

ВИВЧЕННЯ ІНФОРМАЦІЙНИХ МОДЕЛЕЙ ЗА ДОПОМОГОЮ СУБД LIBREOFFICE BASE

В статті розглянуто питання щодо використання СУБД LIBREOFFICE BASE для побудови інформаційних моделей даних. Розглянуто особливості реалізації SQL-моделі даних в цій СУБД. На основі розгляду практико-орієнтованої задачі розглянуто процес побудови відповідних інформаційних моделей даних (ER та SQL), та реалізацію SQL моделі засобами даної СУБД. Детально розглянуто питання, що стосуються розробки запитів, як на вибірку даних, так і на їх модифікацію в цій СУБД.

Наведено низку прикладів виконання запитів до розробленої БД як засобами мови SQL, так і за допомогою графічного інтерфейсу. Зроблено огляд проблем, що виникають при формулюванні запитів на обчислення в 4-й версії LIBREOFFICE BASE і наведено шляхи їх розв'язання. Матеріал статті може бути використаний в практичній роботі викладача інформаційних технологій та в процесі самостійної підготовки різних категорій користувачів до ефективної роботи з базами даних.

Ключові слова: навчання інформатики, інформаційна модель, LibreOffice Base, SQL.

У Чернігівському національному педагогічному університеті на кафедрі інформатики в якості навчальної СУБД обрано СУБД LibreOffice Base. Такий вибір зумовлений тим, що дана СУБД є складовою частиною пакету офісних програм LibreOffice, який є кросплатформним, а також вільно поширюваним. СУБД LibreOffice Base з цього пакету є потужною реляційною СУБД, використання якої дає змогу створювати складні бази даних і використовувати мову SQL для формування запитів. В актуальній 4-й версії запити на вибірку можна створювати і за допомогою мови QBE, але для запитів на оновлення, додавання та вилучення даних потрібно обов'язково застосовувати мову SQL, причому реалізовано це таким чином, що мінімізується можливість необдуманого багаторазового запуску таких запитів. Такий спосіб реалізації запитів на модифікацію бази даних більш вдалиий, ніж наприклад в СУБД MS ACCESS, де запити на модифікацію можуть запускатися недосвідченими студентами багаторазово без будь-яких обмежень, і практика роботи з СУБД MS ACCESS показує, що це часто призводить до спотворення даних.

Для формування у студентів компетентностей щодо аналізу предметної галузі задачі, побудови інформаційної моделі (в даному випадку спочатку ER-моделі, а на її основі SQL-моделі), та дослідження отриманої моделі студентам пропонується цикл з трьох лабораторних робіт. У першій роботі студенти повинні на основі умови задачі у термінах предметної галузі розробити адекватну модель даних, визначивши сутності предметної галузі, їх властивості та зв'язки між ними; на основі цієї моделі даних створити таблиці, встановити зв'язки між таблицями, створити форми для введення і редагування даних з визначенням обмежень на значення у полях. Заповнити таблиці даними. У другій лабораторній роботі студенти повинні розробити запити до бази даних з попередньої лабораторної роботи. У третій лабораторній роботі студенти повинні розробити до бази даних, розробленої у першій лабораторній роботі. Завдання у лабораторних роботах містять 12 варіантів, таким чином кожен студент підгрупи виконує власний варіант.

Наведемо етапи розв'язування (побудова моделі даних, бази даних, форм, запитів, звітів) задачі, подібної до тих, що є в умовах лабораторних робіт.

Задача 1

Створити базу даних FIRMA, в якій зберігатимуться дані про співробітників деякої фірми і зроблені їм виплати заробітної плати упродовж деякого періоду (не більше однієї виплати в день).

У базі даних мають зберігатися такі відомості про кожного співробітника:

- табельний номер;
- прізвище, ім'я, по-батькові;
- дата народження;
- домашня адреса;
- посада у фірмі;
- стаж роботи;
- дата отримання заробітної плати;
- отримана сума заробітної плати.

Розв'язування задачі почнемо з побудови ER-моделі даних, тобто моделі "Сутність-Зв'язок". З умови задачі можна виділити наступні сутності предметної галузі: "Співробітник фірми" та "Виплата". Ці сутності пов'язані відношенням "Один до багатьох", оскільки кожен співробітник фірми отримує багато виплат. Тепер визначимо атрибути кожної з сутностей. За умовою задачі сутність "Співробітник фірми" має атрибути "Табельний номер", "Прізвище ім'я по-батькові (ПІБ)", "Дата народження", "Адреса" (домашня адреса), "Посада" та "Стаж", а сутність "Виплата" має атрибути "Дата виплати" (коли зроблена виплата) та "Сума" (розмір виплати). Первинним ключем сутності "Співробітник фірми" є атрибут "Табельний номер", а сутності "Виплата" – "Дата виплати" (оскільки за умовою задачі не можна робити більше однієї виплати за день).

Далі необхідно перетворити отриману ER-модель в реляційну (SQL) модель. Для кожної сутності створюється відповідна таблиця, полями якої стають атрибути сутності, а первинний ключ сутності стає первинним ключем таблиці. Отримаємо таблиці, які наведемо "Співробітники" та "Виплати".

Для встановлення зв'язків "один до багатьох" між таблицями потрібно у кожну таблицю, що відповідає підлеглий сутності, додати набір полів, що є первинним ключем таблиці, відповідної основній сутності. Цей набір полів стає зовнішнім ключем. Оскільки сутність "Виплата" є підлеглою до сутності "Співробітник фірми", у таблицю "Виплати" потрібно додати поле "Табельний номер" із таблиці "Співробітники".

Співробітники	Виплати
Табельний номер	Табельний номер
ПІБ	Дата виплати
Дата народження	Сума
Адреса	
Посада	
Стаж	

Ключові поля позначено напівжирним шрифтом.

Ці таблиці пов'язані відношенням "один до багатьох" за полем "Табельний номер".

Далі потрібно перевірити утворену модель даних за допомогою правил нормалізації. Результатом процесу нормалізації є отримання такої моделі даних, в якій таблиці мають просту і регулярну структуру. В створеній моделі даних можуть бути аномалії – такі ситуації, що призводять до протиріч у базі даних, або ускладнюють опрацювання даних. Нормалізація надає можливість позбутися таких аномалій. Одним з видів аномалій є аномалія модифікації. Вона виникає, коли зміну значень якогось поля потрібно проводити у багатьох записах. Це може призвести до порушення цілісності бази даних (у разі неправильного введення нового значення поля в одному з записів). У розробленій моделі даних така аномалія виникає, коли потрібно змінити назву посади, оскільки таку зміну потрібно виконати для всіх співробітників, які обіймають цю посаду. Для ліквідації цієї аномалії створимо нову таблицю "Штатний розпис", яка міститиме поля "Номер посади" та "Назва посади". Тоді поле "Посада" у таблиці "Співробітники" відповідатиме полю "Номер посади" у таблиці "Штатний розпис" і повинно мати такий самий тип даних.

Штатний розпис
Номер посади
Назва посади

Для формування у студентів компетентностей щодо розробки запитів в конкретній СУБД студентам необхідно запропонувати такі завдання, які б максимально охоплювали всі види запитів: запити на вибірку даних (в тому числі з використанням параметрів), запити на обчислення, в тому числі і групові, запити на модифікацію даних (зміну існуючих даних, додавання нових, вилучення існуючих). Важливо, щоб студенти мали компетентності щодо створення запитів як мовою QBE, так і мовою SQL.

Наведемо кілька прикладів створення запитів до бази даних, про яку йшла мова у задачі 1, в СУБД LibreOffice Base.

Запити на вибірку даних і виконання обчислень

Для задання умов щодо даних числових, грошових типів і типу дата/час застосовуються оператори порівняння $>$, $<$, $>=$, $<=$, $=$ (= не пишеться).

Для задання кон'юнкції ("і") застосовується оператор *And*, для заперечення ("не") – *Not*, для диз'юнкції ("або") – *Or*, складові диз'юнкції можуть також записуватись в різних рядках бланку запиту.

Константи типу дата/час обмежуються позначками #, наприклад #10/10/02#

Для задання умов щодо текстових полів можна застосовувати символи шаблонів * і ?, які мають загальноприйняте значення (відповідно довільна кількість довільних символів і довільний одиниць

символ). Якщо потрібний повний збіг з умовою відбору, відповідний текст записується без символів шаблонів. При вказуванні шаблону слід записати оператор *Like* (подібний), а потім шаблон, взятий в одинарні лапки. Наприклад, умова *Like* '*бухгалтер' є критерієм відбору таких рядків, як, бухгалтер, головний бухгалтер, старший бухгалтер тощо.

Логічна константа ЕМРТУ означає відсутність значення у полі.

В СУБД LibreOffice Base передбачено можливість формувати умови відбору даних в процесі виконання запиту. Така властивість запитів реалізується завдяки використанню параметрів. Щоб створити запит з параметром, слід у бланку запиту в рядку *Критерій* вказати не константу певного типу, з якою будуть порівнюватись дані, а ім'я змінної, перед яким поставити двокрапку. При виконанні такого запиту на екран виводиться вікно з назвою *Введення параметра*, в якому є поле для введення значення параметра. Після введення значення можна побачити відповідь на запит.

Можна вказати у бланку запиту кілька параметрів у різних полях. Тоді при виконанні запиту слід послідовно вказати значення всіх параметрів.

Для створення обчислюваного поля слід у бланку запиту в рядку *Поле* ввести вираз, за яким обчислюються значення у створюваному полі. Він може містити знаки математичних операцій, дужки, стандартні функції LibreOffice Base, імена полів таблиці, до якої створюється запит, взяті в подвійні лапки. Ім'я нового поля слід ввести в рядок бланку запиту *Псевдонім*.

Для обчислення підсумкових значень необхідно включити у бланк запиту рядок *Функція* (для цього слід натиснути кнопку *Функції* на панелі інструментів *Дизайн* вікна програми).

В СУБД LibreOffice Base передбачено виконання таких групових операцій над даними:

Group – операція призначена для групування даних у тому полі, в якому вона встановлена (відповідні записи виводяться підряд);

Сума (Sum) – знаходження суми значень у відповідному полі;

Середнє значення (Average) – знаходження середнього арифметичного (в Конструкторі запитів для виконання цієї операції слід ввести відповідний вираз);

Мінімум (Min) – знаходження мінімального значення у полі;

Максимум (Max) – знаходження максимального значення у полі;

Кількість (Count) – знаходження кількості записів.

Приклад 1. За допомогою мови QBE створити запит для формування даних (ПІБ, Дата виплати, Сума) про розміри всіх виплат співробітнику Петренку, зроблених з початку 2010 року.

Поле	ПІБ	Дата виплати	Сума		
Псевдонім					
Таблиця	Співробітники	Виплати	Виплати		
Сортувати					
Видимий	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Функція					
Критерій	LIKE 'Петренко*'	> #01/01/2010#			
Або					

Рис. 1. Запит до Прикладу 1

Приклад 2. Використовуючи дані таблиць "Співробітники" і "Виплати", обчислити суми, відраховані у фонд соціального страхування (1%) з виплат усім співробітникам за 2009 рік. Вивести у запиті такі відомості: ПІБ, Сума, Податок.

Поле	ПІБ	Дата виплати	Сума	"Сума" * 0.001	Дата виплати
Псевдонім				Податок	
Таблиця	Співробітники	Виплати	Виплати		Виплати
Сортувати					
Видимий	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Функція					
Критерій		>= #01/01/2009#			<= #31/12/2009#
Або					

Рис. 2. Запит до Прикладу 2

Приклад 3. Використовуючи дані таблиць "Співробітники" і "Виплати", обчислити кількість виплат, зроблених кожному співробітнику упродовж другого півріччя 2009 року. У відповіді вивести поля ПІБ, Кількість виплат у 2 півр. 2009 р.

Поле	ПІБ	Сума	Дата виплати	Дата виплати
Псевдонім		Кількість виплат у 2 півр. 2009 р.		
Таблиця	Співробітники	Виплати	Виплати	Виплати
Сортувати				
Видимий	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Функція	Group	Кількість		
Критерій			>= #01/07/2009#	<= #31/12/2009#
Або				

Рис. 3. Запит до Прикладу 3

Запити на модифікацію даних

В СУБД LibreOffice Base передбачено можливість створювати запити, результатами яких є не нові таблиці, а зміни, внесені у вихідні таблиці бази даних. Такі запити створюються за допомогою засобів мови SQL. Після запуску запиту на виконання в LibreOffice Base не виводиться попередження про можливі зміни даних. Виконання запиту призводить до внесення незворотних змін у таблицю. Тому виконувати такі запити слід помірковано і зважено. Після виконання запитів на модифікацію їх результат переглядають, відкривши відповідну таблицю на закладці *Таблиці* вікна бази даних.

При створенні запитів на зміну таблиць слід перейти на закладку *Таблиці* вікна бази даних і вибрати команду меню *Сервіс – SQL*. У допоміжному вікні *Виконати інструкцію SQL* в поле *Виконувана команда* слід ввести текст запиту і натиснути кнопку *Виконати*. Повідомлення про результат виконання команди (успішне виконання або знайдені помилки) виводиться в нижній частині вікна в полі *Стан*. Після завершення роботи з запитом слід натиснути кнопку *Закрити*. Текст SQL-запитів у LibreOffice Base не зберігається. Тому для повторного запуску або збереження запиту в певному звіті текст запиту слід скопіювати до буфера обміну для подальшого використання.

Використовуючи запити на оновлення, можна змінювати значення окремих полів вихідної таблиці. Для полів, що підлягають зміні, можуть бути вказані певні умови.

Найпростіша структура запиту на оновлення:

```
UPDATE "Ім'я таблиці"
SET "Ім'я поля"="нове значення"
WHERE <умова>
```

<умова> – це вираз, в якому можуть використовуватись імена полів, константи, знаки математичних операцій і відношень, логічні оператори.

Текстові константи беруться в одинарні лапки, наприклад: *'текст'*

Константи типу "Дата" записуються у вигляді *{d 'rrrr-мм-дд'}*, наприклад:

```
{d '2010-12-01'} означає 1 грудня 2010 року.
```

За допомогою запитів на вилучення даних користувач може вилучати з таблиці записи, у яких значення певних полів відповідають вказаним умовам.

Найпростіша структура запиту на вилучення:

```
DELETE FROM "Ім'я таблиці"
WHERE <умова>
```

Запити на додавання дозволяють поповнювати таблицю даними з іншої таблиці або вказаними безпосередньо у тексті запиту.

Найпростіша структура запиту на додавання:

```
INSERT INTO "Ім'я таблиці"
VALUES(список значень полів)
```

Приклад 4. Замінити у таблиці "Виплати" всі входження дати 1 травня 2010 року на 11 травня 2010 року.

Виконання. У допоміжному вікні *Виконати інструкцію SQL* в поле *Виконувана команда* ввести такий текст запиту:

```
update "Виплати" set "Дата виплати"={d '2010-05-11'}
where "Дата виплати"={d '2010-05-01'}
```

Приклад 5. Вилучити з таблиці "Виплати" відомості про виплати, зроблені директору.

Виконання. У допоміжному вікні *Виконати інструкцію SQL* в поле *Виконувана команда* ввести такий текст запити:

```
delete from "Виплати" where "Табельний номер" in
(select "Табельний номер" from "Співробітники" where "Посада" in
(select "Номер посади" from "Штатний розпис"
where "Назва посади"='Директор'))
```

Приклад 6. Додати до таблиці "Співробітники" всі анкетні відомості про нового співробітника: Табельний номер – 7777, ПІБ – Ткач О.О., Дата народження – 10.10.1980, Адреса – Проспект Перемоги, 33/22, Посада – 3, Стаж – 10.

Виконання. У допоміжному вікні *Виконати інструкцію SQL* в поле *Виконувана команда* ввести такий текст запити:

```
insert into "Співробітники"
values(7777, 'Ткач О.О.', {d '1980-10-10'}, 'Просп Перемоги, 33/22', 3, 10)
```

Приклад 7. Збільшити на один рік стаж всіх співробітників, які працюють на посаді інженера і чий стаж більше одного року.

Виконання. У допоміжному вікні *Виконати інструкцію SQL* в поле *Виконувана команда* ввести такий текст запити:

```
update "Співробітники" set "Стаж"="Стаж"+1
where "Стаж">1 and "Посада" in
(select "Номер посади" from "Штатний розпис"
where "Назва посади"='Інженер')
```

Приклад 8. Подвоїти всі виплати директору фірми.

Виконання. У допоміжному вікні *Виконати інструкцію SQL* в поле *Виконувана команда* ввести такий текст запити:

```
update "Виплати" set "Сума"="Сума"*2 where "Табельний номер" in
(select "Табельний номер" from "Співробітники"
where "Посада" in
(select "Номер посади" from "Штатний розпис"
where "Назва посади"='Директор'))
```

Приклад 9. Вилучити з таблиці "Співробітники" всі відомості про інженерів, стаж роботи яких більше 20 років.

Виконання. У допоміжному вікні *Виконати інструкцію SQL* в поле *Виконувана команда* ввести такий текст запити:

```
delete from "Співробітники" where "Стаж">20 and "Посада" in
(select "Номер посади" from "Штатний розпис"
where "Посада" in
(select "Номер посади" from "Штатний розпис"
where "Назва посади"='Інженер'))
```

Приклад 10. Вилучити відомості про виплати бухгалтерам, стаж яких більше 10 років.

Виконання. У допоміжному вікні *Виконати інструкцію SQL* в поле *Виконувана команда* ввести такий текст запити:

```
delete from "Виплати" where "Табельний номер" in
(select "Табельний номер" from "Співробітники"
where "Стаж">10 and "Посада" in
(select "Номер посади" from "Штатний розпис"
where "Назва посади"='Бухгалтер'))
```

Приклад 11. Додати до таблиці "Співробітники" окремі анкетні відомості (Табельний номер, ПІБ, Посада) про нового співробітника.

Виконання. У допоміжному вікні *Виконати інструкцію SQL* в поле *Виконувана команда* ввести такий текст запити:

```
insert into "Співробітники" ("Код", "ПІБ", "Посада")
values(8888, 'Шевченко А.А.', 1)
```

Використані джерела

1. Горошко Ю.В. Система інформаційного моделювання у підготовці майбутніх учителів математики та інформатики. – Рукопис. Дисертація на здобуття наукового ступеня доктора педагогічних наук за спеціальністю 13.00.02 – теорія та методика навчання (інформатика) / Юрій Васильович Горошко. – Національний педагогічний університет імені М.П. Драгоманова. – Київ, 2013. – 486 с.
2. Гайна Г.А. Основи проектування баз даних: Навчальний посібник. / Г.А. Гайна. – К. : Кондор, 2008. – 200 с.
3. Дейт К. Введение в системы баз данных. / К. Дейт. – М.-СПб. : Вильямс. – 8-е изд., 2005. – 234 с.
4. Рамський Ю.С. Проектування і опрацювання баз даних. Посібник для вчителів / Ю.С. Рамський, Г.Ю. Цибко. – К, 2003. – 136 с.

Horoshko Y.

**INFORMATION MODELS STUDY WITH DBMS
LIBREOFFICE BASE**

The article deals with the issue of using database management program LibreOffice Base for building data models. LibreOffice Base is a part of free, cross-platform and open source office suite LibreOffice. This is the cause of choosing this database for the presented research.

Features of implementation of SQL data model in the database LibreOffice Base are reviewed. The process of developing of appropriate data models (ER model and SQL model) and implementation of SQL model by means of the database is explored on the base of consideration of practice-oriented problem. The issues concerned with development of both select queries and modification queries in the database LibreOffice Base are researched in details.

LibreOffice Base is a powerful relational database management program. The use of it allows to create databases with complex structure and use SQL language to create various types of queries. It should be noted that select queries can be created also by means of query language QBE, but for creating queries that intended to update, insert and delete data it is necessary to use language SQL. The realization of modification queries in SQL is implemented in a way that minimizes the possibility of their multiple inconsiderate execution.

It is proved that formation of competence of students to develop queries in a specific database it is necessary to offer the problems that cover all types of queries as much as possible. There should be presented such query types as select queries (including parameter queries), calculation queries, queries with aggregate functions, modification queries (changing existing data, add new ones, removal of existing). It is important for students to have competences of creating queries with the help of both language QBE and language SQL.

The article contains a number of examples of queries for the specified database that are designed both by means of language SQL, and graphical interface of language QBE. The problems in creating calculation queries in the 4th version of LibreOffice Base database are reviewed and possible solutions are offered. The article can be used in practice of teacher of information technologies and in the process of self-directed learning of different categories of users who want to effectively work with databases.

Key words: *computer science study, information model, LibreOffice Base, SQL.*

Стаття надійшла до редакції 21.01.2015