

**5 ТЕОРІЯ І ПРАКТИКА АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ**

4. Фрадкіна Р. Н. Говорящие руки: Тематический словарь жестового языка глухих. – М.: Знание, 2001. – 62 с.;
5. Cross modal comprehension in ZARDOZ an English to sign-language translation system – [Electronic resource]. URL: <http://dl.acm.org/citation.cfm?id=1641450>;
6. A Machine Translation System from English to American Sign Language - [Electronic resource]. URL: <http://dl.acm.org/citation.cfm?id=749243>;
7. VisiCAST Milestone: Final Report – [Electronic resource]. URL: [http://www.visicast.co.uk/members/milestones/milestones\\_list.htm](http://www.visicast.co.uk/members/milestones/milestones_list.htm);
8. Язык программирования С#. Классика Computers Science. 4-е издание / Хейлсберг А., Торгерсен М., Вилтамут С., Голд П. – К.: «Питер», 2012. – 784 с;
9. Стиллмен Э. Изучаем С#. 2-е издание / Э. Стиллмен, Дж Грин. – М.: «Питер», – 2012. — 704 с;
10. Албахари Джозеф. С# 5.0. Справочник. Полное описание языка / Джозеф Албахари, Бен Албахари. – М.: «Вильямс», – 2013. – 1008 с.;
11. Скит Джон. С#: программирование для профессионалов, 2-е издание / Джон Скит. – М.: «Вильямс», 2011. – 544 с;
12. Роберт Шелдон, Джоффри Мойе. MySQL 5: базовый курс = Beginning MySQL. – М.: «Диалектика», 2007. – 880 с;
13. Гринфилд Джек. Фабрики разработки программ (Software Factories): потоковая сборка типовых приложений, моделирование, структуры / Джек Гринфилд. – «Диалектика», 2006.– 592 с;
14. Буч Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, А.Джекобсон. – М.: ДМК, 2000. – 432 с.;
15. Глотова Т.В. Объектно-ориентированная методология разработки сложных систем. Учебное пособие / Т.В. Глотова. – Х.: Пенза, 2001. – 49 с.

УДК 004.652.4

## АНАЛІЗ ШВИДКОДІ РОЗПОДІЛЕНИХ СИСТЕМ УПРАВЛІННЯ БАЗАМИ ДАНИХ

Юхимчук М. С., Осіпенко Г. А., Ковтун В. В., Мазур О. В.

Copyright © 2014 by author and the journal “Automation technological and business - processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>ONAF  
Open Access

DOI: 10.15673/

### Анотація

В роботі проведено огляд найуживаніших сучасних систем управління базами даних та проведено їх порівняльний аналіз за критерієм швидкодія, для керованої ними бази даних на створеній множині запитів. Результати дозволили зробити висновки, щодо ефективності досліджуваних СУБД.

### Abstract

In article the review of the most used modern database management systems is carried out and they are carried out the comparative analysis behind criterion speed, for the database operated by them on the created plural of inquiries. Results allowed to draw conclusions, concerning efficiency of the studied DBMS.



## 5 ТЕОРІЯ І ПРАКТИКА АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

### Ключові слова

СУБД, розподілені СУБД, реляційна модель БД, аналіз швидкодії.

### Постановка основного завдання СУБД

Основне завдання систем управління базами даних (СУБД) в розподіленій системі полягає в забезпеченні засобів інтеграції локальних баз даних (БД), що розташовані в деяких вузлах обчислювальної мережі, з тим, щоб користувач, який працює в будь-якому вузлі мережі, мав доступ до всіх цих баз даних як до єдиної бази даних. При цьому повинні забезпечуватися: простота використання системи; розвинені засоби забезпечення цілісності; можливості автономного функціонування при порушеннях зв'язності мережі або при адміністративних потребах; висока ефективність [1].

### Аналіз структури розподілених СУБД

На сьогоднішній день структура СУБД має такий вигляд:

1. Процесор опису та підтримки структури БД або ядро, реалізує модель даних і реалізується засобами мови SQL.
2. Інтерфейс введення даних – аналог паперових документів (реалізує вхідну інформаційну мову).
3. Інтерфейс запитів – забезпечує типові запити, що відображають потреби користувачів.
4. Процесор запиту – інтерпретує запити до термінів мови маніпулювання даними.
5. Монітор транзакція – організовує спільне виконання транзакцій над загальними даними.
6. Інтерфейс видачі – отримує від процесора запити – результати, і переводить їх в зручну для людини форму [2].

Розподілена БД – це набір таблиць для реляційної БД, що зберігається в різних вузлах інформаційної мережі, та з точки зору користувача і прикладного програміста складає єдину логічну схему даних.

Більшість розподілених СУБД базується на реляційній моделі даних і розраховані на використання в локальних обчислювальних мережах. Реляційна модель вимагає від сервера СУБД високого рівня складності. У ній робиться спроба позбавити програміста від виконання рутинних операцій по керуванню даними [3].

У реляційній моделі база даних являє собою централізоване сховище таблиць, що забезпечує безпечний одночасний доступ до інформації з боку багатьох користувачів. У рядках таблиць частина полів містить дані, що стосуються безпосередньо запису, а частина – посилання на записи інших таблиць. Таким чином, зв'язки між записами є невід'ємною властивістю реляційної моделі [4].

В розподілених СУБД застосовується мова SQL, що дозволяє формулювати довільні, нерегламентовані запити. Це мова четвертого покоління, тому будь-який користувач може швидко навчитися складати запити. До того ж, існує безліч додатків, що дозволяють будувати логічні схеми запитів у графічному вигляді [5].

Системам управління реляційними базами даних властива відмінність у реалізації мови SQL, хоча це не є суттєвою проблемою реляційної моделі. Кожна розподілена СУБД реалізує якусь підмножину стандарту SQL плюс набір унікальних команд, які розширюють можливості управління даними, проте, ускладнює завдання програмістам, які намагаються перейти від однієї СУБД до іншої [6].

**Мета статті:** розглянувши структуру розподілених СУБД та особливості реляційної моделі даних, ми взяли за мету проаналізувати швидкість виконання запитів у різних розподілених СУБД.

### Матеріали аналізу

В рамках аналізу були обрані СУБД таких версій: OracleDatabase 11g Express Edition, Firebird-2.5.2., 5.5.25 – MySQLCommunity Server, Microsoft SQL Server 2012 Express Edition.

За основу було взято структуру БД «Заробітна плата» з навчального посібника [7]. На рис. 1 представлена спрощена структура БД, що дозволяє вести облік працівників, які працюють на підприємстві, а також зберігати всі видані заробітні плати з їх розшифровкою по дохідних і видаткових статтях.

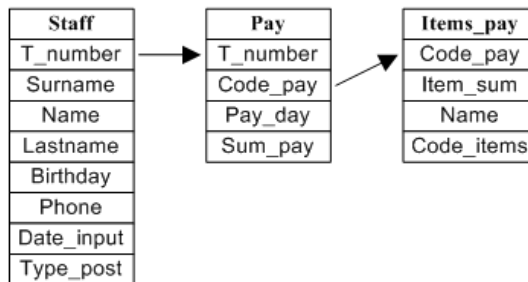
**5 ТЕОРІЯ І ПРАКТИКА АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ**

Рис.1. Спрощена структура БД «Заробітна палата»

Зв'язок між таблицями здійснюється за допомогою таких пар полів:

1. Staff.T\_number – Pay.T\_number;
2. Pay.Code\_pay – Items\_pay.Code\_pay.

Для обраної БД було створено такі запити:

Запит 1. Вибір деяких полів з двох (трьох) таблиць.

Вивести список прізвищ співробітників, значення їх заробітних плат та дати отримання, їх табельні номери та посади, сортуючи отримані результати за датою отримання зарплати та табельним номером співробітника:

– в FirebirdSQL, MySQL, MS SQL Server:

```
– SELECT Surname, Staff.T_number, Post, Type_post, Sum_pay, Pay_day, Item_pay, Item_sum FROM Staff, Pay, Items_pay WHERE (Staff.T_number = Pay.T_number) AND (Pay.Code_pay = Items_pay.Code_pay) ORDER BY Pay_day DESC, Staff.T_number
```

– в Oracle:

```
– SELECT Surname, Staff.T_number, Post, Type_post, Sum_pay, Pay_day, Item_pay, Item_sum FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay, ADMIN_PAY.Items_pay WHERE (Staff.T_number = Pay.T_number) AND (Pay.Code_pay = Items_pay.Code_pay) ORDER BY Pay_day DESC, Staff.T_number
```

Запит 2. Вибір записів по діапазону значень (Between).

Вивести список співробітників і розміри отриманих зарплат за період з 01.01.2000 по 01.01.2014, сортуючи отримані результати за датою отримання зарплати та прізвищем співробітника:

– в FirebirdSQL, MySQL:

```
– SELECT Surname, Name, Lastname, Sum_pay, Pay_day FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day BETWEEN '2000- 01- 01' AND '2014- 01- 01' ORDER BY Pay_day, Surname
```

– в MS SQL Server:

```
– SELECT Surname, Name, Lastname, Sum_pay, Pay_day FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day BETWEEN '01- JAN- 2000' AND '01- JAN- 2014' ORDER BY Pay_day, Surname
```

– в Oracle:

```
– SELECT Surname, Name, Lastname, Sum_pay, Pay_day FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day BETWEEN '01- JAN- 2000' AND '01- JAN- 2014' ORDER BY Pay_day, Surname
```

Запит 3. Багатотабличні запити (вибірка з двох таблиць з використанням JOIN).

Вивести унікальний список всіх співробітників, у яких розмір заробітної плати становив від 1000 до 20000 грн., сортуючи отримані результати за сумою зарплати та прізвищем співробітника:

– в FirebirdSQL, MySQL, MS SQL Server:

```
– SELECT DISTINCT Surname, Name, Lastname, Sum_pay FROM Staff INNER JOIN Pay ON Staff.T_number = Pay.T_number WHERE (Sum_pay >= 1000) AND (Sum_pay < 20000) ORDER BY Sum_pay DESC, Surname
```

– в Oracle:

**5 ТЕОРІЯ І ПРАКТИКА АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ**

- SELECT DISTINCT Surname, Name, Lastname, Sum\_pay FROM ADMIN\_PAY.Staff INNER JOIN Pay ON Staff.T\_number = Pay.T\_number WHERE (Sum\_pay >= 1000) AND (Sum\_pay < 20000) ORDER BY Sum\_pay DESC, Surname

Запит 4. Обчислення підсумкових значень з використанням агрегатних функцій.

Вивести кількість співробітників по кожній посаді, на якій працює менше 5 людей:

- в FirebirdSQL, MySQL, MS SQL Server:
  - SELECT Post, COUNT(T\_number) FROM Staff GROUP BY Post HAVING COUNT(T\_number) < 5
- в Oracle:
  - SELECT Post, COUNT(T\_number) FROM ADMIN\_PAY.Staff GROUP BY Post HAVING COUNT(T\_number) < 5

Запит 5. Використання змінних замість назв таблиць.

Вивести список співробітників, табельний номер і сумарну зарплату кожного, групуючи отримані результати за табельним номером співробітника:

- в FirebirdSQL, MySQL, MS SQL Server:
  - SELECT Name, Lastname, Surname, d.T\_number, SUM(Sum\_pay) FROM Staff d, Pay f WHERE (d.T\_number = f.T\_number) GROUP BY d.T\_number
- в Oracle:
  - SELECT Name, Lastname, Surname, d.T\_number, SUM(Sum\_pay) FROM ADMIN\_PAY.Staff d, ADMIN\_PAY.Pay f WHERE (d.T\_number = f.T\_number) GROUP BY d.T\_number

Запит 6. Використання спільно з підзапитом квантора існування.

Вивести унікальний список співробітників, які отримували премію, сортуючи отримані результати за прізвищем співробітника:

- в FirebirdSQL, MySQL, MS SQL Server:
  - SELECT DISTINCT Surname, Name, Lastname FROM Staff, Pay WHERE Staff.T\_number = Pay.T\_number AND EXISTS (SELECT \* FROM Items\_pay WHERE Items\_pay.Code\_pay = Pay.Code\_pay AND Item\_pay = 'премия') ORDER BY Surname
- в Oracle:
  - SELECT DISTINCT Surname, Name, Lastname FROM ADMIN\_PAY.Staff, ADMIN\_PAY.Pay WHERE Staff.T\_number = Pay.T\_number AND EXISTS (SELECT \* FROM ADMIN\_PAY.Items\_pay WHERE Items\_pay.Code\_pay = Pay.Code\_pay AND Item\_pay = 'премия') ORDER BY Surname

Запит 7. Використання спільно з підзапитом квантора існування.

Вивести список співробітників, які жодного разу не отримували зарплати, сортуючи отримані результати за табельним номером співробітника:

- в FirebirdSQL, MySQL, MS SQL Server:
  - SELECT Staff.T\_number, Surname, Name, Lastname FROM Staff WHERE NOT EXISTS (SELECT \* FROM Pay WHERE Staff.T\_number = Pay.T\_number) ORDER BY Staff.T\_number
- в Oracle:
  - SELECT Staff.T\_number, Surname, Name, Lastname FROM ADMIN\_PAY.Staff WHERE NOT EXISTS (SELECT \* FROM ADMIN\_PAY.Pay WHERE Staff.T\_number = Pay.T\_number) ORDER BY Staff.T\_number

Запит 8. Використання функцій спільно з підзапитом з використанням агрегатних функцій.

Вивести список співробітників і дати з розмірами отриманих зарплат, які перевищили середній розмір їх же зарплат:

- в FirebirdSQL, MySQL, MS SQL Server:
  - SELECT Surname, Name, Lastname, Sum\_pay, Pay\_Day FROM Staff INNER JOIN PAY ON Staff.T\_number = Pay.T\_number WHERE Pay.Sum\_pay > (SELECT AVG(Sum\_pay) FROM Pay)
- в Oracle:

**5 ТЕОРИЯ І ПРАКТИКА АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ**

– SELECT Surname, Name, Lastname, Sum\_pay, Pay\_Day FROM ADMIN\_PAY.Staff INNER JOIN PAY ON Staff.T\_number = Pay.T\_number WHERE Pay.Sum\_pay > (SELECT AVG(Sum\_pay) FROM ADMIN\_PAY.Pay)

**Результати аналізу та висновки**

Отримані результати дозволили зробити висновок, що СУБД Oracle має максимальну швидкість та мінімальні затрати часу для виконання сформованої множини запитів (табл. 1). Це досягається завдяки властивості Oracle кешувати результати, що істотно прискорює процес повторної обробки запитів і викликів функцій [8].

Таблиця 1  
Результати аналізу

Назва СУБД	Час виконання запиту, мс							
	1	2	3	4	5	6	7	8
Oracle	3,9	10,1	4,0	9,4	4,9	8,0	4,5	3,0
FirebirdSQL	4,3	10,0	5,2	10,0	5,1	7,9	5,1	3,0
MySQL	4,8	13,6	5,6	12,6	5,3	8,4	5,7	3,5
Microsoft SQL Server	4,4	13,8	5,5	12,8	5,3	8,4	5,5	3,6

СУБД Firebird SQL 2.5 показала другу після Oracle швидкість завдяки паралельній обробці запитів. Firebird SQL також працює з великою кількістю користувачів, тому узагальненою ефективністю мало поступається цьому показнику для СУБД Oracle.

MySQL та Microsoft SQL Server менш швидкодіючі від наведених вище СУБД. Проте СУБД MS SQL Server перевершує MySQL як платформу даних у всіх істотних для бізнесу областях. Вона забезпечує доступ до великих ресурсів, провідну в галузі продуктивність і масштабованість корпоративного класу, найвищий рівень безпеки та доступності, комплексну платформу бізнес-аналітики. Незважаючи на достатньо великий перелік переваг, є також суттєвий мінус – те, що продукти компанії Microsoft платні, це може створити істотну заваду для користувача при виборі СУБД [9].

Все ж таки, даний аналіз проводився на основі структури БД саме невеликої компанії, і час виконання запитів варіюється у десятих частинах мілісекунд, що є непомітним, для користувача, показником. Проте, обираючи СУБД для більш масштабних проектів, все ще потрібно приділяти увагу таким критеріям як: продуктивність, надійність, вимоги до робочого середовища, особливості розробки додатків та ін.

**Література**

1. Пасічник В.В. Організація баз даних та знань / В.В. Пасічник, В.А.Резніченко – К.: Видавнича група ВНУ, – 2006. – 384 с.: іл. ISBN 966– 552– 156– X;
2. Ісаченко А. М. Модели данных и системы управления базами данных: пособие для студ. – Минск: БГУ, 2007. – 220 с;
3. Дейт, К. Дж. Введение в системы баз данных, 8– е издание: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 1328 с.: ил. – Парал. тит. англ. ISBN 5– 8459– 0788– 8 (рус.);
4. Пушников А.Ю. Введение в системы управления базами данных. Часть 1. Реляционная модель данных: Учебное пособие/Изд– е Башкирского ун– та. – Уфа, 1999. – 108 с. – ISBN 5– 7477– 0350– 1;
5. Грабер М. Введение в SQL. – М.: Лори, 2010. – 228 с. ISBN 5– 85582– 010– 6;
6. Класифікація СКБД [Електронний ресурс]/ SQLInfo/ Режим доступу: [http://www.sqlinfo.org.ua/classification\\_subd.php](http://www.sqlinfo.org.ua/classification_subd.php);
7. Семенова І. I.SQL стандарт в СУБД MS SQL SERVER, ORACLE, VFP И ACCESS: манипулирование данными. – Омск: Изд– во СибАДИ, 2008. – 57 с. – ISBN: 978– 5– 93204– 422– 3;
8. Oracle Corporation. Информационный документ: Надежный доступ к информации СУБД OracleDatabase 11g: Обзор новых возможностей, Oracle Corporation, 2007 – 12 с;
9. Сравнение SQLServer с MySQL [Електронний ресурс]/ MicrosoftSQLServer 2008/ Режим доступу: <http://www.microsoft.com/sqlserver/2008/ru/compare-mysql.aspx>.