

¹ Доцент кафедри «Програмне забезпечення комп'ютерних систем», Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37, м. Київ, 03056, Україна. ORCID: <https://orcid.org/0000-0003-1541-1681>

* Автор, відповідальний за листування: inna.saiapina@gmail.com

ОПТИМІЗАЦІЯ РОБОТИ РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА ТРАНСПОРТІ

Інформатизація суспільства ставить нові виклики та вимоги перед інформаційними системами на транспорті, що відповідають за обробку та зберігання даних. Значна кількість проблем продуктивності розроблених додатків пов'язана з базами даних, бо часто низька продуктивність бази даних безпосередньо впливає на уповільнення роботи програми. Для нормального функціонування процесів діагностики рухомого складу та систем автоматизації, прийняття рішень, формування звітів у транспортній галузі постає задача оптимізації управління базами даних та забезпечення їх продуктивності за умови одночасного багатокористувацького доступу. У статті розглянуті етапи виконання запиту та підходи щодо підвищення ефективності роботи оптимізатора запитів. Проаналізовані основні фактори, від яких залежить продуктивність виконання запитів у інформаційних системах на основі реляційних баз даних. До критеріїв оптимізації можна віднести такі оцінки продуктивності, як час, необхідний на генерацію звіту, час виконання запитів, швидкість знаходження даних у полях без індексування, максимальна кількість одночасних звернень до даних при багатокористувацькому режимі, швидкість індексування, а також операції оновлення, видалення та додавання.

Ключові слова: реляційні бази даних, інформаційні системи, оптимізація запитів, системи управління базами даних, SQL.

Вступ. Щороку кількість інформації, яку виробляє людство, невідомо зростає. Це ставить нові виклики та вимоги перед системами, що відповідають за обробку та зберігання інформації. Транспортна галузь не є винятком і потребує оптимізації систем зберігання та обробки різних видів даних відповідно до викликів, що постають із ростом інформатизації суспільства. Як приклад можна навести інформаційні системи, що відповідають за продаж білетів, автоматизацію обліку пасажирських перевезень чи вантажів, які функціонують на основі відповідних баз даних.

Аналіз останніх досліджень і постановка проблеми. База даних – це колекція логічно пов'язаних даних та їх описів, призначених для задоволення інформаційних потреб організації (БД). Програмна система, яка дозволяє користувачам визначати, створювати, підтримувати дані та керувати доступом до бази даних, називається системою управління базами даних (СУБД) [1]. Відповідно до даних статистики, станом на 2020 рік 70% ринку займають продукти реляційних СУБД, у той час як на нереляційні СУБД відводиться 30%.

Популярність реляційних СУБД значною мірою викликана їх надійністю та підтримкою принципів ACID. Але слід зазначити, що через те, що вони віддають перевагу підтримці узгодженості, а не доступності та продуктивності, це ускладнює горизонтальне масштабування

таких систем [2]. У результаті були розроблені нові СУБД, де послаблені деякі обмеження щодо узгодженості задля забезпечення кращої масштабованості і продуктивності. Було запроваджено багато нових технологій, включаючи сховища з широким стовпчиком, наприклад Google Bigtable, Apache Cassandra [3], Apache HBase; сховища ключ-значення Amazon DynamoDB [4], LevelDB і RocksDB; документоорієнтовані сховища AsterixDB [5], ArangoDB і MongoDB [6, 7, 8]; колонкоорієнтовані сховища, наприклад, Apache Druid і ClickHouse [9]; графові сховища, наприклад Neo4j.

У роботі [10] наведений приклад проєктування бази даних опрацювання параметрів пасажиропотоків громадського транспорту для вдосконалення транспортної системи міста. У дослідженні [11] автори наводять приклад бази даних міського транспорту Китаю для обробки даних міського транспорту, управління та оцінки ефективності його роботи.

Великі і складні міжнародні проєкти з розробки, проєктування та будування транспортних систем, транспортних засобів та їх частин проходять стадії комп'ютерного моделювання, що ставить необхідність зберігання великих обсягів даних. При цьому застосовуються нові перспективні технології, що вимагають обробки та зберігання об'ємних даних. Наприклад, технологія цифрового двійника, що представляє собою цифрову модель фізичного об'єкта чи процесу та динамічно відображає його характеристики протягом певного обсягу часу [12].

У роботах [13, 14] наведено, як створення бази даних параметрів засобів транспорту та проведений на її основі інтелектуальний аналіз дозволяє вдосконалити характеристики роботи транспортної системи.

У статті [15] наводиться аналіз, що 71% проблем продуктивності додатків пов'язані з базами даних, бо часто низька продуктивність бази даних безпосередньо впливає на уповільнення роботи програми. Оптимізацію баз даних пропонується розділити на дві основні категорії: оптимізацію на основі фізичної реалізації та оптимізацію на основі параметрів конфігурації.

Мета і завдання дослідження. Метою дослідження є аналіз засобів підвищення продуктивності роботи реляційних баз даних та способів оптимізації запитів.

Матеріали та методи дослідження. Основна взаємодія користувача з реляційними СУБД відбувається за допомогою SQL (Structured Query Language) – мови структурованих запитів. Запит – це мовний вираз, що описує дані, які підлягають вибірці із бази даних. Процес виконання запитів складається з декількох етапів (рис. 1).

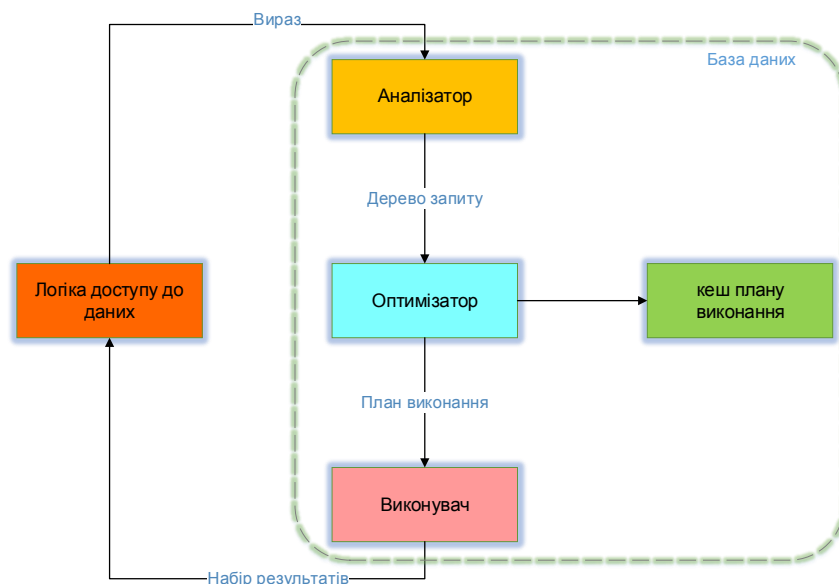


Рис. 1. Схема виконання запиту SQL

При виконанні запиту, СУБД повинна виконати такі кроки:

1. СУБД аналізує (синтаксично перевіряє) код SQL запиту. Далі зв'язує таблиці та стовпці, до яких звертається запит, у попередню конструкцію (дерево) перед виконанням.

2. Зв'язання таблиць та стовпців у дерево. Ця зв'язана конструкція передається компілятору, який перетворює SQL-код на виконавчий машинний код.

3. Створюється виконавчий файл, що передається оптимізатору. Код ще повинен пройти через оптимізатор. Оптимізатор – це шар програмного забезпечення у складі СУБД. Він аналізує, що у результаті запиту потрібно отримати, та розраховує оптимальний план виконання.

4. Обчислення оптимізатором плану виконання. При цьому оптимізатор аналізує, які таблиці та колонки потрібні та як їх найефективніше дістати. Оптимізатор призначений для пошуку найшвидшого та найефективнішого вирішення виконання запиту.

5. СУБД реалізує план виконання.

Завдання оптимізації полягає у забезпеченні максимальної пропускної спроможності при заданій кількості ресурсів, або забезпеченні мінімального споживання ресурсів при цій пропускній спроможності. Вартість, що підлягає мінімізації, складається з таких складових: час використання ресурсів процесора, вартість використання центрального процесора та доступу до вторинної пам'яті, вартість зберігання та вартість комунікацій, що включає у себе вартість затримок у обробці. До критеріїв оптимізації можна віднести такі оцінки продуктивності, як час, необхідний на генерацію звіту, час виконання запитів, швидкість знаходження даних у полях без індексування, максимальна кількість одночасних звернень до даних при багатокористувацькому режимі, швидкість індексування, а також операцій оновлення, видалення та додавання.

Слід зазначити, що у централізованих БД переважає вартісний компонент часу доступу до вторинної пам'яті та використання центрального процесора у разі складних запитів. Для географічно розподілених СУБД зростає вартісний компонент комунікаційних затримок. У локально-розподілених СУБД всі компоненти мають рівний вплив, що ускладнює розрахування оціночної функції при оптимізації.

Ключовим завданням оптимізації запитів є пошук оптимального порядку виконання операцій з'єднання на основі витрат. Через існування великої кількості можливих дерев об'єднання для певного запиту, витрати на алгоритм перерахування об'єднань для кожного дійсного дерева об'єднань мають бути мінімальними.

На продуктивність виконання запитів значною мірою впливає нестача допоміжних індексів. У цьому випадку СУБД доводиться проводити повне сканування таблиці, що вимагає додаткового процесорного часу, тому є менш ефективним. Ще однією проблемою є неузгодженість типів даних, що об'єднуються. У цьому випадку оптимізатор втрачає можливість ефективно приєднувати стовпці та таблиці через необхідність попереднього перетворення всіх значень.

Оптимізатор обчислює вартості запропонованих планів, генерує їх та обирає план з найменшою вартістю, тобто який найшвидше виконуватиметься при використанні найменшої кількості ресурсів.

Авторами роботи [16] експериментально досліджено, що оптимізація запитів є важливою для ефективної для обробки запитів і що із збільшенням кількості приєднань (JOIN) помилки оцінювання вартості планів виконання запитів у системах реляційних баз даних швидко зростають.

Також слід зазначити, що у виконанні запиту беруть участь різні структури пам'яті комп'ютера, які мають дуже різну швидкість звернення процесора до даних (рис. 2). Для порівняння відносні швидкості звернення до регістрів процесору вимірюються у наносекундах, до пам'яті – у мікросекундах, а до диску – у мілісекундах. Тобто, сервер може виконати мільйон циклів процесора за час, потрібний для виконання одного дискового введення-виведення.

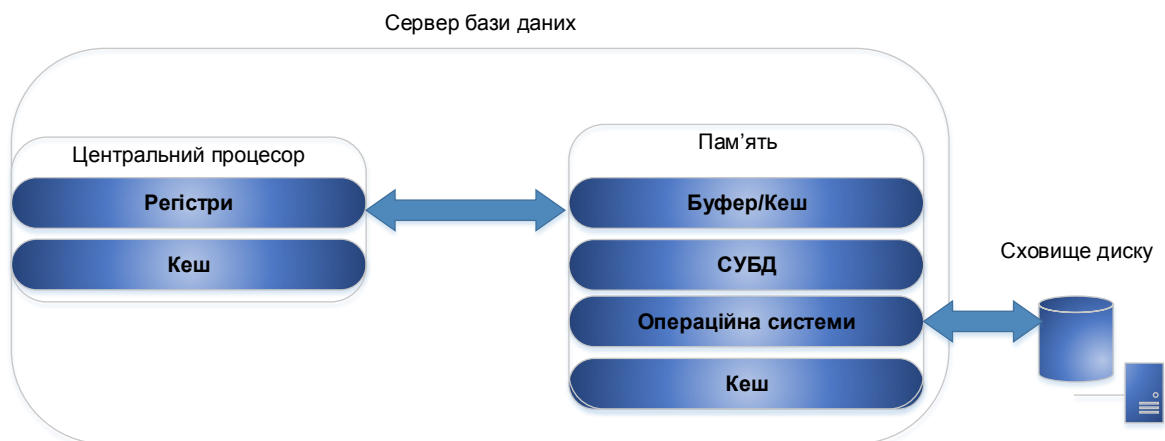


Рис. 2. Зберігання даних у інформаційній системі

За результатами проведеного аналізу для оптимізації виконання запитів БД пропонуються такі рекомендації:

1. Мінімізація процесів дискового введення-виведення для підвищення швидкості виконання запитів;
2. Створення індексації параметрів, що підлягають частим запитам;
3. Зазначення у запиті конкретних полів SELECT, замість SELECT * для уникнення залучення додаткових ресурсів БД;
4. Для фільтрації даних надання переваги оператору WHERE замість HAVING, який виконується пізніше відповідно до порядку виконання та не використовує переваги індексації у разі їх наявності, як це робить WHERE;
5. Обмеження необхідної кількості рядків результатів запиту за допомогою LIMIT, що дозволить зменшити обчислювальну потужність, необхідну для виконання;
6. Заміна підзапитів об'єднанням, що потребує менше процесорного часу на виконання;
7. Оптимізація порядку прямуювання операцій JOIN при мультиоб'єднаннях, наприклад, завдяки застосуванню існуючих бенчмарків [16].

Висновки. У зв'язку з суттєвим збільшенням інформації все більш актуальним стає питання зберігання та обробки великих масивів даних. Для нормального функціонування процесів діагностики рухомого складу та систем автоматизації, прийняття рішень, формування звітів у транспортній галузі постає задача оптимізації управління базами даних та забезпечення їх продуктивності за умови одночасного багатокористувацького доступу. У статті проведено аналіз основних факторів, що впливають на продуктивність, швидкість та ефективність роботи реляційних баз даних. Розглянуто процес оптимізації виконання запитів, особлива увага при цьому приділена можливості зменшення необхідної обчислювальної потужності. Запропоновані відповідні рекомендації для користувачів інформаційної системи з метою оптимізації та пришвидшення роботи.

ЛІТЕРАТУРА

1. B. Shah, P. M. Jat, and K. Sashidhar. Performance Study of Time Series Databases. arXiv preprint arXiv:2208.13982. 2022. <https://doi.org/10.48550/arXiv.2208.13982>.
2. Mostafa J., Wehbi S., Chilingaryan S., Kopmann A. SciTS: A Benchmark for Time-Series Database in Scientific Experiments and Industrial Internet of Things. 34th International Conference on Scientific and Statistical Database Management (SSDBM 2022), July 6–8, 2022, Copenhagen, Denmark. ACM, New York, NY, USA, 2022. 1-11 pages. <https://doi.org/10.1145/3538712.3538723>.
3. Chebotko A., Kashlev A., Lu S. A big data modeling methodology for Apache Cassandra. 2015 IEEE International Congress on Big Data. IEEE, New York, NY, USA, 2015. P. 238–245. <https://doi.org/10.1109/BigDataCongress.2015.41>.

4. Sivasubramanian S. (2012, May). Amazon dynamoDB: a seamlessly scalable non-relational database service. *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data (Scottsdale, Arizona, USA) (SIGMOD'12)*. Association for Computing Machinery, New York, NY, USA, 2012. P. 729–730. <https://doi.org/10.1145/2213836.2213945>.
5. Alsubaiee S., Altowim Y., Altwaijry H., Behm A., Borkar V., Bu Y., Carey M., Cetindil I., Cheelangi M., Faraaz K., Gabrielova E., Grover R., Heilbron Z., Kim Y.-S., Li C., Li G., Ok J. M., Onose N., Pirzadeh P., Tsotras V., Vernica R., Wen J., Westmann, T. *Proc. VLDB Endow*, 2014. P. 1905–1916. <https://doi.org/10.14778/2733085.2733096>.
6. Chickerur S., Goudar A., & Kinnerkar A. Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications. 2015 8th International Conference on Advanced Software Engineering and Its Applications (ASEA). IEEE, New York, NY, USA, 2015. P. 41–47. <https://doi.org/10.1109/ASEA.2015.19>.
7. Chickerur S., Goudar A., Kinnerkar A. Comparison of relational database with document-oriented database (mongodb) for big data applications. 2015 8th International Conference on Database Theory and Application (DTA). IEEE, New York, NY, USA, 2015. P. 14–17. <https://doi.org/10.1109/DTA.2015.14>.
8. Kang Y. S., Park I. H., Rhee J., Lee Y. H. MongoDBBased Repository Design for IoT-Generated RFID/Sensor Big Data. *IEEE Sensors Journal*, 2016. Vol. 16(2). P. 485–497. <https://doi.org/10.1109/JSEN.2015.2483499>.
9. Imasheva B., Azamat N., Sidelkovskiy A., Sidelkovskaya A. The practice of moving to big data on the case of the nosql database, clickhouse. Springer, Cham./Editors - Hoai An Le Thi, Hoai Minh Le, & Tao Pham Dinh. Springer International Publishing, Cham, 2020. P. 820–828. https://doi.org/10.1007/978-3-030-21803-4_113.
10. Боре́йко О. Ю. Розроблення бази даних для автоматизованої системи опрацювання параметрів пасажиропотоків громадського транспорту. Моделювання та інформаційні технології, 2018. Вип. 82. С. 177-184. Режим доступу: http://nbuv.gov.ua/UJRN/Mtit_2018_82_26.
11. Jiang T., Wu Z., Song Y., Liu X., Liu H., Zhang H. (2013). Sustainable transport data collection and application: china urban transport database. *Mathematical Problems in Engineering*, 2013. Vol. 2013. P. 1-10. <https://doi.org/10.1155/2013/879752>
12. Liu M., Fang S., Dong H., Xu C. Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems*, 2021. Vol 58. P. 346-361. <https://doi.org/10.1016/j.jmsy.2020.06.017>.
13. Саяпіна І.О., Горобченко О.М., Демченко В.О., Штомпель Ю.М. Моделювання та інтелектуальний аналіз параметрів тонального рейкового кола. Збірник наукових праць ДУІТ серія «Транспортні системи і технології». К.: ДУІТ, 2022, Вип. 39. С. 167-174. <https://doi.org/10.32703/2617-9040-2022-39-16>
14. Саяпіна І.О. Застосування методу класифікації даних та нейронних мереж для підвищення завадостійкості рейкового кола. Збірник наукових праць ДУІТ серія «Транспортні системи і технології». К.: ДУІТ, 2022. Вип. 39, С. 266-277. <https://doi.org/10.32703/2617-9040-2022-39-26>
15. Samson S., Aponso A. An Analysis on Automatic Performance Optimization in Database Management Systems. 2020 World Conference on Computing and Communication Technologies (WCCCT), 2020. P. 6-9. <https://doi.org/10.1109/WCCCT49810.2020.9169995>.
16. V Leis, V., Gubichev, A., Mirchev, A., Boncz, P., Kemper, A., & Neumann, T. How good are query optimizers, really? *Proc. VLDB Endow*, 2015, Vol. 9(3). P. 204–215. <https://doi.org/10.14778/2850583.2850594>.

REFERENCES

1. Shah, B., Jat, P. M., & Sashidhar, K. (2022). Performance Study of Time Series Databases. *arXiv preprint arXiv:2208.13982*. <https://doi.org/10.48550/arXiv.2208.13982>.
2. Mostafa, J., Wehbi, S., Chilingaryan, S., & Kopmann, A. (2022, July) SciTS: A Benchmark for Time-Series Databases in Scientific Experiments and Industrial Internet of Things. *34th International Conference on Scientific and Statistical Database Management (SSDBM 2022)*. (6–8 July, 2022). (pp. 1-11). Copenhagen, Denmark. ACM, New York, NY, USA. <https://doi.org/10.1145/3538712.3538723>.
3. Chebotko A., Kashlev A., & Lu, S. (2015, June). A big data modeling methodology for Apache Cassandra. In *2015 IEEE International Congress on Big Data*. (pp. 238-245). IEEE., New York, NY, USA, 2015. <https://doi.org/10.1109/BigDataCongress.2015.41>
4. Sivasubramanian, S. (2012, May). Amazon dynamoDB: a seamlessly scalable non-relational database service. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. (pp. 729-730). <https://doi.org/10.1145/2213836.2213945>.
5. Alsubaiee, S., Altowim, Y., Altwaijry, H., Behm, A., Borkar, V., Bu, Y., Carey, M., Cetindil, I., Cheelangi, M., Faraaz, K., Gabrielova, E., Grover, R., Heilbron, Z., Kim, Y.-S., Li, C., Li, G., Ok, J. M., Onose, N., Pirzadeh, P., Tsotras, V., Vernica, R., Wen, J., & Westmann, T. (2014). AsterixDB: A Scalable, Open Source BDMS. *Proc. VLDB Endow*, 1905–1916. <https://doi.org/10.14778/2733085.2733096>.
6. Chickerur, S., Goudar, A., & Kinnerkar, A. (2015, November). Comparison of relational database with document-oriented database (mongodb) for big data applications. In *2015 8th International Conference on Advanced Software Engineering & Its Applications (ASEA)*. (pp. 41-47). IEEE. <https://doi.org/10.1109/ASEA.2015.19>

7. Jung, M. G., Youn, S. A., Bae, J., & Choi, Y. L. (2015, November). A study on data input and output performance comparison of mongodb and postgresql in the big data environment. In *2015 8th international conference on database theory and application (DTA)*. (pp. 14-17). IEEE. <https://doi.org/10.1109/DTA.2015.14>.
8. Kang, Y. S., Park, I. H., Rhee, J., & Lee, Y. H. (2015). MongoDB-based repository design for IoT-generated RFID/sensor big data. *IEEE Sensors Journal*, 16(2), 485-497. <https://doi.org/10.1109/JSEN.2015.2483499>
9. Imasheva, B., Azamat, N., Sidelkovskiy, A., & Sidelkovskaya, A. (2019, July). The practice of moving to big data on the case of the nosql database, clickhouse. In *World Congress on Global Optimization* (pp. 820-828). Springer, Cham. https://doi.org/10.1007/978-3-030-21803-4_113.
10. Боре́йко О. Ю. (2018) Rozroblennia bazy danykh dlia avtomatyzovanoi systemy opratsiuvannia parametriv pasazhyropotokiv hromadskoho transportu [Modeling and information technologies]. *Modeliuvannia ta informatsiini tekhnologii - Modeling and information technologies*, 82, 177-184. http://nbuv.gov.ua/UJRN/Mtit_2018_82_26.
11. Jiang, T., Wu, Z., Song, Y., Liu, X., Liu, H., & Zhang, H. (2013). Sustainable transport data collection and application: china urban transport database. *Mathematical Problems in Engineering*, 2013, 1-10. <https://doi.org/10.1155/2013/879752>
12. Liu, M., Fang, S., Dong, H., & Xu, C. (2021). Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems*, 58, 346-361. <https://doi.org/10.1016/j.jmsy.2020.06.017>.
13. Саяпіна І.О., Горобченко О.М., Демченко В.О., Штомпель Ю.М. (2022) Modeliuvannia ta intelektualnyi analiz parametriv tonalnoho reikovooho kola [Modeling and intellectual analysis of the parameters of the tonal rail circuit]. // *Zbirnyk naukovykh prats DUIT serii «Transportni systemy i tekhnologii»*. K.: DUIT - Collection of scientific works DUIT series «Transport systems and technologies». K.: DUIT, 2022, 39, 167-174 [in Ukrainian]. <https://doi.org/10.32703/2617-9040-2022-39-16>.
14. Саяпіна І.О. (2022) Zastosuvannia metodu klasyfikatsii danykh ta neironnykh merezh dlia pidvyshchennia zavadostiikosti reikovooho kola [Application of the method of data classification and neural networks to increase the interference resistance of the rail circuit]// *Zbirnyk naukovykh prats DUIT serii «Transportni systemy i tekhnologii»*. K.: DUIT - Collection of scientific works DUIT series «Transport systems and technologies». K.: DUIT, 2022, 39, 266-277 [in Ukrainian]. <https://doi.org/10.32703/2617-9040-2022-39-26>.
15. Samson, S., & Aponso, A. (2020, May). An Analysis on Automatic Performance Optimization in Database Management Systems. In *2020 World Conference on Computing and Communication Technologies (WCCCT)* (pp. 6-9). IEEE. <https://doi.org/10.1109/WCCCT49810.2020.9169995>.
16. Leis, V., Gubichev, A., Mirchev, A., Boncz, P., Kemper, A., & Neumann, T. (2015). How good are query optimizers, really?. *Proceedings of the VLDB Endowment*, 9(3), 204-215. <https://doi.org/10.14778/2850583.2850594>.

Inna Saiapina¹

¹ Associate Professor, Department of Computer Systems Software, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Prosp. Peremohy, Kyiv, Ukraine, 03056, Ukraine. ORCID: <https://orcid.org/0000-0003-1541-1681>

RELATIONAL DATABASE WORK OPTIMIZATION OF THE TRANSPORT INFORMATION SYSTEM

Informatization of society poses new challenges and requirements for information systems in transport, which are responsible for data processing and storage. A significant number of performance problems in developed applications are related to databases, because often poor database performance directly affects the application's slowdown. For the normal functioning of the processes of diagnostics of rolling stock and automation systems, decision-making, and the formation of reports in the transport industry, the task of optimizing the management of databases and ensuring their productivity under the condition of simultaneous multi-user access arises. The article discusses the stages of query execution and approaches to increasing the efficiency of the query optimizer. The main factors that depend on the performance of queries in information systems based on relational databases are analyzed. Optimization criteria can include such performance estimates as time required to generate a report, query execution time, speed of finding data in non-indexed fields, maximum number of simultaneous accesses to data in multi-user mode, speed of indexing, as well as update, delete and add operations.

Keywords: relational databases, information systems, query optimization, database management systems, SQL.