

DOI: [https://doi.org/10.26642/ten-2023-1\(91\)-193-199](https://doi.org/10.26642/ten-2023-1(91)-193-199)
УДК 004.652

**І.І. Сугоняк, к.т.н., доц.
О.В. Коротун, к.пед.н.
А.А. Левицький, здобувач
В.А. Сугоняк, здобувач**

Державний університет «Житомирська політехніка»

Проектування бази даних до гри «Minecraft»

Мета статті – виконати зворотну розробку (англ. reverse engineering) популярної гри Minecraft в аспектах взаємодії ігрових даних та їх зберігання для подальшого використання. На основі отриманих даних спроектувати базу даних і текстову симуляцію названої вище гри для отримання навичок з формування структурованих систем даних для ігрових продуктів, які потребують складної взаємодії даних. У статті висвітлено основні алгоритми формування бази даних та взаємодії між даними так, як це вимагає тип ігрового продукту та його вміст. Продемонстровано чотири етапи розробки програмного продукту. На першому етапі сформовано концепт для проектування бази даних, проаналізовано дані, які потрібно зберігати, та визначено ідею для створення інформаційної системи, як і використовувати засоби проектування програмних продуктів та системи управління базами даних (СУБД). Другий етап описуватиме формування бази даних зі всіма особливостями, представленнями, процедурами та тригерами для комфортної обробки даних при проектуванні програмного продукту, та використання бази даних в інформаційній системі, та роботу з нею при різних діях гравця. Третій – вигляд, структура та робота симуляції гри у вигляді вебдодатка, особливості роботи такої платформи з базою даних та концепт ігрового процесу. Також на цьому етапі висвітлено готовий програмний продукт, який представляє роботу сформованої бази даних та симуляцію гри «Minecraft». В базі даних також важлива безпека доступу до даних, тому на четвертому етапі буде розглянуто реалізацію захисту бази даних. Для роботи з даними використано СУБД Microsoft SQL Server, створено симуляцію гри у вигляді сайту за допомогою мови програмування C#, платформи ASP.NET та фреймворку для роботи з базами даних Entity Framework.

Ключові слова: СУБД; ігровий продукт; структуризація; SQL; C#; ASP.NET; Minecraft.

Актуальність теми. Ігрова індустрія кожного року підтверджує той факт, що вона є однією з передових аспектів у програмній розробці, адже людям цікаво провести час в ігровому процесі різних видів: змагатися з іншими гравцями, перемагати у грі або зануритися в цікаву історію. СУБД стали доцільними і в цьому напрямі, адже великий потік даних від гравців або ігрового процесу зручно зберігати у структурованому та логічно пов'язаному між кожною його частиною місці, тому використання такого способу зберігання даних є актуальним для ігрових розробників.

Аналіз останніх досліджень та публікацій, на які спираються автори. Основні поняття теорії баз даних (БД), процес нормалізації БД, методологію проектування БД, архітектуру БД, сучасні системи керування базами даних (СКБД), мову запитів SQL та питання їх навчання у ЗВО розглядали такі вітчизняні науковці: Г.А. Гайна [2], Л.С. Глоба [9], В.М. Гужва [3], І.О. Завадський [4], О.М. М'якшило [7], А.Е. Мазуренко [8], В.В. Осадчий і С.В. Шаров [10] та ін. У статті [1] проведено порівняльний аналіз реляційної бази даних з нереляційною та виділено основні переваги кожної з них, що нададуть можливість проаналізувати та визначити, в яких випадках можна і краще використовувати нереляційні бази даних, а в яких навпаки. Дослідження та порівняльний аналіз проведено для СКБД MS SQL Server і MongoDB. Науковцями [5] висвітлено систематизацію та узагальнення існуючих типів і видів баз даних, розглянуто основні підходи до класифікації баз даних та наведено загальну їх характеристику. Визначення основних типів та видів баз даних наведено у хронологічному порядку їх розвитку.

Метою статті є проектування бази даних для зберігання подібної до гри «Minecraft» інформації за допомогою Microsoft SQL Server, створення симуляції названої вище гри у вигляді сайту, використовуючи мову програмування C# та платформу ASP.NET з його інструментами та особливостями для отримання й закріплення знань використання баз даних в ігрових продуктах.

Викладення основного матеріалу. Проаналізуємо сам ігровий продукт, на основі якого буде спроектовано альтернативну базу даних. Minecraft – комп'ютерна гра, розроблена інді-студією «Mojang Studios» та випущена вперше в 2011 році. Наразі належить компанії «Microsoft», яка оновлює гру по сьогодні. Гра має безмежний світ з великою кількістю цікавих місць, кожна з яких складається з окремих елементів – блоків. Блоки можна руйнувати та отримувати різні предмети. Останні можна отримати також з сутностей – мобів, які можна знайти по світу. З отриманих ресурсів можна створювати нові або використовувати в світі для різних цілей. Метою гри є дійти до темного світу та перемогти дракона.

Насправді вбити дракона – не основний центр тяжіння гравців до гри. Вона дає можливість будувати будь-які споруди або краєвиди, робити цікаві проекти та навіть збудувати справжній простий комп'ютер за допомогою спеціальних блоків.

Судячи з вказаного вище опису гри, потрібно зазначити, що даних для збереження всіх видів блоків, мобів, ресурсів тощо, а також характеристики для їх використання або взаємодії є величезна кількість, тому використання СУБД дозволить впорядкувати ці дані та отримувати потрібні в швидкий спосіб.

Перед вибором СУБД для формування бази даних проведено аналіз трьох існуючих систем (Microsoft SQL Server, Oracle, MongoDB) та визначено, що перший найкраще всього підходить для реалізації поставлених завдань. Для симуляції обраного ігрового продукту та взаємодії з базою даних вирішено використати платформу ASP.NET та мову програмування C#. Мова програмування обрана для отримання практичних навичок використання сформованої бази даних у продвинутих ігрових двигунах, де вона застосовується як основний інструмент написання взаємодії між об'єктами та станами. Платформа в свою чергу допоможе легко створити програмний продукт, що представлятиме поведінку даних. Для симуляції ігрового процесу буде створена текстова гра за допомогою вебтехнологій та засобів розробки.

Проектування бази даних. Перед проектуванням бази даних та продукту представлення потрібно визначити точний функціонал програмного продукту, де вона буде використана. Як вже було раніше зазначено, інформаційну систему буде реалізовано у вигляді текстової гри зі змінюваними параметрами на сайті. Власне сайт буде містити такі функціональні блоки: робота з гравцем, робота зі світами, ігровий режим та режим адміністратора. Детальна взаємодія з продуктом буде наведена у вигляді схеми (рис. 1).

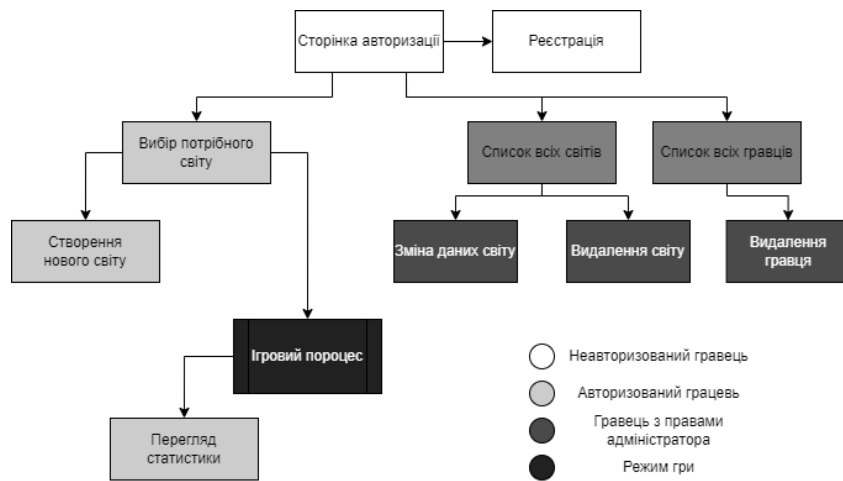


Рис. 1. Схема повного функціоналу продукту

Можна побачити, що блок «Режим гри» виділено окремо, тому що саме там зосереджена найбільша кількість звернень до бази даних та власне функціоналу гри, тому для цього блоку буде надано ще одну схему (рис. 2).

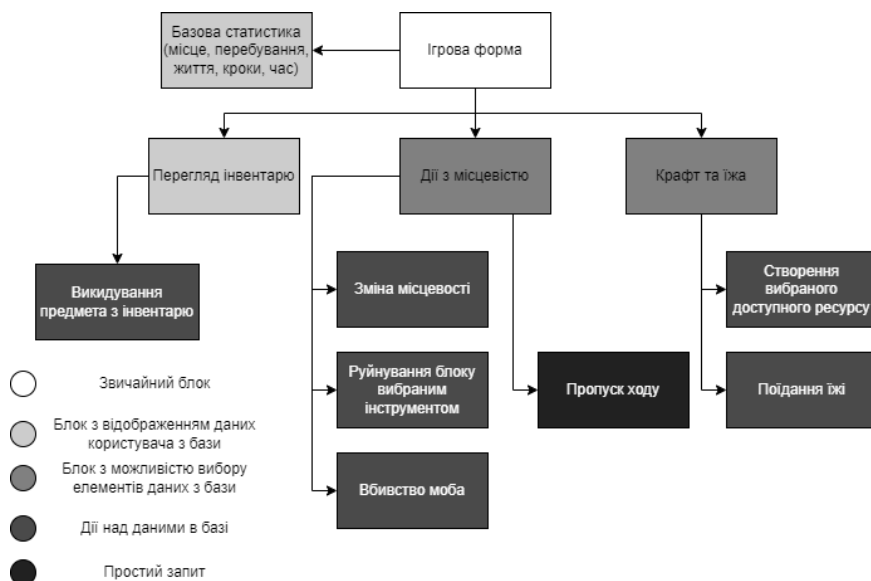


Рис. 2. Схема функціоналу режиму гри ігрового продукту

Схема бази даних буде великою, адже гра містить велику кількість як сутностей, так і записів у них, тому розбір спроектованої схеми бази даних (рис. 3) буде розділено на пару блоків, які описуватимуть свою сутність та пов'язані з ними доповнення.

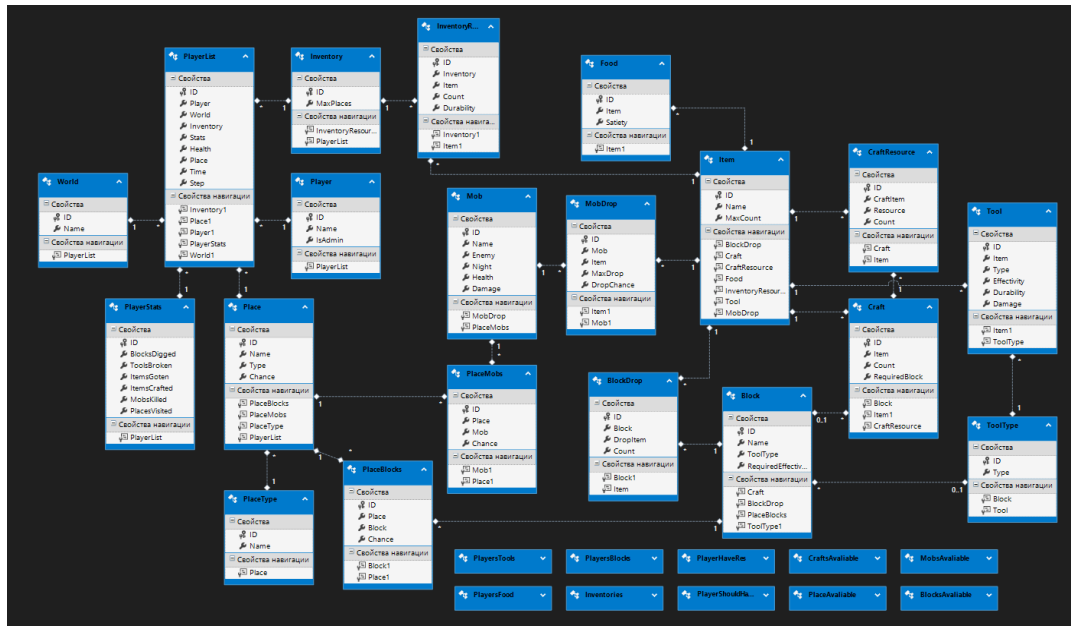


Рис. 3. Схема бази даних

У таблиці 1 наведено блоки бази даних та вказано їх зміст. Всього таких блоків буде 11, зокрема це блок світу, гравця, інвентарю тощо.

Таблиця 1

Блоки бази даних та їх зміст

Блок	Опис	Таблиця
Світу	Зберігає лише всі створені світи та їх назви. Аспекти, що стосуються світу, використовуються однаково для всіх світів	World
Гравця	Посилання на таблиці, які й формують гру: світ, інвентар, місце розташування, кількість сердець, зроблених кроків та час у світі	Player
		PlayerList
		PlayerStats
Інвентарю	Власне інвентар для кожного гравця у відповідному світі з вказівкою на максимальну кількість предметів та список ресурсів, що зберігає всі наявні в гравця предмети, які він створив або отримав внаслідок дій	Inventory
		InventoryResources
Місця	Власне локації та його типу, щоб можна було відділити, наприклад, підземні та надземні локації, а також додатковий тип, який з'єднуватиме два попередні. Також до нього належать таблиці блоків та мобів для кожної локації, які додатково зберігають шанс появи	Place
		PlaceType
		PlaceMobs
		PlaceBlocks
Блоків	Список блоків та предметів, що з них можна отримати після видобутку	BlockDrop
Мобів	Власне мобі та предмети, що з них можна отримати	Mob
		MobDrop
Їжі	Список їжі	Food
Інструментів	Список інструментів, який зберігає ефективність, довговічність та кількість шкоди предмета, та типу інструменту	Tool
		ToolType
Крафтів	Список предметів, на які є рецепти, яка містить також кількість отриманого предмета після створення, та список ресурсів, які потрібні для створення та які спишуться з інвентаря після дії	CraftResource
Предметів	Більшість блоків мають посилання на цю таблицю, адже вона описує всі існуючі предмети всіх типів, їх назву та максимальну їх кількість в одному слоті	Craft
		Item
Узагальнюючий	Блок представлень. Непов'язаний з основною схемою блок, адже фактично це запити до таблиць. Він містить всі представлення, які використовуватимуться для простого отримання інформації про потрібні речі	Рисунок 4

Представлення узагальнюючого блоку наведено на рисунку 4.

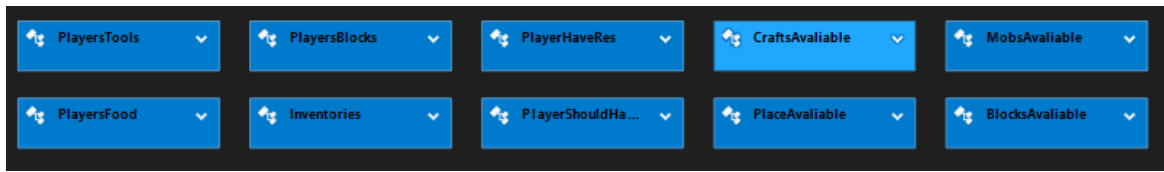


Рис. 4. Представлення узагальнюючого блоку

Для спрощення роботи з отриманим об'ємом даних БД потрібно передбачити процедури та тригери, які використовуватимуться. Серед реалізованих процедур розглянемо три найцікавіших.

Процедура *CraftItemToInventory* створює переданий предмет за рецептом, якщо такий існує. Спочатку перевіряє та прибирає предмети, які потрібні для створення, за допомогою *DropItemFromInventory* та при успішній перевірці додає вказану в рецепті кількість предмета до інвентарю. Процедура *DropItemFromInventory* прибирає вказану кількість переданого предмета по ідентифікатору зі списку всіх предметів інвентарю вказаного гравця, який знаходиться у вказаному світі. Перед прибиранням перевіряє чи є слоти, які містять потрібний предмет. Процедура *CreateNewWorld* створює новий світ з вказаною назвою та додає до списку гравців створеного світу переданого гравця.

Таким же чином опишемо три тригери, які полегшать внесення даних в базу даних.

Тригер *AddingPlayerToPlayerList* – перед додаванням гравця до списку гравців світу створює записи в таблиці статистики та інвентарю і записує саме їх до запису. Також вибирає довільну наземну локацію як точку появи. Тригер *DeletingWorld* – перед видаленням світу видаляє всіх гравців, що з ним пов'язані. Тригер *CheckingStepsInPlayerList* – перед зміненням кількості зробленим гравцем кроків перевіряє їх кількість. Якщо кількість націло ділиться на 40, змінює час у світі.

Всі названі вище аспекти будуть доречні при розробці програмного продукту. Для використання бази даних знадобиться спеціальний для обраної платформи фреймворк Entity Framework.

Реалізація програмного продукту. Розглянемо інтерфейс програмного продукту та симуляцію, що будуть реалізовані за допомогою використання HTML, CSS та мови програмування C#. При різних статусах авторизації та правах навігаційна панель змінюється. Вигляд головної сторінки презентовано на рисунку 5.

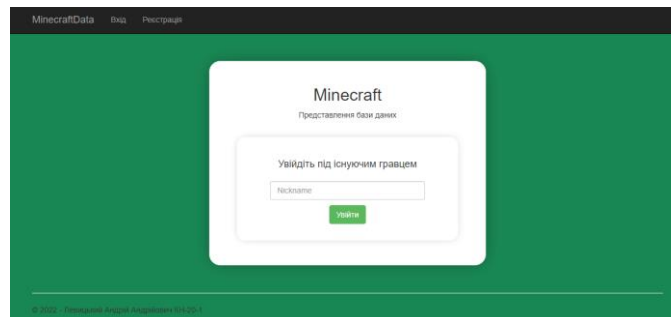


Рис. 5. Головна сторінка програмного продукту

Інтерфейс вибору світу для симуляції гри продемонстровано на рисунку 6.

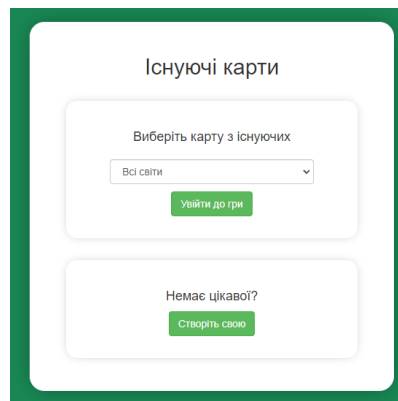


Рис. 6. Інтерфейс вибору світу для симуляції гри

Інтерфейс ігрового режиму реалізовано таким чином (рис. 7).

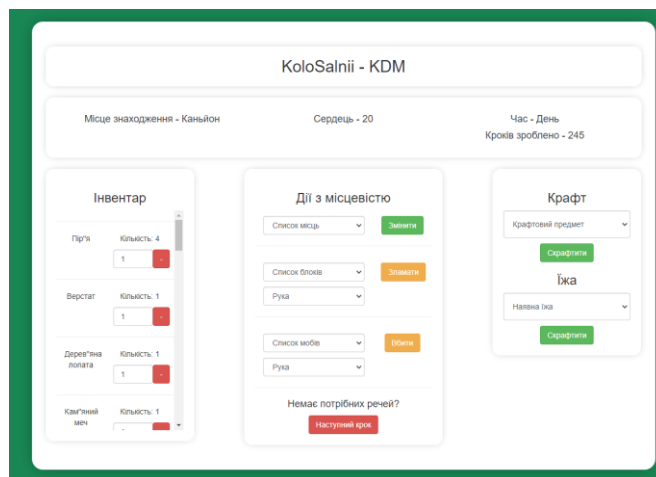


Рис. 7. Інтерфейс ігрового режиму

Після кожної дії, особливо в ігровому режимі, виводиться інформаційний блок з успішністю або неуспішністю виконання та повідомленнями про невиконані дії, якщо, наприклад, гравець не авторизований або гравець довго не взаємодіяв з сайтом. Також про успішність виконання дії свідчить колір тексту, який відображається в інформаційному блоці. Для звітності існує окрема таблиця в базі даних, наведена в таблиці 1, що відповідає за кількість всіх дій гравців на відповідній карті.

Для виведення інформації про статистику гравця створено окрему сторінку (рис. 8). Вона проста в оформленні, але за допомогою неї можна побачити дещо розширену інформацію про дії гравця.

Статистика гравця KoloSalnii в світі KDM	
Основні відомості	
Сердець	20
Зроблено ходів	245
Розширена інформація	
Викопано блоків	87
Отримано предметів	186
Створено предметів	69
Мобів вбито	9
Розвідано місць	39

Рис. 8. Інтерфейс зведеної статистики поточного гравця на обраному світі

Адміністрування бази даних. Адміністрування у розробленій БД поділяється на два рівні: користувачський та системний.

Користувачський – рівень, що дозволяє керувати базою даних за рахунок дозволів, які вказуються в таблиці їх визначення. Всього є 2 ролі: гравець – має можливість грати в гру, реєструватися та створювати власні карти; адміністратор – має права гравця та можливості переглядати списки всіх гравців та світів, видаляти їх та змінювати дані світів.

Системний – захист на рівні бази даних. Він також має 2 ролі: гравець – має можливість лише оновлювати або додавати до таблиць гравців та світів, виконувати всі дії над списком гравців кожного світу, статистики, інвентарю та списку ресурсів в них і лише зчитувати дані з інших таблиць; адміністратор має повний доступ до всієї бази даних. Саме він в змозі надати права адміністратора в користувачському рівні.

Системний рівень стосується власне СУБД та налаштовується в ній. Для користувачського рівня в інформаційній системі створень підключення до бази даних саме за роллю «Гравець» системного рівня.

Налаштування підключення до бази даних відбувається при імпортуванні схеми бази. Під час цього етапу проситься ввести дані користувача, під яким буде виконуватися робота з базою даних, тому значення підключення генерується автоматично та записується до основного коду сутності бази. З огляду на ці алгоритми виконувати налаштування з боку користувача не є обов'язковим.

Висновки та перспективи подальших досліджень. Під час виконання поставлених завдань здобуто практичні навички з використання основних засобів створення та налаштування бази даних за допомогою однієї з розповсюджених СУБД Microsoft SQL Server. Отримано налагоджену базу даних на базі гри «Minecraft», яку використано в розробці інформаційної системи у вигляді текстової гри на сайті. В результаті проектування бази даних здобуто цікавий досвід створення структурованої бази збереження ігрової інформації, яка дозволить у подальшому створювати ігри з великою кількістю інформаційних потоків.

Список використаної літератури:

1. Брацький В.О. Дослідження особливостей застосування реляційних і нереляційних баз даних на прикладі SQL Server та MongoDB / В.О. Брацький, О.М. М'якишко // Наукові праці Національного університету харчових технологій. – 2016. – Т. 22, № 5. – С. 15–24 [Електронний ресурс]. – Режим доступу : http://nbuv.gov.ua/UJRN/Npukht_2016_22_5_4.
2. Гайна Г.А. Основи проектування баз даних : навч. посіб. / Г.А. Гайна. – К. : КНУБА, 2005. – 204 с.
3. Гужва В.М. Інформаційні системи і технології на підприємствах : навч. посіб. / В.М. Гужва. – К. : КНЕУ, 2001. – 400 с.
4. Завадський І.О. Основи баз даних : навч. посіб. / І.О. Завадський. – К. : Видавець І.О. Завадський, 2011. – 192 с.
5. Зінов'єва І.С. Сучасні підходи до подальшої еволюції концепції баз даних / І.С. Зінов'єва // Scientific Publishing Center «Sci-conf. com. Ua». – 2019. – С. 34–44 [Електронний ресурс]. – Режим доступу : https://ir.kneu.edu.ua/bitstream/handle/2010/38140/Zin_2019_2.pdf?sequence=1.
6. Теоретичні аспекти розробки системи управління навчанням / О.Коротун, Т.Вакалюк, В.Зубрицький, І.Гордієнко // Таврійський науковий вісник. Сер. : Технічні науки. – 2022. – № 1. – С. 36–46. DOI: 10.32851/tnv-tech.2022.1.4.
7. М'якишко О.М. Організація баз даних та знань : навч. посіб. / О.М. М'якишко. – К. : НУХТ, 2013. – 148 с.
8. Мазуренко А.Е. Аудит настроек безопасности MS SQL SERVER / А.Е. Мазуренко // Захист інформації. – 2017. – Т. 19, № 1. – С. 43–46. [Електронний ресурс]. – Режим доступу : http://nbuv.gov.ua/UJRN/Zi_2017_19_1_8.
9. Створення та обробка баз даних : навч. посіб. / Л.С. Глоба, М.Ю. Тернова, Р.Л. Новогрудська, О.С. Штогриня. – К. : НТУ України «КПІ», 2013. – 477 с.
10. Шаров С.В. Бази даних та інформаційні системи : навч. посіб. / С.В. Шаров, В.В. Осадчий. – Мелітополь : Вид-во МДПУ ім. Б. Хмельницького, 2014. – 352 с.

References:

1. Bratskyi, V.O. and Miakshylo, O.M. (2016), «Doslidzhennia osoblyvostei zastosuvannia reliatsiinykh i nereliatsiinykh baz danykh na prykladi SQL Server ta MongoDB», *Naukovi pratsi Natsionalnoho universytetu kharchovykh tekhnolohii*, Vol. 22, No. 5, pp. 15–24, [Online], available at: http://nbuv.gov.ua/UJRN/Npukht_2016_22_5_4
2. Haina, H.A. (2005), *Osnovy proektuvannia baz danykh*, navch. posib., KNUBA, K., 204 p.
3. Huzhva, V.M. (2001), *Informatsiini systemy i tekhnolohii na pidpriemstvakh*, navch. posib., KNEU, K., 400 p.
4. Zavadskyi, I.O. (2011), *Osnovy baz danykh*, navch. posib., Vydavets I.O. Zavadskyi, K., 192 p.
5. Zinovieva, I.S. (2019), «Suchasni pidkhody do podalshoi evoliutsii kontseptsii baz danykh», *Scientific Publishing Center «Sci-conf. com. Ua»*, pp. 34–44, [Online], available at: https://ir.kneu.edu.ua/bitstream/handle/2010/38140/Zin_2019_2.pdf?sequence=1
6. Korotun, O., Vakaliuk, T., Zubrytskyi, V. and Hordiienko, I. (2022), «Teoretychni aspekty rozrobky systemy upravlinnia navchanniam», *Tavriiskyi naukovyi visnyk. Ser. Tekhnichni nauky*, No. 1, pp. 36–46, doi: 10.32851/tnv-tech.2022.1.4.
7. Miakshylo, O.M. (2013), *Orhanizatsiia baz danykh ta znan*, navch. posib., NUKhT, K., 148 p.
8. Mazurenko, A.E. (2017), «Audyt nastroek bezopasnosty MS SQL SERVER», *Zakhyst informatsii*, Vol. 19, No. 1, pp. 43–46, [Online], available at: http://nbuv.gov.ua/UJRN/Zi_2017_19_1_8.
9. Hloba, L.S., Ternova, M.Iu., Novohrudska, R.L. and Shtohryna, O.S. (2013), *Stvorennia ta obrobka baz danykh*, navch. posib., NTU Ukrainy «KPI», K., 477 p.
10. Sharov, S.V. and Osadchyi, V.V. (2014), *Bazy danykh ta informatsiini systemy*, navch. posib., Vyd-vo MDPU im. B. Khmelnytskoho, Melitopol, 352 p.

Сугоняк Інна Іванівна – кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0002-0484-4839>.

Наукові інтереси:

- бази даних;
- адміністрування баз даних;
- хмарні обчислення.

Коротун Ольга Володимирівна – кандидат педагогічних наук, доцент кафедри комп’ютерних наук Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0003-2240-7891>.

Наукові інтереси:

- бази даних;
- інтелектуальний аналіз баз даних.

Левицький Андрій Андрійович – здобувач вищої освіти Державного університету «Житомирська політехніка».

<https://orcid.org/0009-0009-5138-3535>.

Наукові інтереси:

- бази даних;
- ігрові симуляції.

Сугоняк Владислав Андрійович – студент-магістр Державного університету «Житомирська політехніка».

<https://orcid.org/0009-0005-1576-2555>.

Наукові інтереси:

- хмарні технології.

Suhoniak I.I., Korotun O.V., Levytskyi A.A., Suhoniak V.A.

Designing a database for the game «Minecraft»

The purpose of the paper is to describe reverse engineering of the popular game Minecraft in terms of the interaction of game data and its storage for further use, and on the basis of the obtained data to design a database and a text simulation of the above-mentioned game in order to acquire skills in the formation of structured data systems for game products, which require complex data interaction. The paper highlights the basic algorithms of database formation and interaction between data as required by the type of game product and its content. Four stages of software product development are demonstrated. At the first stage, the concept for designing the database was formed, the data to be stored was analyzed, and the idea for creating an information system was determined, as well as the tools used for projecting software products and DBMS. The second stage will describe the formation of a database with all features, representations, procedures and triggers for comfortable data processing when designing a software product, and the use of the database in the information system and work with it during various actions of a player. The third stage is the appearance, structure and operation of game simulations in the form of a web application, the peculiarities of the interactions between this platform and a database and the concept of the game process. Also, at this stage, the finished software product, which operates with the created database and simulates the game «Minecraft», is presented. The implementation of database access security is considered at the fourth stage. The Microsoft SQL Server DBMS was used to work with the data, the simulation of the game was created in the form of a web-site using the C# programming language, the ASP.NET platform, and the Entity Framework database framework.

Keywords: DBMS; gaming product; structuring; SQL; C#; ASP.NET; Minecraft.

Стаття надійшла до редакції 11.04.2023.