
2	Alice	Smith	Math	Basic Mathematics Course
3	John	Doe	Physics	Introduction to Physics
4	Bob	Johnson	Physics	Introduction to Physics
5	Bob	Johnson	History	World History Overview