Міністерство освіти і науки України Ніжинський державний університет імені Миколи Гоголя

Навчально-науковий інститут природничо-математичних, медико- біологічних наук та інформаційних технологій

Кафедра інформаційних технологій, фізико-математичних та економічних наук

Освітня програма: Комп’ютерні науки Спеціальність:122 Компʼютерні науки

**КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття освітнього ступеня *магістр*

Програмна реалізація задачі автоматизації роботи готельного бізнесу

## студентки Лахно Анастасії Ігорівни Науковий керівник:

Фетісов Валерій Сергійович, кандидат

економічних наук, доцент

## Рецензенти:

Казачков Іван Васильович, доктор технічних наук, професор

Кресан Тетяни Анатоліївни, доцент, кандидат технічних наук

**Допущено до захисту:** 2024 р.

Завідувач кафедри

проф. Казачков І.В.

Ніжин – 2024

АНОТАЦІЯ

Метою роботи є створення інтегрованої системи для оптимізації та автоматизації управління готелем. Об’єкт дослідження – це бізнес-процеси готелів, що підлягають автоматизації для підвищення ефективності та якості обслуговування. Предмет дослідження – автоматизація завдань готельного бізнесу та її вплив на ефективність управління.

Магістерська робота складається зі вступу, двох розділів основної частини, висновків та списку джерел. У першому і другому розділах розглянуто теоретичні аспекти систем автоматизації та популярні програмні продукти. Третій і четвертий розділ описує розробку бази даних як ключового елемента системи.

За результатами роботи буде розроблено базу даних для автоматизації готельного підприємства, яка допоможе покращити виконання процесів на підприємстві у перспективі.

**Ключові слова:** система автоматизації, готелі, оптимізація.

# ANNOTATION

The aim of this work is to create an integrated system for optimizing and automating hotel management. The object of the study is the business processes of hotels that are subject to automation to improve efficiency and service quality. The subject is the automation of hotel business tasks and its impact on management efficiency.

The master's thesis consists of an introduction, two chapters of the main part, conclusions and a list of references. The first and second chapters discuss the theoretical aspects of automation systems and popular software products. The third and fourth sections describe the development of a database as a key element of the system.

Based on the results of the work, a database for the automation of a hotel enterprise will be developed, which will help to improve the implementation of processes at the enterprise in the future.

**Keywords**: automation system, hotels, optimization.

ЗМІСТ

|  |  |
| --- | --- |
| ВСТУП | 4 |
| Розділ 1. Системи автоматизації готельних підприємств………………………………….. | 7 |
| 1.1. Загальні тенденції розвитку готельного бізнесу…………………………………….. | 7 |
| 1.2 Структура систем автоматизації готелів……………………………………………… | 10 |
| 1.3 Мови програмування для систем автоматизації готельного бізнесу………………... | 12 |
| 1.4 Приклади існуючих систем автоматизації для готелів………………………………. | 16 |
| РОЗДІЛ 2. БАЗА ДАНИХ, ЯК НЕВІД’ЄМНА СКЛАДОВА СИСТЕМИ  АВТОМАТИЗАЦІЇ……………………………………………………………………………. | 26 |
| РОЗДІЛ 3. ХАРАКТЕРИСТИКА СУБД MYSQL…………………………………………… | 28 |
| 3.1.1 Основні можливості та архітектура MySQL ………………………………………… | 28 |
| 3.1.2 Функціональність MySQL…………………………………………………………….. | 29 |
| 3.2 Характеристика СУБД MySQ……………………………………………………………. | 30 |
| 3.3 Переваги та обмеження MySQL ………………………………………………………... | 33 |
| 3.4 Використання MySQL у практичних проєктах ………………………………………... | 33 |
| 3.5 Клієнт HeidiSQL: функції, приклади застосування та переваги впровадження……… | 34 |
| РОЗДІЛ 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ……………………………………………………... | 36 |
| 4.1 Встановлення на комп’ютер необхідного програмного забезпечення……………… | 36 |
| 4.1.1 Встановлення HeidiSQL……………………………………………………………… | 38 |
| 4.2 Концептуальне моделювання даних…………………………………………………….. | 39 |
| 4.2.1 Побудова ER-діаграми………………………………………………………………… | 40 |
| 4.3 Реалізація запитів, збережених процедур та тригерів в HeidiSQL……………… | 44 |
| 4.3.1 Створення запитів…………………………………………………………………….. | 44 |
| 4.3.1.1 Список всіх бронювань з інформацією про гостей готелю та номери……………… | 45 |
| 4.3.1.2 Переоцінка цін вартості номерів з урахування вартості коефіцієнту інфляції | 46 |
| ВИСНОВКИ…………………………………………………………………………………… | 46 |
| ДОДАТКИ……………………………………………………………………………………… | 49 |

# ВСТУП

Актуальність теми. Сьогодні під впливом розвитку цифрових технологій готельний бізнес зазнає швидких змін, змінюються методи менеджменту та обслуговування клієнтів. Автоматизація всіх ключових готельних процесів стала необхідною умовою забезпечення конкурентоспроможності на ринку та підвищення ефективності роботи.

Зростаючі вимоги до якості обслуговування з боку гостей готелю та необхідність постійної оптимізації витрат спонукали власників готелів до впровадження автоматизованих систем, які допоможуть покращити процеси управління ресурсами та допоможуть створити позитивний досвід для гостей, що, у свою чергу, забезпечить стабільний дохід.

Системи автоматизації для готельних підприємств допомагають автоматизувати такі процеси, як обробка даних про клієнтів, бронювання номерів в готелі і столиків в ресторані, та управління фінансами. Дане програмне забезпечення не тільки допомагає полегшити роботу персоналу, але й дозволяє надавати клієнтам більш високий рівень обслуговування, зокрема, завдяки можливості персоналізації послуг та швидкому реагуванню на потреби гостей підприємства. Системи автоматизації створюють єдину інформаційну базу, що сприяє оптимізації роботи всіх підрозділів готелю, включаючи відділ обслуговування, фінанси, логістику і маркетинг.

Актуальність дослідження полягає в зростаючому попиті на рішення автоматизації в сфері гостинності, особливо в умовах цифровізації економіки та посилення конкуренції. Системи управління готелем (PMS) і комплексні платформи автоматизації стали невід’ємною частиною успішних готелів у всьому світі.

В межах даного дослідження будуть розглянуто ключові особливості готельного програмного забезпечення, а також розглянуто, як автоматизація впливає на якість обслуговування та оптимізацію витрат.

Дослідження теми створення бази даних для системи автоматизації в готельному бізнесі є практичним і необхідним для обрання шляхів

підвищення ефективності роботи підприємств сфери гостинності та забезпечення відповідності сучасним стандартам обслуговування клієнтів.

**Об'єктом дослідження** даної дипломної роботи є інформаційна система для автоматизації управління операційною діяльністю, яка може бути впроваджена у готелях для підтримки повсякденної діяльності та забезпечення конкурентоспроможності в індустрії гостинності.

**Предмет дослідження** – методи, інструменти та технології, що забезпечують розробку і впровадження системи автоматизації для управління ключовими процесами готелю.

**Метою** кваліфікаційної роботи є розробка бази даних для системи автоматизації готельного бізнесу, яка дозволить оптимізувати ключові процеси управління та обслуговування клієнтів, підвищити якість сервісу та знизити витрати.

Відповідно до вказаного об’єкту, предмету і для досягнення поставленої мети визначені такі завдання роботи:

* визначити вимоги до автоматизованої системи на основі потреб готельного бізнесу, включаючи управління бронюванням, облік клієнтів, контроль заїздів і виїздів, управління фінансами та інші важливі аспекти діяльності;
* розробка методів інтеграції системи з іншими програмними рішеннями, що використовуються в готелях (CRM, системи управління доходами, POS- системи), для забезпечення комплексного підходу до автоматизації всіх процесів;
* реалізувати прототип системи автоматизації та його тестування на базі даних з метою оцінки ефективності розробленого програмного рішення;
* зробити аналіз існуючих на ринку системи автоматизації готелів, і проаналізувати їх сильні та слабкі сторони, зробити висновки, щодо ключових характеристики, що повністю відповідають вимогам новітніх технологій в сучасних готелях;
* розробити користувальницький інтерфейс системи так, щоб він був

зручним та інтуїтивно зрозумілим для персоналу готелю, спрощуючи процес навчання та підвищуючи ефективність взаємодії з системою;

* проаналізувати результатів тестування прототипу для виявлення можливих дефектів, оптимізації системи та внесення необхідних корекцій;
* розробити базу даних для систем автоматизації з урахуванням потреб управління ресурсами, обслуговування клієнтів і зберігання інформації. Це передбачає проектування основних модулів системи та визначення їх функціональних можливостей;
* визначити економічну доцільність впровадження даної бази даних для автоматизованих систем у готельну діяльність та зробити висновок, щодо їх впливу на якість обслуговування клієнтів, заповнюваність номерів та загальну ефективність управління.

**Методи дослідження**. Аналітичні методи оброблення інформації, методи аналізу та синтезу.

**Наукова новизна** отриманих результатів полягає в розробці та впровадженні програмного забезпечення для автоматизації основних бізнес- процесів готельного господарства.

Робота **має практичне значення**, оскільки результати дослідження можуть бути використані малими та середніми підприємствами для вибору оптимальних рішень для вирішення завдань автоматизації процесів в готельних підприємств.

**Апробація результатів дослідження.** Опубліковано статтю

«Характеристика систем автоматизації готелів» у «Віснику студентського наукового товариства» випуск 30 [7]. Також були опубліковані тези на тему

«Переваги впровадження системи автоматизації готельного підприємства» на XVI Всеукраїнській студентській науковій конференції «Перспективи розвитку точних наук, економіки та методики їх викладання» (Ніжин, Україна, 13-14 листопада, 2024 р.) [8].

РОЗДІЛ 1.СИСТЕМИ АВТОМАТИЗАЦІЇ ГОТЕЛЬНИХ ПІДПРИЄМСТВ

Система автоматизації готельного бізнесу передбачає використання технологій для виконання завдань, що раніше виконувались вручну, з метою підвищення ефективності та зниження витрат.

Такі системи дають можливість менеджерам та власникам готелів покращувати процес управління, забезпечуючи повний контроль над підприємством. Завдяки цим рішенням можна коригувати ціни на номери, керувати запасами, моніторити зміни на ринку та загалом *підвищувати рівень задоволення клієнтів*.

Автоматизація в готельному бізнесі існує давно, але застосування комп'ютерних систем управління готелем (PMS) з'явилося в 1980-х роках. Зараз автоматизація охоплює все більше завдань – від розподілу номерів до керування доходами та спілкування з гостями. Найбільшої актуальності системи автоматизації набули під час пандемії COVID-19 у 2020 році, коли готелі масово впроваджували безконтактні технології для задоволення нових вимог і скорочення витрат через нестачу персоналу.

Сучасні інформаційні системи також значно спрощують взаємодію готелів з постачальниками і клієнтами через інтернет. Автоматизація внутрішніх процесів включає ведення обліку бронювань, запасів та управління персоналом.

* 1. Загальні тенденції розвитку готельного бізнесу

На сьогоднішній день, сфера готельного бізнес розвивається під впливом глобалізації, цифровізації та змін у поведінці споживачів. Ці фактори визначають тенденції, які суттєво змінюють підходи до ведення готельного бізнесу і формують нові вимоги до організації послуг.

Зміни в роботі після пандемії COVID-19, зростаючий попит на безконтактне обслуговування та попит на екологічність, активне використання новітніх технологій в роботі готельних підприємств, стали основними чинниками, що слугували змінам в індустрії гостинності.

1. *Розвиток технологій та цифровізація*. Однією з найбільш важливих тенденцій в готельному бізнесі є оцифрування всіх аспектів обслуговування. Автоматизація може значно підвищити ефективність бізнесу за рахунок зниження витрат на операційні процеси і мінімізації людського фактору. Зокрема, системи готельного менеджменту (системи управління нерухомістю, PMS) надають можливості для автоматизації бронювання, обліку клієнтів, управління запасами номерів і фінансових потоків.

Крім того, застосовуються рішення, засновані на штучному інтелекті (ШІ) та машинному навчанні, що дозволяє персоналізувати послуги таким чином, щоб вони відповідали потребам гостей. До них відносяться, наприклад, чат- боти для цілодобового спілкування з клієнтами, система рекомендацій і автоматична настройка інтер'єру кімнати відповідно до переваг гостей (наприклад, температура, освітлення і ін.).

1. *Безконтактне обслуговування і цифрові сервіси для обслуговування гостей*. Пандемія COVID-19 стала поштовхом для впровадження безконтактного обслуговування, що в багатьох готелях і досі є стандартом. Для заміни фізичних ключів, дедалі частіше починають використовуватись цифрові ключі або карти, котрими можна скористатись через спеціальні додатки[3].

Процеси заселення та виселення спрощуються за допомогою мобільних додатків, а також звернення до адміністрації і замовлення послуг стали доступними через цифрові канали. Даний підхід знижує ризики передачі інфекцій та забезпечує зручність для клієнтів, які прагнуть мінімізувати контакт з обслуговуючим персоналом.

1. Персоналізація послуг. В час зростаючої конкуренції готелі прагнуть залучати та утримувати клієнтів за допомогою індивідуального підходу та індивідуального обслуговування.

CRM-система (система управління взаємовідносинами з клієнтами) допомагає збирати, узагальнювати та аналізувати дані про гостей, їх смаки та вподобання, на основі попередніх візитів та взаємодій з підприємствами готельного бізнесу.

За допомогою цієї системи, готельне підприємство може пропонувати спеціальні пропозиції та послуги, які відповідають очікуванням гостей. Персоналізація формує лояльності гостей, що особливо важливо в конкурентному середовищі.

1. *Екологічність та сталий розвиток*. Тема екологічної відповідальності привертає до себе ще більше уваги в готельному бізнесі. Готелі постійно впроваджують "зелені" стратегії, орієнтовані на зниження екологічного впливу: використовують енергоефективні технології, знижують споживання води та електроенергії, сортують сміття та зменшують обсяги одноразового пластику.

Гості все частіше обирають готелі, які дотримуються принципів сталого розвитку, тож екологічна відповідальність стала важливою конкурентною перевагою.

1. Орієнтування на локальний вид туризму. Оскільки під час пандемії виникли обмеження на міжнародні подорожі, готелі стали більше орієнтувати курс на місцевих туристів. Саме це підштовхнуло власників готельних підприємств адаптувати свої пропозиції до потреб і можливостей внутрішнього ринку, таким чином створюючи спеціальні пропозиції та пакети для жителів своєї країни. Розвиток локального туризму допомагає підтримувати бізнес навіть за умов обмежень на подорожі[4].
2. Соціальні мережі та онлайн-платформи. Роль сучасних соціальних мереж важко переоцінити, адже більшість комунікації з потенційними гостями відбувається сама завдяки ними. Також вони відіграють дуже важливу роль у просуванні послуг на ринку, і за допомогою відгуків гостей на сайті можна більш детально проаналізувати сильні і слабкі сторони підприємства, що допоможе в удосконаленні роботи в майбутньому. Такі популярні онлайн- платформи для бронювання як Airbnb та Booking.com, мають дуже серйозний вплив на готельний бізнес, оскільки багато гостей обирають спосіб бронювання номерів для проживання саме через Інтернет.

Позитивні відгуки та високі оцінки, що сприяють підвищенню рейтингу серед інших готелів, важливі задля залучення нових клієнтів, тому багато готелів активно працюють над створенням гарних зображень у соціальних мережах та платформах.

На сьогоднішній день, розвиток сфери готельного бізнесу можна відзначити декількома основними факторами, такими як технологічна інноваційність, персоналізація надаваних послуг, екологічність і зростання ролі онлайн-комунікацій.

* 1. Структура систем автоматизації готелів

Система автоматизації готелів є комплексом взаємопов’язаних програмних і апаратних рішень, які спрямовані на оптимізацію всіх аспектів роботи готелю. Вона включає модулі та підсистеми, які інтегруються для забезпечення безперебійного управління операційними процесами[5]. Структура таких систем залежить від розміру готелю, його послуг і вимог, але зазвичай включає кілька ключових компонентів(рис.1).



Система автоматизації готелів

Система бронювання (CRS)

Центральна система управління (PMS)

Система

бізнес-аналітики

Системи, які забезпечують облік та оплату в точках продажу (POS-системи)

Система управління взаємовідносинами з клієнтами (CRM)

Рис. 1 Схема складових компонентів системи автоматизації

Системи автоматизації готелів містять в собі багаторівневу структуру, що об’єднує в собі дані ключові компоненти:

*Центральна система управління (PMS – Property Management System)*

Це основний компонент системи автоматизації, який забезпечує облік та контроль ключових процесів готелю. Також містить функцію керування

бронюванням, реєстрацією гостей, фінансовим обліком і обробляти дані про незаселені номери.

PMS поєднує в собі функції керування структурою господарської служби (housekeeping), проводить аналізування рівня завантаженості готелю та містить необхідні інструменти для підвищення ефективності роботи персоналу.

*Функції PMS:*

* управління бронюваннями (зокрема, інтеграція з платформами бронювання, такими як Booking.com або Expedia);
* адміністрування реєстрації, поселення та виселення гостей;
* контроль стану номерів (зайнятість, прибирання, технічне обслуговування);
* автоматизація фінансових операцій (розрахунок рахунків, інтеграція з банківськими терміналами).

*Особливості:*

* можливість налаштування під специфіку конкретного готелю;
* інтеграція з іншими модулями, такими як CRM або POS;
* можливість автоматичної синхронізації інформації про наявність вільних номерів, що допомогає уникнути овербукінгу.

*CRM-система (Customer Relationship Management)*

CRM-система є ключовим компонентом автоматизації готельного бізнесу, спрямованим на управління взаємовідносинами з гостями. Цей інструмент дозволяє готелю формувати персоналізований підхід до кожного клієнта, підвищувати їхню лояльність і забезпечувати якісний сервіс на всіх етапах взаємодії.

*Функціонал*:

* + збір і збереження інформації про гостей (уподобання, історія бронювань, особливі запити);
  + автоматизація маркетингових кампаній (розсилка спеціальних пропозицій, нагадування);
  + аналіз зворотного зв'язку для покращення сервісу.

*Переваги:*

* + збільшення лояльності гостей;
  + підвищення рівня персоналізації послуг.

*POS-системи (Point of Sale)*

POS-системи (Point of Sale) у готельних системах автоматизації — це ключові інструменти, які забезпечують ефективність, точність і зручність управління фінансовими операціями, продажами та обслуговуванням гостей. Вони інтегруються з іншими модулями готельного програмного забезпечення для створення єдиного інформаційного середовища, що сприяє покращенню сервісу та оптимізації бізнес-процесів[6].

*Функції:*

* + обробка замовлень;
  + інтеграція з рахунком номера гостя;
  + генерація звітів про продажі.

*Переваги*:

* + прискорення обслуговування;
  + точний облік витрат і доходів;
  + допомагає підвищити ефективність роботи персоналу, автоматизуючи рутинні процеси, такі як облік товарів, обробка замовлень та ведення звітності.

*Системи управління енергоресурсами*

Такі системи спрямовані на автоматизацію споживання ресурсів для зменшення витрат та підвищення екологічної ефективності.

*Функціонал:*

* + контроль температури, освітлення, вентиляції;
  + автоматичне вимкнення електрики у незайнятих номерах;
  + відслідковування витрат води та електроенергії;

*Результати:*

* + економія ресурсів;
  + підтримка екологічних стандартів.

*Системи управління безпекою*

Ця складова відповідає за забезпечення безпеки гостей і персоналу.

*Елементи:*

* + електронні замки, які працюють за допомогою карток, кодів або мобільних додатків;
  + відеоспостереження в загальних зонах;
  + системи пожежної сигналізації та аварійного оповіщення.

## Особливості:

* + захист від несанкціонованого доступу;
  + інтеграція з іншими системами, а саме PMS або IoT.
  1. Мови програмування для систем автоматизації готельного бізнесу Фундаментальна частина систем автоматизації в готельному бізнесі є

програмне забезпечення, спрямоване на покращення функцій продуктивності, управління ресурсами та взаємозв’язок систем.

Щоб правильно обрати мову для майбутньої проектованої системи автоматизації готелю необхідно відштовхуватись від висунутих вимог, зо напряму залежать від розміру системи, її функціоналу, простоти обслуговування та синхронізації з іншими системами та Інтернет-платформами.

Розглянемо популярні мови програмування для створення даних систем більш докладніше.

1. **Python** є однією з найпопулярніших мов програмування завдяки її простоті, великій екосистемі бібліотек та кросплатформеності.

Ця мова ідеально підходить для створення бекенд-логіки систем автоматизації готелів, таких як бронювання, управління клієнтами та облік ресурсів[3].

## Переваги:

* + простий синтаксис, який дозволяє швидко розробляти та тестувати функціонал;
  + велика кількість бібліотек для роботи з веб-фреймворками (Django, Flask), базами даних (SQLAlchemy), автоматизації (Celery) та машинного навчання (TensorFlow, Scikit-learn);
  + підтримка інтеграції з іншими системами через API (наприклад, для інтеграції з платіжними системами чи сервісами бронювання).

## Приклади використання:

* + розробка системи бронювання номерів з інтеграцією до сторонніх платформ (Booking.com, Airbnb);
  + аналіз даних для прогнозування завантаженості номерів на основі історичних даних.

1. **Java** є потужною мовою для розробки масштабованих систем, які потребують високого рівня надійності та безпеки. Вона широко використовується в корпоративному середовищі для створення складних систем автоматизації.

## Переваги:

* незалежність від платформи завдяки JVM (Java Virtual Machine), що забезпечує роботу програми на різних операційних системах.
* розвинена екосистема для веб-розробки (Spring Framework), роботи з базами даних (Hibernate) та управління безпекою.
* підтримка багатопотоковості для роботи з великими обсягами даних та виконання кількох задач одночасно.

## Приклади використання:

* розробка модульної ERP-системи для управління готелем, що включає бронювання, облік витрат, управління персоналом і фінансову звітність.
* побудова клієнт-серверної архітектури для синхронізації даних між готельними відділами.

1. **JavaScript (та його фреймворки)** використовується для створення інтерактивних користувацьких інтерфейсів (фронтенду) та серверної логіки завдяки Node.js. У системах автоматизації готельного бізнесу JavaScript забезпечує зручність роботи для адміністратора та користувачів[8].

## Переваги:

* швидкий рендеринг інтерфейсів на стороні користувача завдяки фреймворкам (React, Vue.js, Angular).
* Node.js дозволяє використовувати JavaScript і для бекенду, що спрощує розробку завдяки використанню однієї мови.
* потужна інтеграція з RESTful API та WebSocket для обміну даними в реальному часі.

*Приклади використання*:

* розробка адаптивного веб-додатка для бронювання, який працює на ПК і мобільних пристроях.
* створення реального часу панелі управління, яка відображає завантаженість номерів, статуси прибирання та фінансові показники.

1. **C#** є основною мовою для розробки програм у середовищі NET. Ця мова підходить для створення як веб-додатків, так і десктопних програм, що використовуються в системах автоматизації готелів.

## Переваги:

* глибока інтеграція з платформою Windows, що є популярною в корпоративному середовищі;
* підтримка розробки сучасних веб-застосунків через ASP.NET Core;
* висока швидкість виконання коду завдяки компіляції в проміжний код;

*Приклади використання*:

* розробка локальної системи управління для реєстрації гостей та обліку послуг (з інтеграцією касових апаратів);
* створення мобільних додатків для адміністративного персоналу готелю.

# PHP

PHP є класичним вибором для створення серверної частини веб-додатків. Ця мова часто використовується для готельних систем завдяки простоті та великій кількості готових CMS (наприклад, WordPress).

## Переваги:

* легка інтеграція з популярними веб-серверами (Apache, Nginx).
* підтримка великих спільнот розробників, які створили безліч розширень.
* простота впровадження у вже існуючих веб-сайти готелів.

## Приклади використання:

* розробка системи онлайн-бронювання з інтеграцією платіжних модулів.
* управління контентом на сайті готелю, включаючи відгуки клієнтів та галереї номерів.

## SQL (Structured Query Language)

SQL не є мовою програмування в традиційному розумінні, але є базовим інструментом для роботи з базами даних, які є ядром будь-якої системи автоматизації.

В готельному бізнесі SQL використовується для зберігання, управління та аналізу даних[9].

## Переваги:

* універсальність: SQL підтримується всіма популярними системами управління базами даних (MySQL, PostgreSQL, Microsoft SQL Server).
* оптимізація запитів для роботи з великими наборами даних.
* потужні засоби для створення складних звітів та аналітики.

## Приклади використання:

* зберігання даних про бронювання, клієнтів, номери, співробітників та фінансові операції.
* генерація звітів про завантаженість номерів, фінансові надходження та витрати.
* інтеграція з BI-системами для візуалізації даних.

Отже, кожна з описаних мов має свої сильні сторони і може бути використана в залежності від потреб конкретної системи автоматизації. Python і Java підходять для розробки бекенду, JavaScript забезпечує інтерактивний інтерфейс, а SQL є ключовим інструментом для управління даними. Поєднання цих технологій дозволяє створювати ефективні, гнучкі та масштабовані рішення для готельного бізнесу.

* 1. Приклади існуючих систем автоматизації для готелів

У сучасному готельному бізнесі системи автоматизації є ключовим інструментом для ефективного управління, покращення обслуговування клієнтів та оптимізації внутрішніх процесів. Нижче детально розглянемо

найбільш популярні системи, що використовуються готелями різного масштабу і класу.

## «Opera Property Management System» (Opera PMS)

«Opera Property Management System» (Opera PMS) – це одна з найпопулярніших систем управління готелями, створена компанією «Oracle Hospitality». Вона забезпечує комплексну автоматизацію бізнес-процесів у готельній індустрії, дозволяючи значно полегшити роботу адміністративного персоналу, підвищити ефективність управління та покращити якість обслуговування гостей.

*Основні функції та можливості:*

* + управління всіма аспектами бронювання: від онлайн-бронювання до індивідуальних запитів клієнтів;
  + інтеграція з системами глобальної дистрибуції (GDS) для централізованого управління бронюванням з різних джерел (Booking.com, Expedia тощо) ;
  + управління фінансами: автоматичне створення рахунків для гостей, облік депозитів, обробка платежів;
  + контроль за станом номерів у реальному часі (наприклад, зайняті, вільні, заброньовані).

*Переваги використання «Opera PMS»*

*Масштабованість*: система автоматизації «Opera PMS» підходить як для невеликих готелів, так і для великих міжнародних мереж за допомогою неї можна одночасно керувати одним підприємством або навіть цілою мережею за допомогою однієї системи[10].

*Хмарні та локальні рішення*: cистема може працювати як у хмарному середовищі (Cloud), так і на локальних серверах готелю, залежно від потреб бізнесу.

*Гнучкість налаштувань*: Opera PMS легко адаптується до індивідуальних потреб готелю. Адміністрація може налаштувати систему під свої стандарти обслуговування та управління.

*Покращення якості обслуговування*: за допомогою Opera PMS персонал готелю швидше реагує на запити гостей, краще розуміє їхні потреби та забезпечує високий рівень сервісу.

*Ефективне управління ресурсами*: система дозволяє оптимізувати використання людських і матеріальних ресурсів, автоматизуючи рутинні процеси.

*Недоліки*:

* висока вартість ліцензії, що робить систему недоступною для малих та середніх готелів;
* складність у налаштуванні та необхідність тривалого навчання персоналу;
* потреба у високій якості інтернет-з'єднання для коректної роботи.

**Приклад використання**: мережі «Marriott» і «Hilton» активно застосовують Opera PMS для автоматизації управління своїми готелями. Ця система дозволяє мережевим готелям об'єднувати дані з усіх об'єктів у єдиній базі.

## «Protel Hotel Software»

«Protel Hotel Software» – це сучасна система автоматизації управління готелями, яка забезпечує простоту використання, багатофункціональність і гнучкість. Розроблена компанією Protel, ця система орієнтована на потреби готелів різних розмірів – від невеликих бутик-готелів до великих мережевих комплексів.

Завдяки хмарній архітектурі та модульному підходу, Protel Hotel Software може адаптуватися до специфічних вимог кожного закладу.

## Функції та можливості:

* + Інтеграція з різними каналами бронювання для автоматичного оновлення доступності номерів у реальному часі.
  + Модуль управління клієнтськими відносинами (CRM), який дозволяє зберігати історію гостей та персоналізувати їхній досвід.
  + Створення звітів для аналізу доходів, завантаженості номерів та ефективності персоналу.
  + Підтримка мобільного доступу для адміністраторів, що дозволяє управляти процесами з будь-якого пристрою.

*Переваги «Protel Hotel Software»*

*Масштабованість*: Protel підходить як для невеликих готелів, так і для великих готельних мереж. Завдяки модульному підходу готелі можуть вибирати лише ті функції, які їм потрібні.

*Простота використання* зручний і зрозумілий інтерфейс дозволяє персоналу швидко освоїти роботу із системою.

*Широка інтеграція*: Protel підтримує інтеграцію з понад 100 сторонніми платформами та сервісами, що робить її гнучким рішенням для будь-якого готелю.

*Підтримка кількох мов і валют*: система забезпечує зручну роботу з іноземними гостями та дозволяє вести фінансовий облік у різних валютах.

*Оперативна підтримка*: компанія Protel надає клієнтам цілодобову технічну підтримку та регулярні оновлення програмного забезпечення[7].

*Недоліки***:**

* *cкладність освоєння*: для нових користувачів може знадобитися значний час на навчання через велику кількість функцій;
* вартість: система може бути дорогою для невеликих готелів, особливо якщо враховувати плату за ліцензію, підтримку та додаткові модулі;
* потреба в технічній підтримці, тому що налаштування та інтеграція з іншими сервісами часто вимагають допомоги технічних спеціалістів;
* обмеження хмарної версії: у деяких випадках, залежність від стабільного інтернет-з'єднання може створювати труднощі.

*Приклад використання*: дана програма популярна серед невеликих і середніх готелів через простоту використання та швидкість впровадження. Наприклад, багато бутик-готелів використовують Protel для інтеграції з платформами онлайн-бронювання.

## «RoomKeyPMS»

*«RoomKeyPMS»* – це хмарна система управління готелем (PMS), яка забезпечує інструменти для управління бронюваннями, обслуговування гостей, фінансового обліку та інтеграції з іншими платформами.

Система була розроблена компанією RSI International, яка спеціалізується на розробці рішень для готельної індустрії. RoomKeyPMS орієнтована на невеликі та середні готелі, пропонуючи зручні інструменти для щоденного управління[7].

Функції та можливості:

* + Управління онлайн-бронюванням із синхронізацією з популярними платформами.
  + CRM для зберігання історії гостей та створення персоналізованих пропозицій.
  + Мобільний додаток для адміністраторів і гостей, який дозволяє оперативно обробляти запити та керувати номерним фондом.
  + Інструменти для створення звітності та аналізу даних про ефективність роботи готелю.

*Переваги «RoomKeyPMS»*:

*Хмарна платформа*: забезпечує доступ до системи з будь-якого пристрою та місця, усуваючи потребу в локальних серверах.

*Інтеграція з OTA та Channel Manager*: дозволяє синхронізувати бронювання з платформами, такими як Booking.com та іншими, в режимі реального часу.

*Інтуїтивний інтерфейс*: зручний дизайн спрощує навчання персоналу та щоденне використання.

*Інтеграція з POS та CRM*: дозволяє об’єднати всі операції готелю в єдину систему для підвищення ефективності.

*Недоліки «RoomKeyPMS»*:

* залежність від Інтернету: як і будь-яка хмарна система, «RoomKeyPMS» потребує стабільного інтернет-з'єднання, що може бути проблемою в регіонах зі слабким покриттям.
* обмежений функціонал для великих готелів: система більше підходить для невеликих та середніх готелів; для великих готелів може бракувати деяких розширених функцій.
* тривале налаштування інтеграцій: інтеграція з іншими системами може потребувати часу та участі технічної підтримки.

«RoomKeyPMS» є чудовим вибором для готелів, які шукають хмарне рішення з базовим та середнім рівнем функціоналу, але для великих об'єктів може знадобитися розгляд альтернатив із ширшими можливостями.

*Приклад використання*: «RoomKeyPMS» активно використовується в невеликих готелях та пансіонатах, що орієнтуються на доступні за ціною рішення для автоматизації.

## «Amadeus Hospitality Solutions»

«Amadeus Hospitality Solutions» – це комплекс програмних рішень для управління готелями, конференц-центрами та іншими об’єктами гостинності. Система розроблена компанією Amadeus IT Group, глобальним лідером у галузі технологій для туристичної та готельної індустрії.

Функції та можливості:

* + управління ціноутворенням на основі аналізу попиту та пропозиції (Revenue Management);
  + глобальна інтеграція бронювань через системи GDS;
  + хмарне управління об’єктами (Property Management);
  + автоматизація розкладу персоналу для оптимального розподілу задач;
  + інструменти для аналітики, які дозволяють визначати найбільш прибуткові напрями діяльності.
  + глобальне охоплення: cистема доступна в багатьох країнах і підтримує багатомовний інтерфейс.

*Переваги «Amadeus Hospitality Solutions»*:

*Широкий спектр функціона*лу: наявні модулі управління бронюваннями, конференціями та подіями, а також інструменти для прогнозування попиту та управління доходами.

*Хмарна архітектура*: доступ до системи з будь-якого місця та пристрою, що забезпечує гнучкість для команди.

*Інтеграція з глобальними дистрибуційними системами (GDS)*: сприяє кращій видимості об'єкта на туристичних платформах і забезпечує ефективну синхронізацію даних.

*Аналітика та звітність*: інструменти для збору й аналізу даних, що допомагають приймати обґрунтовані бізнес-рішення.

*Персоналізація обслуговування гостей*: Система зберігає дані про гостей, що дає змогу надавати персоналізовані послуги та покращувати досвід клієнтів. *Приклад використання*: мережа Accord активно використовує рішення

«Amadeus для координації своїх готелів у різних регіонах, завдяки чому забезпечується стандартизований підхід до управління[9].

## «Mews Hospitality Cloud»

«Mews Hospitality Cloud» – це сучасна хмарна система управління готелями (PMS), розроблена компанією «Mews Systems», яка була заснована в 2012 році в Празі, Чехія.

«Mews» орієнтована на автоматизацію, зручність використання та інтеграцію з іншими платформами, щоб забезпечити безперебійне управління готельними операціями.

Система особливо популярна серед невеликих готелів, хостелів і апартаментів, а також серед сучасних бутик-готелів.

*Функції та можливості*:

* + Інтеграція з понад 500 сторонніми додатками, включаючи POS-системи, CRM та інструменти маркетингу.
  + Функція self-check-in для гостей через кіоски або мобільні додатки.
  + Автоматизація розрахунків, включаючи безконтактні платежі.
  + Можливість інтеграції з "розумними" пристроями для управління енергоспоживанням у номерах.

*Переваги «Mews Hospitality Cloud»*:

*Автоматизація процесів*: спрощує процеси реєстрації гостей через мобільний check-in та check-out та знижує потребу в ручній роботі завдяки автоматичному оновленню бронювань і синхронізації даних.

*Інтуїтивно зрозумілий інтерфейс*: система розроблена з урахуванням сучасного дизайну, що спрощує навчання персоналу і роботу з платформою.

*Хмарна архітектура*: доступ з будь-якого пристрою без потреби у встановленні програмного забезпечення.

*Інтеграції з різними платформами*: підтримка інтеграції з Channel Manager, CRM, платіжними системами та POS-рішеннями для ресторанів.

*Розширені можливості для гостей*: онлайн-бронювання через спеціалізовані платформи або сайт готелю.

*Додаткові функції для персоналізованого сервісу*, наприклад, можливість замовлення послуг через гостьовий портал.

*Аналітика та звітність*: зручні інструменти для відстеження фінансових показників, завантаженості номерів та поведінки гостей.

*Недоліки «Mews Hospitality Cloud»*:

*Потреба в налаштуванні інтеграцій*: для повної функціональності може знадобитися час на інтеграцію з іншими системами.

*Обмеженість для великих готельних мереж*: система більше орієнтована на середні та невеликі готелі, тому що для великих мереж даного функції може бути недостатньо.

«Mews Hospitality Cloud» – це зручне рішення для готелів, які прагнуть автоматизувати процеси, підвищити гнучкість та забезпечити сучасний підхід до управління і обслуговування гостей.

*Приклад використання*: «Mews» широко впроваджується в сучасних "цифрових" готелях, таких як Zoku Amsterdam, які прагнуть мінімізувати адміністративний вплив на процеси обслуговування.

Також дана система використовується «Ovolo Hotels» — австралійська мережа, яка інтегрує «Mews» для автоматизації бронювань і оптимізації взаємодії з клієнтами10].

## StayNTouch

«StayNTouch Cloud» – це сучасна хмарна система управління готелями (PMS), розроблена компанією «StayNTouch», яка входить до складу MCR Hotels, одного з найбільших готельних операторів у США. Система створена для спрощення операційної роботи та покращення взаємодії з гостями за допомогою інноваційних функцій та мобільних технологій.

### Функції та можливості:

* + Мобільний доступ до системи управління для адміністраторів.
  + Автоматизовані запити гостей, включаючи доставку в номер чи обслуговування в ресторані.
  + Система самостійного заселення та виселення через мобільні додатки.
  + Інтеграція з ключовими платформами для управління персоналом.

### Основні переваги «StayNTouch Cloud»:

*Мобільність*: система доступна з будь-якого пристрою, включаючи смартфони та планшети, що забезпечує гнучкість для персоналу.

*Інструменти для мобільного check-in та check-out*, що підвищує зручність для гостей.

*Інтуїтивний інтерфейс*: простий у використанні дизайн скорочує час навчання персоналу та прискорює роботу.

*Персоналізація обслуговування гостей*: система дозволяє зберігати вподобання клієнтів, щоб пропонувати персоналізовані послуги, покращуючи досвід проживання.

*Інтеграція з іншими сервісами*: «StayNTouch Cloud» сумісна з популярними CRM-системами, POS-сервісами, платіжними платформами та системами управління доходами.

*Покращення взаємодії з гостями*: інтеграція з гостьовими додатками для бронювання послуг, спілкування з персоналом та отримання інформації про готель.

*Швидке впровадження*: завдяки хмарній інфраструктурі та спрощеному налаштуванню система легко інтегрується з існуючими процесами.

*Недоліки «StayNTouch Cloud»*:

*Залежність від Інтернету*: для роботи системи потрібне стабільне Інтернет з’єднання, що може бути проблемою у віддалених регіонах.

*Обмежений функціонал для великих мереж*: Для великих готельних мереж система може виявитися недостатньо масштабованою, особливо у специфічних процесах.

*Залежність від тех-підтримки*: Деякі користувачі зазначають, що впровадження нових інтеграцій може вимагати допомоги від служби підтримки. ***Приклад використання***: StayNTouch використовується у невеликих готелях, які прагнуть підвищити рівень цифровізації та надати гостям більше

свободи в управлінні своїм перебуванням[9].

Отже**,** існуючі системи автоматизації для готелів пропонують різноманітний функціонал, орієнтований на різні масштаби бізнесу – від невеликих бутик-готелів до великих міжнародних мереж. Використання таких рішень дозволяє значно покращити обслуговування клієнтів, оптимізувати роботу персоналу та підвищити доходи.

# РОЗДІЛ 2. БАЗА ДАНИХ, ЯК НЕВІД’ЄМНА СКЛАДОВА СИСТЕМИ

**АВТОМАТИЗАЦІЇ**

У сучасному світі автоматизація процесів є засобом підвищення точності, швидкості та продуктивності практично в усіх сферах діяльності, будь то промисловість чи надання послуг.

Відповідно до цього бази даних утворюють основний компонент, який полегшує зберігання, контроль і обробку інформації, необхідної для роботи систем автоматизації. Без баз даних неможливо, функціонування різних частини автоматизованих систем могли працювати узгоджено як програми, обладнання та інтерфейси для користувачів.

База даних використовується для виконання деяких основних функцій у системах автоматизації:

1. Надання інформаційного центру: База даних є основним сховищем інформації, яке використовується системою. Усі дані зберігаються централізовано без резервування, що забезпечує легкий доступ до необхідної інформації.
2. Обробка даних у режимі реального часу. Організований доступ до інформації в СУБД на основі ролей користувачів і рівнів доступу в складних системах є критично важливим, коли багато різних користувачів взаємодіють з однією базою даних.
3. Аналітика. Для сучасних системи автоматизації є надзвичайно важливим, щоб доступ до інформації був з високою швидкістю. База даних повинна забезпечувати прискорену обробку запитів, щоб система могла працювати в режимі реального часу у відповідь на зміну оперативної обстановки.
4. *Інтеграція компонентів системи*. Зв’язок між різними модулями системи автоматизації — датчиками, приводами, інтерфейсами користувача та аналітичними платформами — стає можливим завдяки базі даних.
5. Масштабованість і адаптивність. Нові потреби системи легко задовольняються базами даних, тому подальше розширення або модернізація автоматизованих систем не потребує багато часу та ресурсів[11].

В системах автоматизації готелів можуть бути використані різні бази даних, задля того максимально забезпечити безвідмовну роботу системи. Кожна з них має свої переваги та особливості:

1. *Реляційні бази даних (SQL)*. Вони мають працювати з добре організованими, структурованими даними у формі таблиці Найбільшими перевагами даних баз, є їх висока надійність, гнучкість запитів і здатність підтримувати транзакції. До найбільш використовуваних належать PostgreSQL, MySQL, Oracle.
2. *Бази даних NoSQL*. Це величезні сховища неструктурованих даних, наприклад, із систем IoT або величезні інформаційні (Big Data) проаналізовані купи. Такі бази є гнучкими та високоефективними. Приклади: MongoDB, Cassandra.
3. Графові бази даних або GDB — це один із типів баз даних, які використовує графові структури для зберігання даних. Замість таблиць або документів для представлення та зберігання даних використовуються вузли, ребра та властивості. Оскільки всі пов’язані дані, як правило, знаходяться в безпосередній фізичній близькості один від одного, тому отримати дані досить швидко. Як Neo4j, Amazon Neptune, ArangoDB тощо 4
4. Інтелектуальні бази даних - це бази даних пов'язані з алгоритмами машинного навчання та штучного інтелекту для прогнозування, оптимізації та аналізу великих об'ємів даних. На сьогоднішній день ці бази даних є досить перспективними в розвитку автоматизованих систем.

Системи автоматизації забезпечені засобами «нервової системи» зв’язку між частинами через бази даних. Наприклад, у системах готельного бізнесу, бази даних збирають інформацію про датчики світла, вологості, води та температури, зберігають її та надають операторам аналіз цієї інформації. Це відбувається шляхом контролю показників через бази даних, і за допомого

персоналізації показників під кожного гостя, створюються комфорті умови, що сприяє створенню позитивного досвіду проживання для гостей.

# РОЗДІЛ 3. ХАРАКТЕРИСТИКА СУБД MYSQL

Однією з найпопулярніших у світі систем керування базами даних з відкритим кодом є MySQL. Він надає гнучкі інструменти для зберігання, обробки та керування даними. Поєднання високої продуктивності з мультиплатформеністю та масштабованістю зробило програмне забезпечення корисним для інших дуже різноманітних програм, що працюють на найпростішому та найскладнішому апаратному та програмному забезпеченні.

* 1. Основні можливості та архітектура MySQL

MySQL побудована на клієнт-серверній архітектурі і містить такі компоненти:

* + - сервер бази даних (mysqld) - зберігає дані, обробляє запити, обробляє дані та контролює доступ до них. Сервер прослуховує мережеві підключення клієнтів і зазвичай використовує кілька різних потоків для обслуговування кількох клієнтів на вимогу;
    - клієнти - це програми або інтерфейси, які взаємодіють із сервером MySQL.

Основні функції MySQL включають:

1. Програмне забезпечення з можливістю комерційного використання, присутнє у вільному доступі через GNU General Public License. Також Oracle надає комерційно ліцензовані дистрибутиви, які включають сертифіковане програмне забезпечення цього дистрибутива з відкритим вихідним кодом, яке підходить для використання у виробництві, зі спеціалізованою технічною підтримкою та додатковими функціями для корпоративних клієнтів.
2. Сумісність з різними платформами. Може працювати на різних операційних системах, включаючи Windows, MacOS і Unix, Linux та ін. Це дає змогу запускати у багатоплатформних середовищах і створювати універсальні рішення.
3. Підтримка різних механізмів зберігання даних. Підтримка кількох механізмів зберігання (Storage Engines), включаючи:
   * InnoDB- це механізм транзакцій, який включає підтримку ACID, індексів і зовнішніх ключів;

-MyISAM — це таблиця без підтримки транзакцій, але вона в свою чергу має високу швидкість обробки запитів;

* + MEMORY — дані зберігаються в пам'яті для швидкого доступу.
  + CSV та інші —для специфічних завдань.

1. Гнучкість і розширюваність

MySQL дозволяє використовувати плагіни, які реалізують додаткові або модифіковані можливості сервера. Наприклад, плагіни автентифікації, плагіни аудиту або плагіни, які дозволяють підтримувати певні формати даних[10].

1. Покращена підтримка реплікації та кластеризації
   * Реплікація — це коли дані з одного сервера (майстра), копіюються на інші сервери (репліки) з метою розподіленого зберігання, для підвищення продуктивності або за умови, що дані в кожному місці не однакові.

MySQL Cluster — забезпечує високу доступність даних, автоматичне балансування навантаження та відмовостійкість.

* 1. Функціональність MySQL

MySQL пропонує багатий функціонал для роботи з базами даних:

1. Робота з багатьма типами даних. Підтримує типи чисел NTEGER, FLOAT), текстові типи (VARCHAR, TEXT), типи, пов’язані з часом (DATE, TIME) та інші, які дозволяють створювати структури даних для більшості будь- яких потреб.
2. Інтеграція з іншими мовами програмування. MySQL має драйвери і бібліотеки для роботи з багатьма популярними мовами програмування, такими як PHP, Python, Java, C#, Ruby та іншими. Це дозволяє легко інтегрувати MySQL у веб-програми та програми для настільних ПК.
3. Індексація та оптимізація запитів. Є можливість створення як простих, так і складних індексів для пришвидшення і полегшення пошуку даних. Інструменти аналізу запитів, такі як `EXPLAIN', також можна використовувати для покращення оптимізації запитів і структур бази даних.
4. Автоматизація бізнес-процесів. Включає збережені процедури, тригери, події та планувальники завдань, які зменшать рутинну роботу та зроблять бізнес-процеси ефективнішими та ефективнішими.
5. Швидкість даних та їх обсяг

І горизонтальне, і вертикальне масштабування дають змогу MySQL керувати великими базами даних і багатьма одночасними з’єднаннями[11].

* 1. Характеристика СУБД MySQL

MySQL входить до списку однієї найвідоміших і найпоширеніших систем управління базами даних (СУБД) з відкритим кодом. Вона надає гнучкий інструментарій для зберігання, обробки та управління даними, поєднує в собі високу продуктивність, багатоплатформність та масштабованість. Саме завдяки цьому MySQL використовується для різноманітних завдань, починаючи від створення невеликих веб-додатків до підтримки великих корпоративних систем.

3.1 Основні особливості та архітектура MySQL

MySQL побудована на клієнт-серверній архітектурі, яка складається з двох основних компонентів:

* Сервер баз даних (mysqld) - відповідає за зберігання даних, обробку запитів і забезпечення доступу до них.
* Клієнти- програми або інтерфейси, які надсилають SQL-запити до сервера.

До ключових характеристик MySQL належать:

1. Відкрите програмне забезпечення з можливістю комерційного використання. MySQL є безкоштовною у межах ліцензії GNU General Public License (GPL), що робить її доступною для широкого використання. Крім того, Oracle пропонує комерційні версії з розширеним функціоналом, такими як технічна підтримка та додаткові можливості для підприємств.
2. Сумісність із широким спектром платформ. MySQL працює на різних операційних системах: Windows, Linux, macOS, Unix та інших. Це дозволяє її використовувати в багатоплатформних середовищах і розробляти універсальні рішення.
3. Підтримка різних механізмів зберігання даних. MySQL підтримує кілька механізмів зберігання (Storage Engines), таких як:
   * InnoDB— транзакційний механізм із підтримкою ACID, індексів і зовнішніх ключів.
   * MyISAM — механізм без підтримки транзакцій, але з високою швидкістю обробки запитів.
   * MEMORY — зберігання даних у пам’яті для швидкого доступу.
   * CSV та інші — для специфічних завдань.
4. Гнучкість та розширюваність

MySQL дозволяє використовувати плагіни, які додають або змінюють функціонал системи. Наприклад, плагіни для аутентифікації, моніторингу або підтримки специфічних форматів даних[9].

1. Підтримка реплікації та кластеризації
   * Реплікація — копіювання даних з одного сервера (мастера) на інші (репліки) для розподіленого зберігання, резервного копіювання або підвищення продуктивності.
   * MySQL Cluster — забезпечує високу доступність даних, автоматичне балансування навантаження та відмовостійкість.
   1. Функціональні можливості MySQL

MySQL пропонує багатий набір функцій для роботи з базами даних:

1. Робота з великим спектром типів даних. MySQL підтримує числові типи (INTEGER, FLOAT), текстові типи (VARCHAR, TEXT), датчики (DATE, TIME) та інші, що дозволяє створювати структури даних для будь-яких потреб.
2. Забезпечення інтеграцій із мовами програмування .СУБД має драйвери та бібліотеки для роботи з популярними мовами програмування: PHP, Python, Java, C#, Ruby тощо. Це дозволяє легко інтегрувати MySQL у веб- і десктопні додатки.
3. Індексація та оптимізація запитів. MySQL підтримує створення простих і складених індексів для прискорення пошуку даних. Також доступні

інструменти аналізу запитів, такі як `EXPLAIN`, що допомагають оптимізувати структури запитів і баз даних.

1. Автоматизація процесів. MySQL підтримує збережені процедури, тригери, події та планувальники завдань, що дозволяє зменшити кількість рутинної роботи та автоматизувати складні бізнес-процеси.
2. Масштабування та обробка великих обсягів даних

Завдяки горизонтальному та вертикальному масштабуванню, MySQL здатна обробляти великі бази даних і велику кількість одночасних підключень.

* 1. Переваги та обмеження MySQL

## Переваги:

* + - Простота встановлення та використання. Навіть новачки можуть швидко налаштувати та розпочати роботу з MySQL.
    - Швидкість роботи. Оптимізація системи дозволяє досягти високої продуктивності навіть на обладнанні середнього рівня.
    - Гнучкість конфігурації. Можливість налаштування під специфічні вимоги проєкту.
    - Розвинена екосистема. Велика спільнота користувачів, документація та підтримка[12].

## Обмеження:

* + - Відсутність деяких корпоративних функцій у безкоштовній версії.\*\* Наприклад, розширені інструменти для резервного копіювання або моніторингу.
    - Можливі проблеми з продуктивністю у великих транзакційних системах. У таких випадках необхідна додаткова оптимізація або використання інших СУБД.
    - Залежність від специфіки механізмів зберігання. Наприклад, MyISAM не підтримує транзакції, що обмежує її використання у певних сценаріях.
  1. Використання MySQL у практичних проєктах MySQL є основою для багатьох веб-додатків, таких як:
     + Системи управління контентом (CMS): WordPress, Drupal, Joomla.
     + електронна комерція: Magento, OpenCart.
     + соціальні мережі та форуми: phpBB, MediaWiki.

Завдяки своїй гнучкості та продуктивності, MySQL використовується у сферах:

* + - аналізу даних та побудови звітності.
    - розробки інтернет-магазинів та CRM-систем.
    - створення мобільних та хмарних сервісів.

Таким чином, MySQL - це універсальна система управління базами даних, яка успішно використовується в різних галузях. Його популярність обумовлена поєднанням продуктивності, простоти використання і великих функцій. Однак при обробці великих обсягів даних або жовтневих вимог може знадобитися додаткова оптимізація або інтеграція з іншими системами.

* 1. Клієнт HeidiSQL: функції, приклади застосування та переваги впровадження

HeidiSQL є одним із популярних і функціональних клієнтів для роботи з реляційними базами даних. Він відзначається своєю зручністю у використанні, широким спектром функцій і підтримкою багатьох баз даних, таких як MySQL, MariaDB, PostgreSQL, SQLite та Microsoft SQL Server.

Нижче ми розглянемо основні можливості клієнта HeidiSQL, а також особливості його застосування та переваги у контексті управління базами даних.

Загальний опис клієнта HeidiSQL

HeidiSQL — це безкоштовний інструмент для адміністрування баз даних із графічним інтерфейсом користувача (GUI). Його головна мета — забезпечити простий і зрозумілий спосіб взаємодії з базами даних, включаючи створення, модифікацію та управління таблицями, переглядом даних і виконанням SQL- запитів.

Цей клієнт розроблений німецьким програмістом Ансґаром Бекером і підтримується спільнотою ентузіастів.

HeidiSQL доступний для операційної системи Windows, що робить його зручним для широкого кола користувачів.

Основні функції та можливості

1. Підключення до різних серверів баз даних. HeidiSQL дозволяє встановлювати з'єднання з різними типами серверів баз даних. Підтримуються як локальні, так і віддалені сервери. Інструмент забезпечує просте налаштування підключення за допомогою вказівки хосту, порту, імені користувача та пароля.
2. Інтуїтивний інтерфейс. Інтерфейс HeidiSQL розроблений для того, щоб користувачі могли швидко переміщатися по деках своїх функцій. У головному вікні є вкладки для виконання запитів, перегляду структури баз даних і таблиць, а також для перегляду результатів виконання коду SQL.
3. Редагування та створення таблиць. HeidiSQL забезпечує інтуїтивний редактор для створення нових таблиць і редагування існуючих. Користувачі можуть налаштовувати типи даних, первинні ключі, індекси та інші параметри.
4. Виконання SQL-запитів. HeidiSQL дозволяє виконувати як прості, так і складні SQL-запити. Користувачі можуть зберігати запити, запускати їх у кількох сеансах та експортувати результати в різні формати.
5. Експорт і імпорт даних. Інструмент підтримує експорт даних у формати SQL, CSV, XML і HTML, що є корисним для резервного копіювання або перенесення баз даних. Також доступний зручний механізм імпорту даних із зовнішніх джерел.
6. Управління привілеями користувачів. HeidiSQL дозволяє адмініструвати доступ користувачів до баз даних, змінювати паролі та налаштовувати права доступу[10].

Застосування в практичній діяльності

HeidiSQL є корисним інструментом як для розробників, так і для системних адміністраторів. Він дозволяє швидко створювати прототипи баз даних, аналізувати структуру існуючих баз, оптимізувати запити та виконувати повсякденні адміністративні завдання.

Завдяки підтримці кількох типів баз даних, цей клієнт підходить для використання в проєктах різної складності.

Переваги використання HeidiSQL

1. Простота у використанні. Інтерфейс HeidiSQL є дружнім і зручним навіть для новачків. Це робить інструмент придатним для швидкого освоєння.
2. Багатофункціональність. HeidiSQL надає широкий спектр можливостей, які охоплюють майже всі аспекти управління базами даних.
3. Безкоштовність і відкритий доступ. HeidiSQL розповсюджується безкоштовно, що є великою перевагою для невеликих компаній і фрілансерів.
4. Швидкість роботи. Завдяки оптимізованому інтерфейсу й ефективній роботі з SQL-запитами, клієнт забезпечує високу швидкість виконання операцій.
5. Підтримка багатьох серверів. Можливість працювати з різними типами баз даних робить HeidiSQL універсальним інструментом[12].

HeidiSQL є зручним та функціональним інструментом у роботі з базами даних, що є досить доступним та легким для користувачів. Завдяки великій кількості функціональних можливостей, а підтримці різних типів серверів, цей клієнт став важливим інструментом у роботі фахівців із баз даних, сприяючи ефективності та якості їхньої роботи.

РОЗДІЛ 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ

* 1. Встановлення на комп’ютер необхідного програмного забезпечення

Для виконання практичної частини кваліфікаційної роботи, нам необхідно встановити сервер баз даних MySQL та клієнт HeidiSQL.

Для початку нам необхідно перейти на офіційний сайт MariaDB:

<https://mariadb.com/>[13].

Далі необхідно обрати операційну систему Windows і завантажити інсталяційний файл.

Потім запускаємо завантажений інсталяційний файл та проойдемо через майстра установки, обираючи необхідні пункти вибору(рис. 4.1).

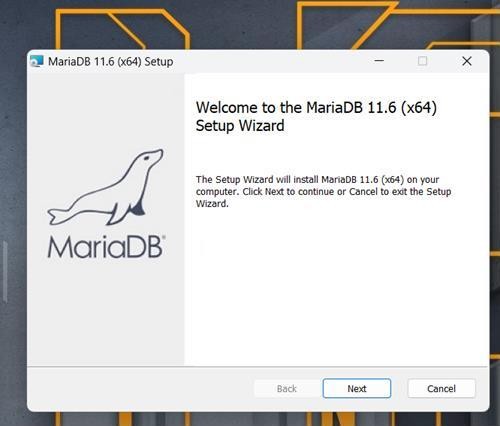


Рис. 4.1 Майстер установки MariaDB

На етапі конфігурації ми задаємо пароль для користувача root(рис. 4.2).

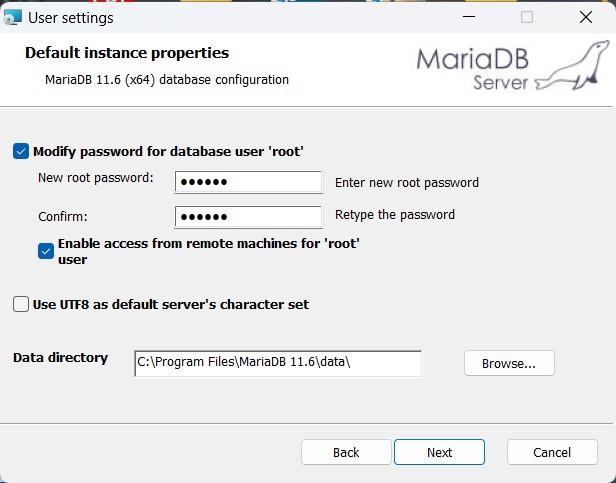


Рис. 4.2 Етап конфігурації

Після завершення установки необхідно переконатись, що MariaDB запущений.

* + 1. Встановлення HeidiSQL

Спочатку для завантаження HeidiSQL нам необхідно перейти офіційний сайт HeidiSQL: <https://www.heidisql.com/download.php>[14].

Наступним кроком є завантаження інсталяційного файлу для операційної системи Windows та його запуск. Далі нам необхідно пройти через майстра установки, вибираючи стандартні опції.

Для підключення до MySQL за допомогою HeidiSQL нам необхідно створимо нове підключення:

1. Запускаємо HeidiSQL.
2. Натискаємо "New" для створення нового підключення.
3. Введимо наступні дані:

Network type: MariaDB or MySQL (TCP/IP) Hostname / IP: 127.0.0.1

User: root

Password: Пароль, встановлений для користувача root

1. Натиснемо "Open" для підключення (рис. 4.3).

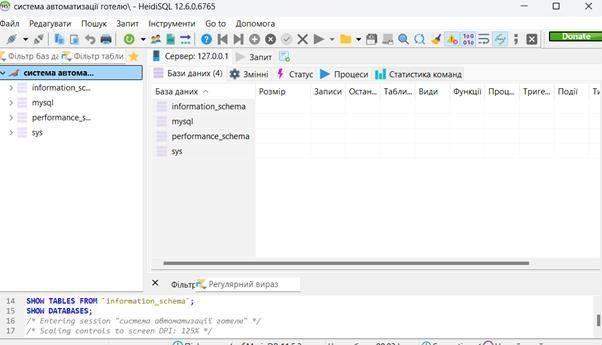


Рис. 4.2 Діалогове вікно HeidiSQL

* 1. Концептуальне моделювання даних

Концептуальне моделювання даних є фундаментальним етапом у процесі моделювання даних, який зосереджується на визначенні структури даних високого рівня в системі. Він слугує схемою, що окреслює критичні концепції та зв’язки, необхідні для розуміння вимог організації до даних. Цей етап є невід’ємною частиною забезпечення спільного розуміння всіма зацікавленими сторонами мети даних та їх взаємозв’язків у контексті бізнес-потреб.

ER-модель складається з:

* Сутностей: Об'єкти або поняття, що зберігаються в базі даних (наприклад, користувачі, продукти).
* Атрибутів: Властивості або характеристики сутностей (наприклад, ім'я користувача, ціна продукту).
* Зв'язків: Відношення між сутностями (наприклад, користувач робить замовлення).

Графічне представлення ER-моделі, що відображає сутності, атрибути та зв'язки між ними називається ER-діаграмою.

* 1. Побудова ER-діаграми

Для побудови ER-діаграми будемо використовувати сайт: <https://app.diagrams.net/>[15].

Використовуючи дані з пункту 4.2.1 маємо:



Рис. 4.1 ER-діаграма Таблиці мають такий вигляд:

1. Таблиця «Номери» містить такі дані: Id номера, Id готелю, категорію номеру, додаткові послуги, статус заселення.
2. Таблиця «Гості» містить такі дані: Id гостя, ім’я, прізвище, по-батькові, дата народження, номер телефону, номер паспорту.
3. Таблиця «Бронювання» містить такі дані: Id броні, Id гостя, Id кімнати, дата бронювання, дата поселення, дата виселення, кількість днів проживання.
4. Таблиця «Категорії номерів» містити такі дані: Id категорії, назва категорії, вартість).
5. Таблиця «Зручності» містить такі дані: Id додаткової послуги, назва послуги, вартість.
6. Таблиця «Статус бронювання» містить такі дані: Id статусу бронювання, наявність для бронювання.
7. Таблиця «Працівники» містить такі дані: Id працівника, прізвище, ім’я, по-батькові працівника, номер телефону, електронна пошта, посада.
8. Таблиця «Графік роботи» містить такі дані: Id запису в графіку, Id працівника, дата зміни, посада.

Опис реляційної моделі

Таблиця 4.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Customer  Відповідає типу сутності Гості | | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NOT NULL  /NULL | FK: ON DELETE  ON UPDATE | Пояснення (перехід від ER-моделі) |
| PK | **client\_id** | int(15) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **full\_name** | varchar(30) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **last\_name** | varchar(30) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **first\_name** | varchar(30) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **middle\_name** | varchar(30) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **email** | varchar(50) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **date\_of\_birth** | varchar(30) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **phone\_number** | varchar(30) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **cust\_password** | varchar(30) | NOT NULL |  | Атрибут простий, обов’язковий |

Таблиця 4.2 Бронювання

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Reservation  Відповідає типу сутності Бронювання | | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NOT NULL  /NULL | FK:  ON DELETE ON UPDATE | Пояснення (перехід від ER-моделі) |
| PK | **r\_id** | int(25) | NOT NULL |  | Атрибут простий, обов’язковий |
| FK | **cust\_id** | int(25) | NOT NULL |  | Атрибут простий, обов’язковий |
| FK | **booking\_date** | date | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **start\_date** | date | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **end\_date** | date | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **amount** | date | NULL |  | Атрибут простий, обов’язковий |
|  | **paid** | tinyint(1) | NOT NULL |  | Атрибут простий, обов’язковий |

Таблиця 4.3 Готель

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Hotel  Відповідає типу сутності Готель | | | | | |
| Кл юч | Ім’я атрибуту | Тип атрибуту | NOT NULL  /NULL | FK:  ON DELETE ON UPDATE | Пояснення (перехід від ER-моделі) |
|  | **hotel\_id** | int(12) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **hotel\_name** | int(12) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **hotel\_ address** | varchar(20) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **hotel\_email** | decimal(5,2) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **hotel\_phone** | int(11) | NOT NULL |  | Атрибут простий, обов’язковий |

Таблиця 4.4 Статус бронювання

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Room\_status  Відповідає типу сутності Статус бронювання | | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NOT NULL  /NULL | FK: ON DELETE ON UPDATE | Пояснення (перехід від ER-моделі) |
| PK | **type\_id** | int(15) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **availability** | varchar(30) | NOT NULL |  | Атрибут простий, обов’язковий |

Таблиця 4.5 Тип номеру

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Room\_type  Відповідає типу сутності Тип номеру | | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NOT NULL  /NULL | FK: ON DELETE ON UPDATE | Пояснення (перехід від ER-моделі) |
| PK | **status\_id** | int(15) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **type\_name** | varchar(30) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **cost** | Float | NULL |  | Атрибут простий, обов’язковий |

Таблиця 4.6 Зручності готелю

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Facilities  Відповідає типу сутності Зручності готелю | | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NOT NULL  /NULL | FK: ON DELETE ON UPDATE | Пояснення (перехід від ER-моделі) |
| PK | **status\_id** | int(15) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **type\_name** | varchar(30) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **cost** | Float | NULL |  | Атрибут простий, обов’язковий |

Таблиця 4.7 Працівники

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Employees  Відповідає типу сутності Працівники | | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NOT NULL  /NULL | FK: ON DELETE ON UPDATE | Пояснення (перехід від ER-моделі) |
| PK | **e\_id** | int(15) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **hotel\_id** | varchar(30) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **full\_name** | varchar(50) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **last\_name** | varchar(50) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **first\_name** | varchar(50) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **middle\_name** | varchar(50) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **date\_of\_birth** | varchar(50) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **phone\_number** | varchar(25) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **email** | varchar(35) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **e\_salary** | varchar(35) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **password** | varchar(35) | NOT NULL |  | Атрибут простий, обов’язковий |

Таблиця 4.8 Графік персоналу

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Room\_type  Відповідає типу сутності Робочому графіку персоналу | | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NOT NULL  /NULL | FK: ON DELETE ON UPDATE | Пояснення (перехід від ER-моделі) |
| PK | **schedule\_id** | int(50) | NOT NULL |  | Атрибут простий, обов’язковий |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **employee\_id** | int(50) | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **work\_date** | DATE | NOT NULL |  | Атрибут простий, обов’язковий |
|  | **position** | varchar(30) | NOT NULL |  | Атрибут простий, обов’язковий |

* 1. Реалізація запитів та збережених процедур в HeidiSQL
     1. Створення запитів

Запити в HeidiSQL стосуються процесу написання та виконання запитів на мові SQL, взаємодія з базами даних здійснюється через графічний інтерфейс користувача шляхом виконання команд SQL для вибору, оновлення, видалення або вставки даних.

Нижче будуть наведені запити, що були розроблені для більш ефективного управління номерним фондом готелю та графіком персоналу.

Ці графіки допоможуть нам забезпечити повний контроль над наявність вільних і заселених номерів, виконання бронювання номерів та контроль над графіком персоналу підприємства.

* + - 1. Список всіх броней з інформацією про гостей готелю та номери Цей запит надасть інформацію про всі замовлення разом з деталями про

гостей і номери кімнат, які вони забронювали, а також, коли була дата бронювання, яка буде дата заселення і виселення, та на яку кількість днів гості зупиняться в готелі.

Це допоможе вірно в рівномірному заселенні номерів в готелі та допоможе уникнути повторного заселення одного того готельного номеру і овербукінгу, і в загальному даний запит відображає ситуацію по бронюванням в готелі.

Запит на мові SQL:

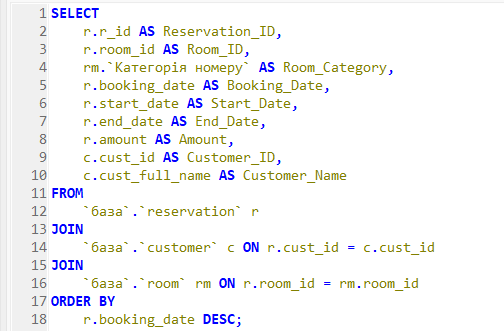


Рисунок 4.1 – Запит списку бронювання номерів



Рисунок 4.2 – Результат виконання запиту «Список всіх броней з інформацією про гостей готелю та номери» в HeidiSQL

* + - 1. Створення запиту на пошук працівників, які працюють по певним

дням

Цей запит надасть інформацію про працівників, що працюють у

визначений день, а саме 12.12.2024 у наведеному прикладі.

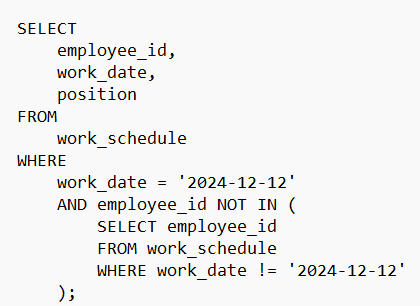


Рисунок 4.3–Запит списку працівників, що працюють у визначений день

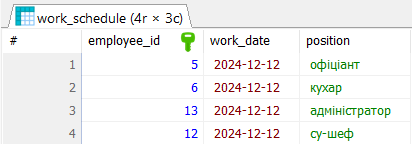


Рисунок 4.4 – Результат виконання запиту «Список працівників, що працюють у визначений день, а саме 12.12.2024» в HeidiSQL

* + - 1. Створення запиту на пошук заселених номерів на визначену дату Цей запит надасть інформацію номери, що що є заселеними у визначений

день, а саме 12.12.2024 у наведеному прикладі, і на коли запланований виїзд з даних номерів.

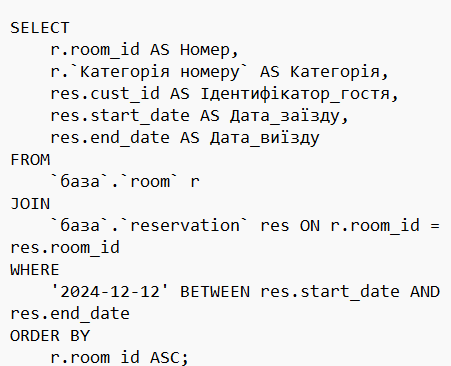


Рисунок 4.5 Запит списку заселених номерів у визначений день

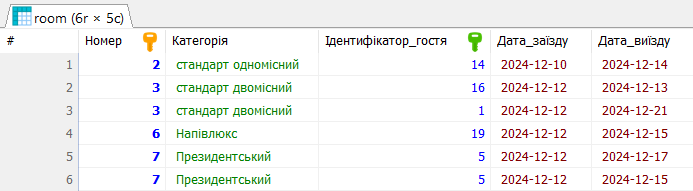


Рисунок 4.6– Результат виконання запиту «Список заселених номерів у визначений день, а саме 12.12.2024» в HeidiSQL

* + - 1. Створення запиту на пошук посад всіх працівників Цей запит надасть інформацію про посади всіх працівників готелю.

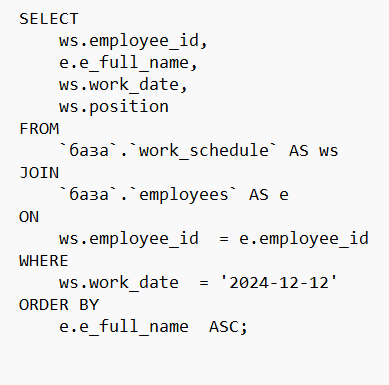


Рисунок 4.7 Запит посад всіх працівників готелю

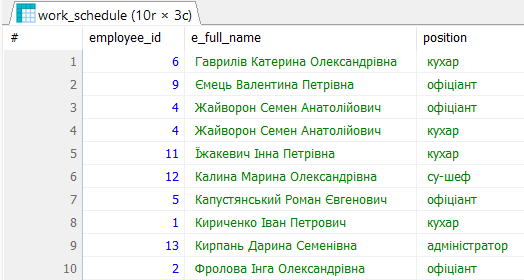


Рисунок 4.8– Результат виконання запиту «Список посад працівників готелю» в HeidiSQL

* + - 1. Створення запиту на пошук електронних адрес адміністраторів готелю

Цей запит надасть інформацію електронні пошти адміністраторів готелю.



Рисунок 4.9 Запит пошук електронних адрес адміністраторів готелю

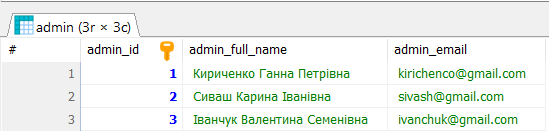


Рисунок 4.10– Результат виконання запиту «Список електронних адрес адміністраторів готелю» в HeidiSQL

* + - 1. Створення запиту на пошук вільних номерів в готелі на визначену

дату

Цей запит надасть інформацію про не заселені номери в готелі, на

визначені дати

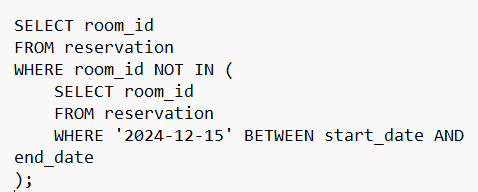


Рисунок 4.11 Запит пошук на пошук вільних номерів в готелі на визначену дату

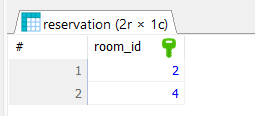


Рисунок 4.12– Результат виконання запиту «Список на пошук вільних номерів в готелі на визначену дату» в HeidiSQL

* + - 1. Створення запиту на пошук номера телефона певного гостя Цей запит надасть інформацію про номер телефона певного гостя.

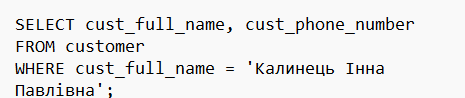


Рисунок 4.13 Запит пошук номера телефону гостя

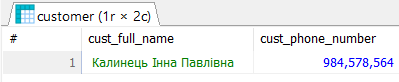


Рисунок 4.14– Результат виконання запиту «Список на пошук номера телефону гостя» в HeidiSQL

* + - 1. Переоцінка цін вартості номерів з урахування вартості коефіцієнту інфляції

В сучасних економічних умовах, переоцінка вартості послуг з урахуванням коефіцієнту інфляції є досить важливим інструментом для підприємства готельного господарства. У зв’язку з тим що інфляція зменшує купівельну здатність підприємства і провокує до збільшення витрат, тому вкрай важливо періодично робити переоцінку для підтримання реалістичної вартості послуг.

Щоб здійснити переоцінку товарів і послуг підприємства, необхідно вартість послуги перемножити на коефіцієнт інфляції і додати умову заокруглення до десятків.

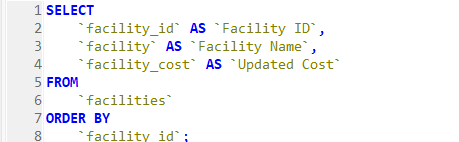


Рисунок 4.15 Запит для переоцінка цін вартості додаткових послуг з урахування вартості коефіцієнту інфляції

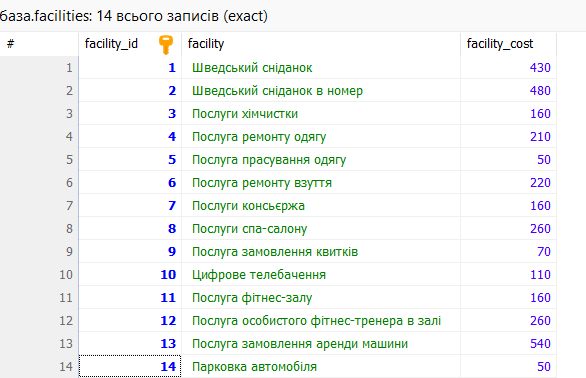


Рисунок 4.16– Результат виконання запиту для переоцінка цін вартості додаткових послуг в HeidiSQL

ВИСНОВКИ

Під час написання дипломної роботи було розглянути такі теоретичні питання:

* поняття готельного господарства;
* поняття про системи автоматизації готельних підприємств і їх різновиди, порівняння найпопулярніших систем на ринку;
* розглянуто бази даних, як невід’ємну складову майбутніх систем автоматизації;
* автоматизація готелів за допомогою баз даних;
* характеристика серверу MySQL та клієнта HeidiSQL.

В результаті аналізу було визначено основні функціональні вимоги до проектованої бази даних готельного підприємства. Для виконання дипломної було використано клієнт HeidiSQL та сервер MariaDB. Практична складова диплому містить:

* розробку бази даних;
* створення групи SQL-запитів, а саме знаходження списку всіх бронювань з інформацією про гостей готелю та номери та переоцінка вартості з урахуванням показнику інфляції.

В дипломній роботі розглянуто, як процес автоматизації готельного підприємства за допомогою бази даних сприятиме наданню більш якісних послуг, через контроль всіх процесу бронювання, за допомогою внесення додаткової інформації про вподобання постійних гостей, що в підсумку буде покращувати досвід їх проживання.

Також завдяки чіткому контролю за графіком працівників можна більш якісно розподілити трудове навантаження між працівниками підприємства, що сприятиме більш якісному виконанню ними своїх обов’язків.

Перерахунок вартості за послугу з врахуванням інфляції сприятиме фінансовій стабільності підприємства на ринку, тому що будуть мінімізуватись втрати коштів в зв’язку з постійними інфляційними процесами.

Отже, власне дослідження вкотре доводить важливість впровадження сучасних баз даних в роботу готельних підприємств, що сприятиме зменшенню витрат часу на дрібну рутинну роботу, постійному контролю за зміною показників готелю, сприятиме зменшенню витрат та оптимізації цін на послуги, і в загальному допоможе покращити показники роботи підприємства та підвищити рівень задоволеності послугами гостей

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ананьєв, О.М. Інформаційні системи і технології в комерційній діяльності [Текст]: підручник / О.М. Ананьєв, В.М. Білик, Я.А. Гончарук. - Львів: Новий Світ-2000, 2006. - 584 с.
2. Апонін В.В., Олексин І.І., Шутовська Н.О., Футало Т.В. Організація і технологія надання послуг. Навч. посібник за редакцією професора В.В. Апоніна. - К.: Видавничий центр "Академія", 2006. -311 с.
3. Бойко М.Г., Гопкало Л.М. Організація готельного господарства: Підручник. - К.: Київ. нац. торг.-екон. ун-т., 2006, 448 с.
4. Бойцова М., Піроженко О. Усе про облік та організацію готельного бізнесу. - Харків: Фактор, 2005. - 232 с.
5. Будя О. Напрями та завдання розвитку інформаційних технологій у туризмі та готельному бізнесі України. - 2001. -№ 1. - ст.46-47.
6. Катренко А.В. Системний аналіз: підручник з грифом МОН / Катренко А.В. – Львів: Магнолія-2006, 2009.
7. Лахно А. «Характеристика систем автоматизації готелей» // «Вісник студентського наукового товариства», випуск 30, ст.112.
8. Лахно А. Переваги впровадження системи автоматизації готельного підприємства // Матеріали Всеукраїнської науково-практичної конференції

«Математичні, природничі та комп’ютерні науки та науки про управління, технології, навчання: науково-практичні рішення та підходи молодих науковців», 2024.

1. Литвин В.В. Інтелектуальні системи: підручник / Литвин В.В., Пасічник В.В., Яцишин Ю. В. — Львів: Новий світ-2000, 2009. – 406 с.
2. Пасічник В. В. Організація баз даних та знань / В.В. Пасічник, В.А. Резніченко. – К.: Видавнича група BHV, 2006. – 384 с.
3. Теоретико-методичні засади та прикладні механізми регулювання розвитку підприємств торгівлі: монографія / за ред. П. О. Куцика, Т. Г. Васильцива. Львів : Видавництво ЛТЕУ, 2016. 426 с.
4. Черняк, О.І. Системи обробки економічної інформації [Текст]: підручник / О.І. Черняк, А.В. Ставицький, Г.О. Чорноус. - К.:Знання, 2006-447 с.
5. MariaDB

URL: https://mariadb.org/uk/ (дата звернення: 01.10.2024).

1. Download HeidiSQL.

URL: https://[www.heidisql.com/download.php](http://www.heidisql.com/download.php) (дата звернення: 01.10.2024).

1. draw.io

URL: https://app.diagrams.net/(дата звернення: 01.10.2024).

ДОДАТОК А

SQL-файл бази даних для готелю

**CREATE TABLE** `customer` (

`cust\_id` **INT**(15) **NOT NULL AUTO\_INCREMENT**,

`cust\_full\_name` **VARCHAR**(50) **NOT NULL COLLATE**

'cp1251\_ukrainian\_ci',

`cust\_last\_name` **VARCHAR**(30) **NOT NULL COLLATE**

'cp1251\_ukrainian\_ci',

`cust\_first\_name` **VARCHAR**(30) **NOT NULL COLLATE**

'cp1251\_ukrainian\_ci',

`cust\_middle\_name` **VARCHAR**(30) **NOT NULL COLLATE**

'cp1251\_ukrainian\_ci',

`cust\_email` **VARCHAR**(50) **NOT NULL COLLATE**

'cp1251\_ukrainian\_ci',

`cust\_date\_of\_birth` **VARCHAR**(30) **NOT NULL COLLATE** 'armscii8\_bin',

`cust\_phone\_number` **INT**(30) **NOT NULL**,

`cust\_password` **VARCHAR**(30) **NOT NULL COLLATE**

'armscii8\_bin',

**PRIMARY KEY** (`cust\_id`) **USING BTREE**)

**COLLATE**='latin1\_swedish\_ci' **ENGINE**=**InnoDB AUTO\_INCREMENT**=96

**CREATE TABLE** `employees` (

`employee\_id` **INT**(11) **NOT NULL AUTO\_INCREMENT**,

`hotel\_id` **INT**(20) **NULL DEFAULT NULL**,

`e\_full\_name` **VARCHAR**(35) **NOT NULL COLLATE**

'utf8mb4\_unicode\_ci',

`last\_name` **VARCHAR**(35) **NOT NULL COLLATE**

'utf8mb4\_unicode\_ci',

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| `first\_name` | **VARCHAR**(35) | **NOT** | **NULL** | **COLLATE** |
| 'utf8mb4\_unicode\_ci', |  |  |  |  |
| `middle\_name` | **VARCHAR**(35) | **NOT** | **NULL** | **COLLATE** |

'utf8mb4\_unicode\_ci',

`e\_phone\_number` **VARCHAR**(50) **NOT NULL DEFAULT** ''

**COLLATE** 'utf8mb4\_unicode\_ci',

`email` **VARCHAR**(20) **NOT NULL COLLATE**

'utf8mb4\_unicode\_ci',

`e\_salary` **VARCHAR**(50) **NOT NULL DEFAULT** ''

**COLLATE** 'utf8mb4\_unicode\_ci',

**PRIMARY KEY** (`employee\_id`) **USING BTREE**, **INDEX** `hotel\_id` (`hotel\_id`) **USING BTREE**,

**CONSTRAINT** `employees\_ibfk\_1` **FOREIGN KEY** (`hotel\_id`) **REFERENCES** `hotel` (`hotel\_id`) **ON UPDATE RESTRICT ON DELETE CASCADE**

)

**COLLATE**='utf8mb4\_unicode\_ci' **ENGINE**=**InnoDB AUTO\_INCREMENT**=14;

**CREATE TABLE** `hotel` (

`hotel\_id` **INT**(11) **NOT NULL AUTO\_INCREMENT**,

`hotel\_name` **VARCHAR**(45) **NOT NULL COLLATE**

'cp1251\_ukrainian\_ci',

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| `hotel\_addr` | **VARCHAR**(130) | **NOT** | **NULL** | **COLLATE** |
| 'cp1251\_ukrainian\_ci',  `hotel\_email` | **VARCHAR**(30) | **NOT** | **NULL** | **COLLATE** |

'latin1\_swedish\_ci',

`hotel\_phone` **VARCHAR**(25) **NOT NULL COLLATE**

'utf8mb4\_unicode\_ci',

**PRIMARY KEY** (`hotel\_id`) **USING BTREE**

)

**COLLATE**='latin1\_swedish\_ci' **ENGINE**=**InnoDB AUTO\_INCREMENT**=2;

**CREATE TABLE** `reservation` (

`r\_id` **INT**(25) **NOT NULL AUTO\_INCREMENT**,

`cust\_id` **INT**(25) **NULL DEFAULT NULL**,

`room\_id` **INT**(25) **NULL DEFAULT NULL**,

`booking\_date` **DATE NOT NULL**,

`start\_date` **DATE NOT NULL**,

`end\_date` **DATE NOT NULL**,

`amount` **FLOAT NOT NULL**,

**PRIMARY KEY** (`r\_id`) **USING BTREE**,

**INDEX** `cust\_id` (`cust\_id`) **USING BTREE**, **INDEX** `room\_id` (`room\_id`) **USING BTREE**,

**CONSTRAINT** `reservation\_ibfk\_1` **FOREIGN KEY** (`cust\_id`) **REFERENCES** `customer` (`cust\_id`) **ON UPDATE RESTRICT ON DELETE CASCADE**,

**CONSTRAINT** `reservation\_ibfk\_2` **FOREIGN KEY** (`room\_id`) **REFERENCES** `room` (`room\_id`) **ON UPDATE RESTRICT ON DELETE CASCADE**

)

**COLLATE**='latin1\_swedish\_ci' **ENGINE**=**InnoDB AUTO\_INCREMENT**=22;

**CREATE TABLE** `room` (

`room\_id` **INT**(20) **NOT NULL AUTO\_INCREMENT**,

`type\_id` **INT**(50) **NULL DEFAULT NULL**,

`facility\_id` **INT**(11) **NULL DEFAULT NULL**,

`status\_id` **INT**(25) **NULL DEFAULT NULL**,

`hotel\_id` **INT**(20) **NULL DEFAULT NULL**,

`Категорія номеру` **VARCHAR**(3000) **NOT NULL COLLATE** 'cp1251\_ukrainian\_ci',

**PRIMARY KEY** (`room\_id`) **USING BTREE**,

**INDEX** `hotel\_id` (`hotel\_id`) **USING BTREE**, **INDEX** `type\_id` (`type\_id`) **USING BTREE**,

**INDEX** `facility\_id` (`facility\_id`) **USING BTREE**, **INDEX** `status\_id` (`status\_id`) **USING BTREE**,

**CONSTRAINT** `room\_ibfk\_1` **FOREIGN KEY** (`hotel\_id`) **REFERENCES** `hotel` (`hotel\_id`) **ON UPDATE RESTRICT ON DELETE CASCADE**,

**CONSTRAINT** `room\_ibfk\_2` **FOREIGN KEY** (`type\_id`) **REFERENCES** `room\_type` (`type\_id`) **ON UPDATE RESTRICT ON DELETE CASCADE**,

**CONSTRAINT** `room\_ibfk\_3` **FOREIGN KEY** (`facility\_id`) **REFERENCES** `facilities` (`facility\_id`) **ON UPDATE RESTRICT ON DELETE CASCADE**,

**CONSTRAINT** `room\_ibfk\_4` **FOREIGN KEY** (`status\_id`) **REFERENCES** `room\_status` (`status\_id`) **ON UPDATE RESTRICT ON DELETE CASCADE**

)

**COLLATE**='latin1\_swedish\_ci' **ENGINE**=**InnoDB AUTO\_INCREMENT**=25;

**CREATE TABLE** `room\_status` (

`status\_id` **INT**(15) **NOT NULL AUTO\_INCREMENT**,

`availability` **VARCHAR**(30) **NULL DEFAULT NULL COLLATE** 'latin1\_swedish\_ci',

**PRIMARY KEY** (`status\_id`) **USING BTREE**

)

**COLLATE**='latin1\_swedish\_ci' **ENGINE**=**InnoDB AUTO\_INCREMENT**=4;

**CREATE TABLE** `room\_type` (

`type\_id` **INT**(11) **NOT NULL AUTO\_INCREMENT**,

`type\_name` **VARCHAR**(50) **NOT NULL COLLATE**

'cp1251\_ukrainian\_ci',

`cost` **FLOAT NULL DEFAULT NULL**, **PRIMARY KEY** (`type\_id`) **USING BTREE**

)

**COLLATE**='latin1\_swedish\_ci' **ENGINE**=**InnoDB AUTO\_INCREMENT**=8;

**CREATE TABLE** `work\_schedule` (

`schedule\_id` **INT**(50) **NOT NULL AUTO\_INCREMENT**,

`employee\_id` **INT**(50) **NOT NULL**,

`work\_date` **DATE NOT NULL**,

`position` **VARCHAR**(50) **NOT NULL COLLATE**

'utf8mb4\_unicode\_ci',

**PRIMARY KEY** (`schedule\_id`) **USING BTREE**,

**INDEX** `FK\_work\_schedule\_employees` (`employee\_id`) **USING BTREE**,

**CONSTRAINT** `FK\_work\_schedule\_employees` **FOREIGN KEY** (`employee\_id`) **REFERENCES** `employees` (`employee\_id`) **ON UPDATE NO ACTION ON DELETE NO ACTION**

)

**COLLATE**='utf8mb4\_unicode\_ci' **ENGINE**=**InnoDB AUTO\_INCREMENT**=4

;