

звітів. Дані операції виконуються або з використанням команд запрограмованих для відповідних дій, або – в розширеному режимі – з використанням SQL-послідовностей.

Відібрану інформацію можна вивести на друк, або експортувати в деякі, широко використовуємі типи файлів.

Для створення більш широких можливостей для аналізу звітів та вибірок, передачі інформації до інших установ, вихідна інформація може бути сформована у вигляді файлів типу "Ексель", щоб дозволити використати різноманітні офісні пакети.

#### Список літератури

1. *Методичні рекомендації до виконання курсового проекту з навчальної дисципліни "Організація баз даних та знань" для студентів напряму підготовки 6.050101 "Комп'ютерні науки" всіх форм навчання / укл. Л. А. Павленко, О. В. Тарасов, М. Ю. Лосєв та ін. – Х. : Вид. ХНЕУ, 2013. – 84 с. (Укр. мов.)*
2. *Медична інформаційна система [Електронний ресурс] Режим доступу: [https://uk.wikipedia.org/wiki/Медична\\_інформаційна\\_система](https://uk.wikipedia.org/wiki/Медична_інформаційна_система), вільний.*
3. *Семенюк, А. М., Ніколюк, П. К. (2022). Розробка АРМ баз даних медичного реєстратора. ПРИКЛАДНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, Матеріали III Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених, Вінниця, 89-91. doi: 2708-5821*

**УДК 004.6**

*Дорофєєв Є.О., студент 1 курсу  
спеціальності 122 «Комп'ютерні науки»  
Гончар В. М., асистент  
кафедри інформаційних технологій*

## **СТВОРЕННЯ ТА ВИКОРИСТАННЯ ДАМПІВ БАЗ ДАНИХ, ДЛЯ ЗАБЕЗПЕЧЕННЯ СТАБІЛЬНОСТІ ВИКОРИСТАННЯ БАЗ ДАНИХ**

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Дамп бази даних це текстовий файл, що містить набір операторів SQL, які необхідно виконати на сервері SQL для створення бази даних з усією внутрішньою структурою та заповнення її значеннями. По суті це тип резервного копіювання, який створюється на «логічному» рівні і може використовуватися для відновлення вмісту бази даних після події втрати даних.

На відміну від "фізичного" резервного копіювання бінарних файлів СУБД (які використовуються виключно сервером і не можуть бути скопійовані без його зупинки), дамп бази даних створюється через інтерфейси, що надаються базою даних [1].

Як правило, дамп бази даних можна використовувати для кількох цілей:

- Резервне копіювання – для відновлення даних у разі аварії
- Реплікація – для налаштування підлеглих серверів та підвищення продуктивності та стабільності додатків на основі баз даних.

- Міграція – щоб перевірити, наскільки складно перейти на іншу систему управління базами даних.

- Експерименти - для зміни дампа та тестування різних конфігурацій Дамп бази даних проти холодного резервного копіювання

Гаряче резервне копіювання. У комерційних середовищах кожна хвилина простою сервера може призвести до прямих і непрямих втрат у тисячі доларів. Технологія гарячого резервного копіювання має на увазі, що резервне копіювання може здійснюватися в режимі реального часу без припинення роботи сервера.

Легко редагується. Дампи бази даних мають дуже чіткий формат, який легко читати та редагувати. Кожен файл починається з CREATE TABLE і продовжується рядом операторів INSERT, тобто містить все необхідне відновлення бази даних до нової установки СУБД. Простота формату дозволяє багато різних сценаріїв використання. Наприклад, якщо ви використовуєте MySQL, ви можете змінити механізм зберігання бази даних на MyISAM, відновити його на іншому сервері і таким чином підготувати все для конфігурації реплікації Master-to-Slave. Дізнайтесь більше про реплікацію MySQL.

Може використовуватись для міграції. Усі реляційні бази даних більш-менш відповідають стандарту SQL. Дампи бази даних містять оператори SQL, які повинні прийматись іншими системами управління базами даних без помилок. Загальні поради щодо міграції з MySQL на інші СУБД ви знайдете на сторінці Відновлення MySQL [2].

Важливість резервного копіювання даних

Компанії, які сподіваються на безперебійну роботу, потребують початкових копій своїх даних у різні моменти часу. Без стратегії резервного копіювання їх нічим захистити у разі аварії. Легкість, з якою дані можуть бути втрачені назавжди, занадто велика, щоб з нею впоратися, оскільки дані можуть бути пошкоджені або втрачені з часом. Зловмисні наміри та стихійні лиха не є обов'язковою умовою для реалізації найгіршого сценарію.

Резервне копіювання через певні проміжки часу дає компанії можливість перемотувати годинник тому, перезавантажуючи попередню базу даних. Якщо щось ламається або виходить із ладу, це діє як рятувальний круг для системи. Компанія також має доступ до версій даних. Для повернення доступні різні версії бази даних та продукту. Критичні зміни, які призведуть до поломки системи, можна буде скасувати, після чого ви зможете без проблем відновити старі версії.

Завдяки резервному копіюванню всього міграція на нові сервери або середовища розробки відбувається без побоювання, що дані будуть втрачені.

Mysqldump – корисний інструмент для резервного копіювання баз даних із мінімальною кількістю команд. Одна команда дає змогу виплеснути всю базу даних в один текстовий файл. Інструмент досить універсальний, щоб створювати резервні копії необхідних частин бази даних, і поставляється з безліччю опцій зміни даних, які необхідно зберегти [3].

Якщо автоматизувати цей процес, то можна не турбуватися про збереження ваших проєктів — бекапи завжди будуть доступними у разі

екстрених ситуацій. Вони не тільки рятують у випадку, коли потрібно повернути інформацію, а також використовуються, якщо є необхідність перенести базу даних на інший сервер і т.д. При зміні хостинг-провайдера завжди зручно користуватись дампом. У phpMyAdmin також присутня функція експорту (викачування собі бази даних), але дуже часто там є ліміти на розмір файлів, що завантажуються/вивантажуються, і це не підходить для великих БД. Також через командний рядок виконувати копіювання зручніше, тому що можна зробити копії декількох або всіх баз відразу, що сильно заощадить час, на подальше відновлення.

Програма `mysqldump` створює сценарій відновлення даних. Тобто, висновок `mysqldump` – це не якісь абстрактні та нечитані двійкові дані, а осмислений текст сценарію.

#### Список літератури

1. Бьюли, А. *Изучаем SQL* / А. Бьюли. - М.: Символ-плюс, 2014.
2. Kalen, Delaney *Inside Microsoft® SQL Server(TM) 2005: Query Tuning and Optimization* / Kalen Delaney и др. - М.: Microsoft Press
3. Бен, Форта *SQL за 10 минут* / Форта Бен. - М.: Диалектика / Вильямс

**УДК 004.6**

*Кравчук Р. Ю., студент I курсу  
спеціальності 122 «Комп'ютерні  
науки»*

*Гончар В.М., асистент  
кафедри інформаційних технологій*

## **ЕВОЛЮЦІЯ ОБ'ЄКТНО-ОРІНТОВАНИХ БАЗ ДАНИХ ВІД ПЕРШИХ JSON ДО ПОВНОЦІННИХ ОБ'ЄКТНО-ОРІНТОВАНИХ СУБД**

*Донецький національний університет імені Василя Стуса, м. Вінниця*

В наші часи, все більше програмістів почали використовувати об'єктно-орієнтовані бази даних. Для початку потрібно чітко визначити, що називають об'єктно-орієнтованою базою даних. Об'єктно-орієнтована база даних – це база даних, в якій дані моделюються у вигляді класів їхніх атрибутів і методів.

Переваги такого способу збереження даних:

1. Якщо у вас є багато різних даних, то об'єктно-орієнтована база даних зможе швидше знайти потрібну вам інформацію.
2. Не має повільних з'єднань, як у RDBMS.
3. Оскільки структура бази даних близька до об'єктів програмування, код є простішим і легшим.

Бази даних можна поділити на 2 типи: SQL та NoSQL, базова різниця яких формат збереження даних. База даних типу SQL зберігають дані у вигляді