Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

(повна назва)

Кафедра Інформатика

(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

до курсової роботи

з дисципліни “ Організація баз даних та знань ”

за темою

«Розробка інформаційної системи «Автосалон» з використанням СУБД MICROSOFT SQL SERVER 2019 та мови запитів SQL»

Виконав:

студент 2 курсу, групи ІНФ-19-1

Єременко І.О

(прізвище, ініціали)

Спеціальності 122 Комп’ютерні науки

(код і повна назва спеціальності)

Освітня програма Інформатика

(повна назва освітньої програми)

Керівник Яковлева О.В.

(посада, прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

(найменування вищого учбового закладу)

Факультет Інформаційно-аналітичних технологій та менеджменту

(повна назва)

Кафедра Інформатики

(повна назва)

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп’ютерні науки

(код і повна назва)

Освітня програма Інформатика

(повна назва освітньої програми)

Дисципліна Організація баз даних та знань

(повна назва освітньої програми)

Курс 2 Группа ІТІНФ-19-1 Семестр 3 .

ЗАВДАННЯ

на курсовий проект (роботу) студента

Єременко Іван Олексійович

(прізвище, ім’я, по батькові)

1. Тема проекту (роботи) «Розробка інформаційної системи «Автосалон» з використанням СУБД MICROSOFT SQL SERVER 2019 та мови запитів SQL»   
2. Строк здачі студентом закінченого проекту (роботи) 27.12.2020

3. Вихідні данні проекту (роботи) *мова запитів SQL, СУБД MS SQL Server, середовище MS SQL Server Management Studio, CASE – засіб візуального проектування даних Erwin, середовище розробки програмних систем MS Visual Studio, мова є програмування С#, відомості про предметну область « Автосалон»*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)1) Принципи проектування та реалізації реляційних баз даних 2) Проектування бази даних для ПО «Автосалон»

3) Проектування та розробка ІС для предметної області «Автосалон»

4) Тестування ІС

5. Перелік графічного матеріала (з точним зазначенням обов’язкових креслень).

1) Схема даних ПО «Автосалон» у синтаксисі Чена

2) Схема даних ПО «Автосалон» з використанням CASE-засіба Erwin

3) UML-діаграма

4) Діаграма взаємодії користувачів і ІС. Use-case діаграма

6. Дата видачі завдання 20.10.20

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| № | Найменування етапів курсового проекту | Термін виконання етапів проекту | Примітки |
| 1 | *Отримання завдання* | *20.10.2020* | *Виконано* |
| 2 | *Вивчення теоретичного матеріалу, ознайомлення з програмним забезпеченням* | *20.10.20 – 30.10.20* | *Виконано* |
| 3 | *Розробка структури РБД* | *31.10.20-20.11.20* | *Виконано* |
| 4 | *Розробка функціональності ІС* | *20.11.20-1.12.20* | *Виконано* |
| 5 | *Тестування ІС* | *2.12.20-10.12.20* | *Виконано* |
| 6 | *Оформлення пояснювальної записки* | *11.12.20-26.12.20* | *Виконано* |
| 7 | *Захист курсової роботи* | *27.12.20* | *Виконано* |

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (підпис)

**РЕФЕРАТ/ABSTRACT**

Пояснювальна записка: с., табл., рис., дод., джерел.

ІНФОРМАЦІЙНА СИСТЕМА, СУБД, РЕЛЯЦІЙНА БАЗА ДАНИХ, СУТНІСТЬ, НОРМАЛІЗАЦІЯ ДАНИХ, ER-МОДЕЛЬ, МОВА ЗАПИТІВ SQL, CASE – ЗАСІБ ERWIN, MS SQL SERVER, MS SQL SERVER MANAGEMENT STUDIO, MS VISUAL STUDIO 2019, C#, ENTITY FRAMEWORK

Робота присвячена розробці інформаційної системи «Автосалон». Дана інформаційна система призначена для зберігання, обробки архіву угод, легшого пошуку і ознайомлення клієнта з моделями, які є в наявності, зберігання інформації про завезення товарів на склад і тд. Також в базі даних інформаційної системи зберігаються користувачі(клієнт, менеджер, адміністратор, постачальник), які мають певну свободу дій в рамках ІС.

Під час розробки інформаційної системи були використані: теорія проектування реляційних баз даних на базі будування ER- моделі; основні принципи нормалізації реляційних баз даних; мова запитів SQL; CASE- засіб візуального проектування даних Erwin, середовище розробки програмних систем MS Visual Studio, мова програмування C#; мова запитів до джерела бази даних LINQ; об'єктно-орієнтована технологія доступу до даних ADO.NET Entity Framework.

Explanatory note: p., tab., figure., dop., sources.

INFORMATION SYSTEM, DMS, RELATIONAL DATABASE, ENTITY, DATA NORMALIZATION, ER-MODEL, SQL QUERY LANGUAGE, CASE - ERWIN TOOL, MS SQL SERVER, MS SQL SERVER MANAGEMENT STUDIO, MS VISUAL STUDIO 2019, C#, ENTITY FRAMEWORK

The work is devoted to the development of the information system "Auto Show". This information system is designed to store, process the archive of transactions, easier search and acquaintance of the client with the available models, storage of information about the delivery of goods to the warehouse, etc. Also in the database of the information system are stored users (customer, manager, administrator, supplier), who have some freedom of action within the IP.

During the development of the information system were used: the theory of relational database design based on the construction of the ER-model; basic principles of normalization of relational databases; SQL query language; CASE - Erwin visual data design tool, MS Visual Studio software development environment, C # programming language; language of queries to the LINQ database source; object-oriented data access technology ADO.NET Entity Framework.

ЗМІСТ

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 7](#_Toc59891402)

[ВСТУП 8](#_Toc59891403)

[1. ПРИНЦИПИ ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЇ РЕЛЯЦІЙНИХ БАЗ ДАНИХ 9](#_Toc59891404)

[1.1 Основні етапи життєвого циклу ІС та підході до керування ІТ проектами 9](#_Toc59891405)

[1.1.1 Класичне управління 9](#_Toc59891406)

[1.1.2 Основні етапи життєвого циклу ІС 10](#_Toc59891407)

[1.2 Основні властивості реляційних баз даних 11](#_Toc59891408)

[1.3 Можливості мови SQL 12](#_Toc59891409)

[1.3.1 Загальні відомості про SQL 12](#_Toc59891410)

[1.3.2. Оператори SQL 13](#_Toc59891411)

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІС – Інформаційна Система

MS – Microsoft

VS – Visual Studio

ІН – ідентифікаційний номер

UML – Unified Modeling Language

СУБД – система управління базами даних

ER – Entity Relationship

SQL – Structured Query Language

ЖЦ – Життєвий цикл

ІТ – Інформаційні технології

НФ – Нормальна форма

ВСТУП

В даній курсовій роботі була розроблена інформаційна система, яка надає користувачам зручний інтерфейс, за допомогою якого вони можуть працювати з даними, які зберігаються в базі даних певним чином. Тобто так, щоб користувачі не змогли внести небажані зміни в базу даних і заносили тільки коректні дані.

В процесі роботи впевнився, що низхідний підхід для проектування схеми даних, а потім перевірка результату низхідного підходу висхідним, а саме за допомогою методу нормалізації дуже допомагає в розробці подібних інформаційних систем.

Головна мета нормалізації бази даних – усунення надмірності та дублювання інформації. В ідеалі при нормалізації треба домогтися, щоб будь-яке значення в базі даних зберігалося в одному екземплярі, причому значення це має бути отримано розрахунковим шляхом з інших даних що зберігаються в базі.

1. ПРИНЦИПИ ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЇ РЕЛЯЦІЙНИХ БАЗ ДАНИХ
   1. Основні етапи життєвого циклу ІС та підході до керування ІТ проектами

### 1.1.1 Класичне управління

У людства за всю історію накопичився значний список успішно реалізованих складних проектів. Від будівництва пірамід у Гізі до відправки людини на Місяць, найсміливіші людські починання вимагали злагодженої роботи тисяч людей. А це має на увазі складну систему управління проектами.

Найбільш очевидний спосіб зробити свій проект більш керованим – це розбити процес його виконання на послідовні етапи. Саме на такій лінійній структурі базується традиційне проектне управління. У цьому сенсі вона нагадує комп’ютерну гру – не можна перейти на наступний рівень не завершивши попередній.

Даний підхід орієнтований на проекти, в яких є строгі обмеження по послідовності виконання завдань. Наприклад, будівництво будинку – не можна зводити стіни без фундаменту.

Великим плюсом даного підходу є те, що він вимагає від замовника і керівництва компанії визначити, що ж вони хочуть отримати, вже на першому етапі проекту. Це привносить певну стабільність в роботу проекту, а планування дозволяє впорядкувати реалізацію проекту. Крім того, цей підхід має на увазі моніторинг показників і тестування, що абсолютно необхідно для реальних проектів різного масштабу. Також, класичний підхід дозволяє уникнути стресів через наявність запасного часу на кожному етапі, закладеного на випадок будь-яких ускладнень і реалізації ризиків. З правильно проведеним етапом планування, керівник проектів завжди знає, якими ресурсами він володіє.

Основна слабка сторона класичного підходу – нетолерантність до змін.

### 1.1.2 Основні етапи життєвого циклу ІС

1.Постановка задачі

На даному етапі описується ідея проекту і перетворюється в чітко поставлену задачу, яка містить в собі всі потрібні аспекти і на виході видає очікуваний результат. Також, ідея може розділятися на декілька задач, які при виконанні приведуть к результату, задуманому спочатку

2.Проектування

На даному етапі команда вирішує, як вона буде досягати мети, поставленої на попередньому етапі. Команда уточнює, деталізує та візуалізує мету і результати проекту. За допомогою цієї інформації формує календарний план, бюджет і оцінує риски.

3.Кодування

На даному етапі починається реалізація поставленої задачі, згідно з планом, якому розробники повинні чітко притримуватись. Також на цьому етапі повинні бути більш жорсткі дедлайни, щоб розробники мали більш конкретний орієнтир

4.Тестування і усунення багів

Коли команда має майже готову інформаційну систему – вона піддається різноманітним тестуванням, в тому числі не тільки на очевидні і заплановані випадки, а коли користувач буде вести себе непередбачувано – програма не повинна руйнуватися. Також система перевіряється на критичне навантаження, тобто об’єм інформації, який може припинити коректну роботу системи.

5.Експлуатація

Після успішного проходження тестів і установлення критичних навантажень – інформаційна система запускається для масового користування

## 1.2 Основні властивості реляційних баз даних

База даних – сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв’язки між їх елементами. Основні типи організації даних: реляційний, мережевий, ієрархічний.

Реляційна БД - це тип бази даних, що зберігає інформацію в таблицях і здійснює пошук даних в одній таблиці на підставі визначених ключових полів іншої таблиці. Простіше кажучи, це двовимірна таблиця, що підпорядковується деяким властивостям.

Властивості реляційної бази даних:

1. Порядок розташування кортежів у відношенні не має значення.

2. Порядок розташування атрибутів у відношенні не має значення.

3. Кожен кортеж є унікальним.

4. Кожен атрибут має унікальне ім'я.

5. Значення атрибута знаходяться в одному домені.

6. Кожна комірка належить одному домену.

7. Ім'я відношення є унікальним в реляційній схемі.

Складові частини реляційної бази даних за Дейтом:

1.Структурна частина моделі визначає, що єдиною структурою даних є нормалізоване n-арне відношення. Відношення зручно представляти у формі таблиць, де кожен рядок є кортеж, а кожен стовпець — атрибут, визначений на деякому домені. Даний неформальний підхід до поняття відношення дає більш звичну для розробників і користувачів форму представлення, де реляційна база даних подається як кінцевий набір таблиць.

2.Маніпуляційна частина моделі визначає два фундаментальних механізми маніпулювання даними — реляційну алгебру і реляційне числення. Основною функцією маніпуляційної частини реляційної моделі є забезпечення заходів реляційності будь-якої конкретної мови реляційних БД: мова називається реляційною, якщо вона має не меншу виразність і потужність, ніж реляційна алгебра або реляційне числення.

3.Цілісна частина моделі визначає вимоги цілісності сутностей і цілісності посилань. Перша вимога полягає в тому, що будь-який кортеж будь-якого відношення відмінний від будь-якого іншого кортежу цього відношення, тобто іншими словами, будь-яке відношення має володіти первинним ключем. Вимога цілісності щодо посилань, або вимога зовнішнього ключа полягає в тому, що для кожного значення зовнішнього ключа, що з'являється у відношенні, на яке веде посилання, повинен знайтися кортеж з таким же значенням первинного ключа, або значення зовнішнього ключа повинно бути невизначеним (тобто ні на що не вказувати).

## 1.3 Можливості мови SQL

### 1.3.1 Загальні відомості про SQL

SQL (Structured query language) — декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації.

На початку 70-х рр. в компанії IBM була розроблена експериментальна СУБД System R на основі мови SEQUEL, який можна вважати безпосереднім попередником SQL. Метою розробки було створення простого непроцедурного мови, яким міг скористатися будь-який користувач, який навіть не має навичок програмування.

Зауважимо, що, хоча SQL і замислювався як засіб роботи кінцевого користувача, врешті-решт, він став настільки складним, що перетворився в інструмент програміста

SQL є стандартною мовою для системи реляційних БД. Всі реляційні СУБД, такі як MySQL, MS Access, Oracle, Sybase, Informix, Postgres і SQL Server, використовують SQL в якості стандартної мови для взаємодії з БД.

Виділяються два різновиди (або діалекти) мови SQL: PL-SQL і T-SQL. PL-SQL використовується в таких СУБД як Oracle і MySQL, в той час як T-SQL (Transact-SQL) застосовується в SQL Server.

### 1.3.2 Оператори SQL

Виділяють такі види SQL запитів:

1. DDL (Data Definition Language) - мова визначення даних. Завданням DDL запитів є створення БД і опис її структури. Запитами такого виду встановлюються правила того, в якому вигляді різні дані будуть розміщуватися в БД.

* CREATE - використовується для створення об'єктів бази даних;
* ALTER - використовується для зміни об'єктів бази даних;
* DROP - використовується для видалення об'єктів бази даних.

1. DML (Data Manipulation Language) - мова маніпулювання даними. У число запитів цього типу входять різні команди, використовуючи які безпосередньо виробляються деякі маніпуляції з даними. DML-запити потрібні для додавання змін до вже внесені дані, для отримання даних з БД, для їх збереження, для поновлення різних записів і для їх видалення з БД. У число елементів DML-звернень входить основна частина SQL операторів.

* SELECT - здійснює вибірку даних;
* INSERT - додає нові дані;
* UPDATE - змінює існуючі дані;
* DELETE - видаляє дані.

1. DCL (Data Control Language) - мова керування даними. Включає в себе запити і команди, що стосуються дозволів, прав і інших налаштувань СУБД.

* GRANT - надає користувачеві або групі дозволу на певні операції з об'єктом;
* REVOKE - відкликає видані дозволи;
* DENY- задає заборона, має пріоритет над вирішенням.

1. TCL (Transaction Control Language) - мова управління транзакціями. Конструкції такого типу застосовують щоб управляти змінами, які виробляються з використанням DML запитів. Конструкції TCL дозволяють нам виробляти об'єднання DML запитів в набори транзакцій. BEGIN TRANSACTION - служить для визначення початку транзакції;

* COMMIT TRANSACTION - застосовує транзакцію;
* ROLLBACK TRANSACTION - відкочується все зміни, зроблені в контексті поточної транзакції;
* SAVE TRANSACTION - встановлює проміжну точку збереження в транзакції.