НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра автоматизованих систем обробки інформації і управління

**КУРСОВА РОБОТА**

з дисципліни: «Бази даних»

на тему: «ІС Військово-патріотична гра»

Студента 2 курсу, групи ІП-92

напряму підготовки «ІПЗ»

спеціальності 121

Прошина Назарія Анатолійовича

Керівник ст. вик. Недашківський Є. А.

Київ - 2020 рік

Національний технічний університет України “КПІ”

Кафедра автоматизованих систем обробки інформації і управління

Дисципліна Бази даних

Курс 2 Група ІП-92 Семестр 1

**ЗАВДАННЯ**

**на курсову роботу студентів**

|  |
| --- |
| **Прошина Назарія Анатолійовича** |
|  |

(прізвище, ім’я, по батькові)

|  |  |
| --- | --- |
| 1. Тема роботи | ІС «Військово-патріотична гра» |
|  | |
|  | |

|  |  |
| --- | --- |
| 2. Строк здачі студентами закінченої роботи | «30» грудня 2020 р. |

|  |
| --- |
| 3. Технічне завдання:  Створено реляційну базу даних у третій нормальній формі. Структура узгоджена з керівником. База включає щонайменше шість таблиць з не менш ніж десятьма рядками у кожній для репрезентації даних. Базу даних доповнює застосунок, що дозволяє вивести на екран вміст однієї з таблиць.  Застосунок дозволяє проводити фільтрацію виведення. Додана можливість посторінкового виведення даних задавши кількість рядків на сторінку та можливість обрати номер сторінки. Передбачено інтерфейс користувача.  Реалізована модифікація даних (додавання, редагування, видалення) через інтерфейс користувача.  Структура проекту на рівні бази даних містить наступні об’єкти: представлення (view), табличну та скалярну функції, збережену процедуру, тригер, принаймні одна з таблиць передбачає значення «за замовчанням», принаймні в одному з вищеперечислених об’єктів присутні оператори JOIN, CASE, використано підзапит.  Передбачено окремого користувача для застосунку, що не матиме прав на редагування структури бази даних, проте матиме досить прав для роботи. |

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

|  |
| --- |
| Вступ, постановка задач, теоретичні відомості, опис програмнго забезпечення, |
| інструкція користувача, висновок, перелік посилань |
|  |
|  |

**Зміст**

[Вступ 4](#_Toc59849426)

[Постановка задачі 5](#_Toc59849427)

[Проектування бази даних 6](#_Toc59849428)

[Опис програмного забезпечення 9](#_Toc59849429)

[Висновок 17](#_Toc59849430)

[Література 19](#_Toc59849431)

[Додатки 20](#_Toc59849432)

# **Вступ**

Автоматизація процесів спрощує процес роботи з даними, що є дуже важливим , адже ,зважаючи на сьогоднішні реалії, розмір даних вимірюється в терабайтах. Однією із сфер, де простота процесу роботи х даними грає велику роль є гейминг. Навіть у звичайній грі база даних може складатися з сотні таблиць та мільйонів рядків , а отже автоматизація є невід’ємною .

Метою курсового проекту з дисципліни "Бази даних" було розробити застосунок для автоматизації роботи “Військово-патріотичної гри”. Мій вибір впав саме на цю тему , оскільки я являюсь фанатом різних шутерів і неодноразово задумувався над можливостями покращення гри.

Для розв'язання задачі мною була розроблена та узгоджена з викладачем реляційна база даних. Для бази даних використана мова MS SQL Server, а для застосунку (інтерфейс користувача), мова програмування C#.

Отриманий застосунок повністю функціональний та виконує ряд наступних задач: надає доступ до окремої інформації про танки(ціна,вартісь,тип...), відслідковування інформації про створення нової одиниці техніки ,її видалення чи оновлення інформації про неї. У застосунку передбачено дві ролі користувачів: Player та Developer. Developer здатний додавати нові одиниці техніки, видаляти їх чи оновлювати інформацію про них, а також переглядати інофрмацію, Player здатний лише переглядати інформацію.

# **Постановка задачі**

Поставлена викладачем задача звучить так: створити реляційну базу даних у третій нормальній формі і застосунок із графічним інтерфейсом для роботи із нею. Фінальний застосунок має забезпечити роботу з базою даних “Військово-патріотичної гри”.

За допомогою застосунку можна переглядати та модифікувати дані: додавати , видаляти , редагувати. Реалізовати модифікований вивід інформації : посторінкове виведення даних , по 11 рядків на кожній сторінці , з змогою переключатися між сторінками та обрати номер сторінки.

На рівні бази даних реалізувати ряд об’єктів: табличну та скалярні функції , представлення, збережену процедуру , тригер. Хоча б в одному з об’єктів використати оператори: JOIN , UNION , CASE , використовувати підзапит. Створити окремого користувача для застосунку , що не матиме прав на редагування структури бази даних, проте матиме досить прав для роботи.

# **Проектування бази даних**

Опис сутностей:

|  |  |
| --- | --- |
| **Сутність** | **Опис сутності** |
| CombatVehicle | Містить основну інформацію про бойову одиницю. |
| Suspension | Містить інформацію ходову бойової одиниці. |
| Tank | Містить детальну інформацію про бойову одиницю. |
| Player | Містить інформацію про гравців гри. |
| PlayerVehicle | Містить інформацію про наявність бойової одиниці в гравця. |
| Armament | Містить основну інформацію про озброєння бойової машини. |
| Guns | Містить детальну інформацію про гармати,які встановлюються на бойові одиниці |
| TypeOfGun | Містить інформацію про тип гармати |

Зв’язки між сутностями:

|  |  |
| --- | --- |
| **Сутність** | **Зв’язок** |
| CombatVehicle | Один до багатьох Tank  Один до багатьоx Suspension  Один до багатьох PlayerVehicle  Один до багатьох Armament |
| Suspension | Багато до одного з Tank,  Один до багатьох із CombatVehicle |
| Tank | Багато до одного з Suspension, Один до багатьох із CombatVehicle |
| Player | Багато до одного з із PlayerVehicle |
| PlayerVehicle | Один до багатьох із CombatVehicle, Один до багатьох із Player |
| Armament | Багато до одного з TypeOfGun, один до багатьох з CombatVehicle |
| Guns | Багато до одного з TypeOfGun ,один до багатьох з Arnament |
| TypeOfGun | Один до багатьох з Guns |

Діаграма сутностей та бази даних

На основі опису сутностей та зв’язків між ними було побудовано дві діаграми: ER – діаграма сутностей, та DB – діаграма бази даних.

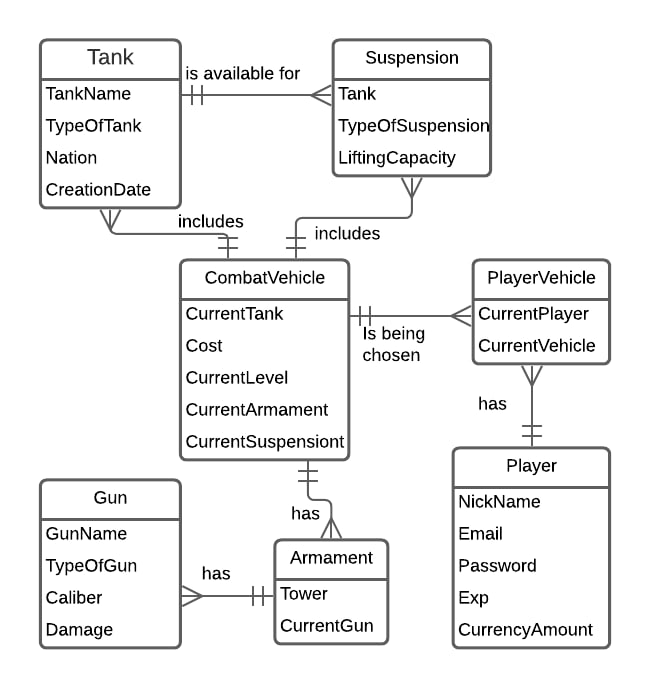


Рисунок 1. ER діаграма

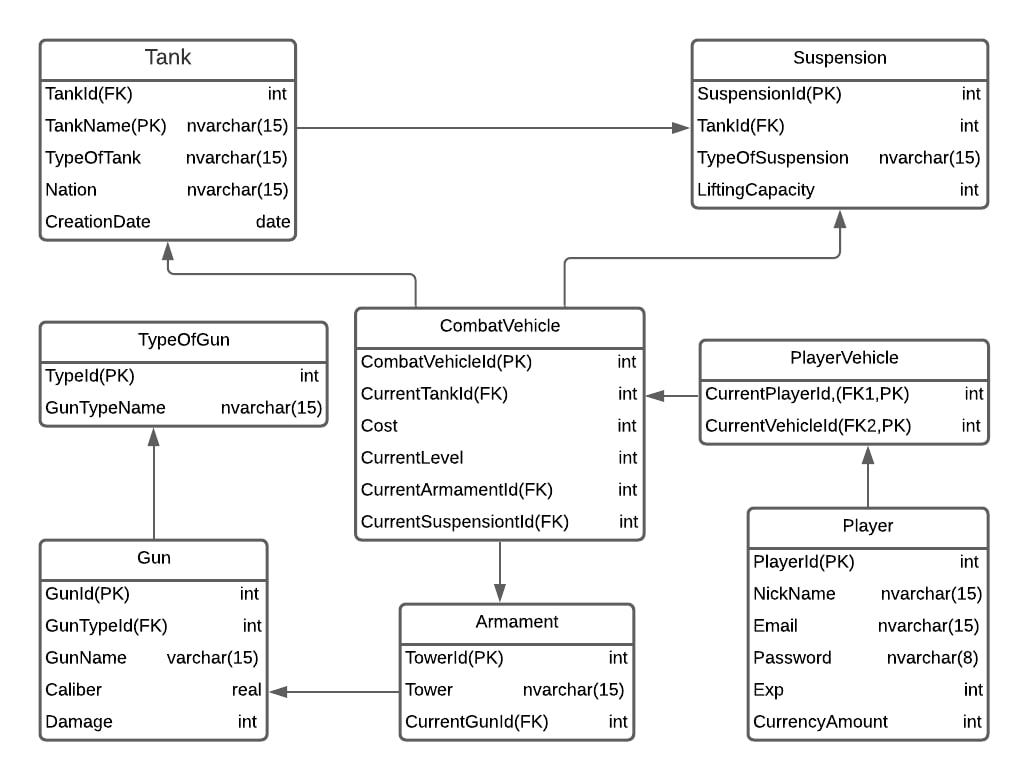


Рисунок 2. DB діаграма

# **Опис програмного забезпечення**

Загальний опис програмного продукту

В застосунку було реалізовано наступний функціонал.

Форма для входу в систему, це реалізовано з метою збільшення надійності застосунку та фіксування ким та коли відбулася зміна в базі даних.

Для Player реалізовані такі можливості: перегляд списку бойових одиниць ,які зараз є частиною гри, а також інформації про них.

Для Developer, включаючи вищеперечислені можливості, також здатний додавати бойову одиницю в гру , видаляти її та змінювати інформацію про неї .

Опис програмної реалізації

Для роботи із базою даних були написані MS SQL Server запити, що потім були під’єднані до користувацького інтерфейсу.

Представлення (view,union) для всіх модулів, які наявні в грі:

CREATE VIEW Moduls

AS

SELECT DISTINCT dbo.Armament.Tower FROM dbo.Armament

UNION ALL

SELECT DISTINCT dbo.Guns.GunName FROM dbo.Guns

UNION ALL

SELECT DISTINCT dbo.Suspension.TypeOfSuspension FROM dbo.Suspension

GO

Представлення (view,inner join) для ,башень , які встановлені на такнки зараз:

CREATE VIEW ArmamentMountedOnTanks AS

SELECT dbo.Tank.TankName AS TankName,

dbo.Armament.Tower AS Tower

FROM dbo.Tank INNER JOIN dbo.CombatVehicle ON dbo.Tank.TankId = dbo.CombatVehicle.CurrentTankId

INNER JOIN dbo.Armament ON dbo.CombatVehicle.CurrentArmamentId = dbo.Armament.TowerId

GO

Таблична функція (також використано підзапит) для отримання даних про певну гармату:

CREATE FUNCTION GunIformation

(

@TankIds int=1

)

RETURNS TABLE

AS

RETURN(

SELECT \* FROM dbo.Guns

WHERE dbo.Guns.GunId IN(SELECT dbo.Armament.CurrentGunId

FROM dbo.Armament

WHERE dbo.Armament.CurrentGunId IN(SELECT dbo.CombatVehicle.CurrentArmamentId FROM

dbo.CombatVehicle WHERE dbo.CombatVehicle.CurrentTankId= @Tankids))

)

GO

Скалярна функція(Використано Case) для отримання кількості досвіду, необхідного для дослідження танка:

CREATE FUNCTION ExperienceForTankResearch

(

@Tankid int

)

RETURNS int

AS

BEGIN

DECLARE @exper int = 0;

DECLARE @type nvarchar(255);

DECLARE @pursh int;

DECLARE @lvl int;

SET @pursh=(SELECT Cost FROM Info WHERE TankNumber=@Tankid)

SET @lvl=(SELECT CurrentLevel FROM Info WHERE TankNumber=@Tankid)

SET @type=(SELECT TypeOfTank FROM Info WHERE TankNumber=@Tankid)

SET @exper= CASE @type

WHEN 'MediumTank' THEN @pursh\*@lvl\*1

WHEN 'HeavyTank' THEN @pursh\*@lvl\*2

WHEN 'LightTank' THEN @pursh\*@lvl\*2

WHEN 'SAU' THEN @pursh\*@lvl\*15

END

RETURN(@exper)

END

GO

Тригер для додавання інформації про видалених коричтувачів:

CREATE TRIGGER Player\_Delete

ON Player

AFTER DELETE

AS

INSERT INTO History (NickName)

SELECT NickName

FROM DELETED

GO

Значення «за замовчанням» у таблиці:

CREATE TABLE [Player] (

PlayerId int NOT NULL,

NickName varchar(255) NOT NULL UNIQUE,

Email varchar(255) NOT NULL UNIQUE,

Passwords varchar(255) NOT NULL,

Expir int NOT NULL DEFAULT 0,

CurrencyAmount int NOT NULL DEFAULT 0,

CONSTRAINT [PK\_PLAYER] PRIMARY KEY CLUSTERED

(

[PlayerId] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

Збережена процедура для додавання нового гравця :

CREATE PROCEDURE Add\_Player

@Player int,

@Nick nvarchar(255),

@Email nvarchar(255),

@Pass nvarchar(255)='12345'

AS

INSERT INTO dbo.Player(PlayerId, NickName,Email, Passwords, Expir ,CurrencyAmount)

VALUES(@Player,@Nick,@Email,@Pass,0,0)

GO

Окремий користувач застосунку, створення advanced користувача:

CREATE ROLE [OnlyRead]

GRANT SELECT ON [dbo].[Info] TO [OnlyRead];

DENY DELETE ON [dbo].[Info] TO [OnlyRead];

DENY UPDATE ON [dbo].[Info] TO [OnlyRead];

DENY Insert ON [dbo].[Info] TO [OnlyRead];

CREATE LOGIN Employee1 WITH PASSWORD = 'qwerty123';

CREATE USER Employee1 FOR LOGIN Employee1;

ALTER ROLE [OnlyRead] ADD MEMBER Employee1

CREATE ROLE [SystemAdmin]

GRANT SELECT ON [dbo].[Info] TO [SystemAdmin];

GRANT DELETE ON [dbo].[Info] TO [SystemAdmin];

GRANT UPDATE ON [dbo].[Info] TO [SystemAdmin];

GRANT INSERT ON [dbo].[Info] TO [SystemAdmin];

CREATE LOGIN Employee2 WITH PASSWORD = 'qwerty123';

CREATE USER Employee2 FOR LOGIN Employee2;

ALTER ROLE [SystemAdmin] ADD MEMBER Employee2

Приклади роботи

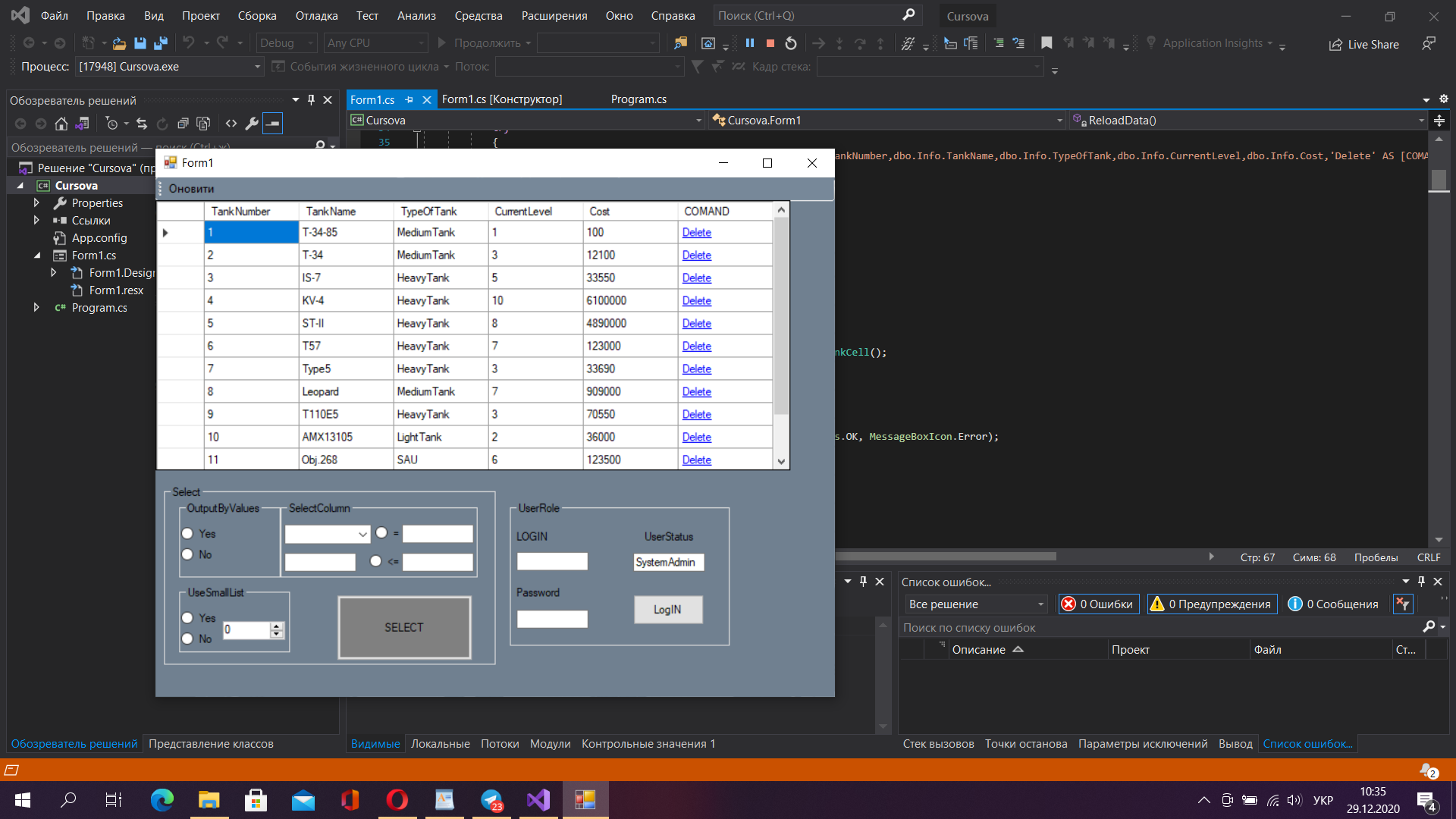


Рисунок 3. Інтерфейс застосунку

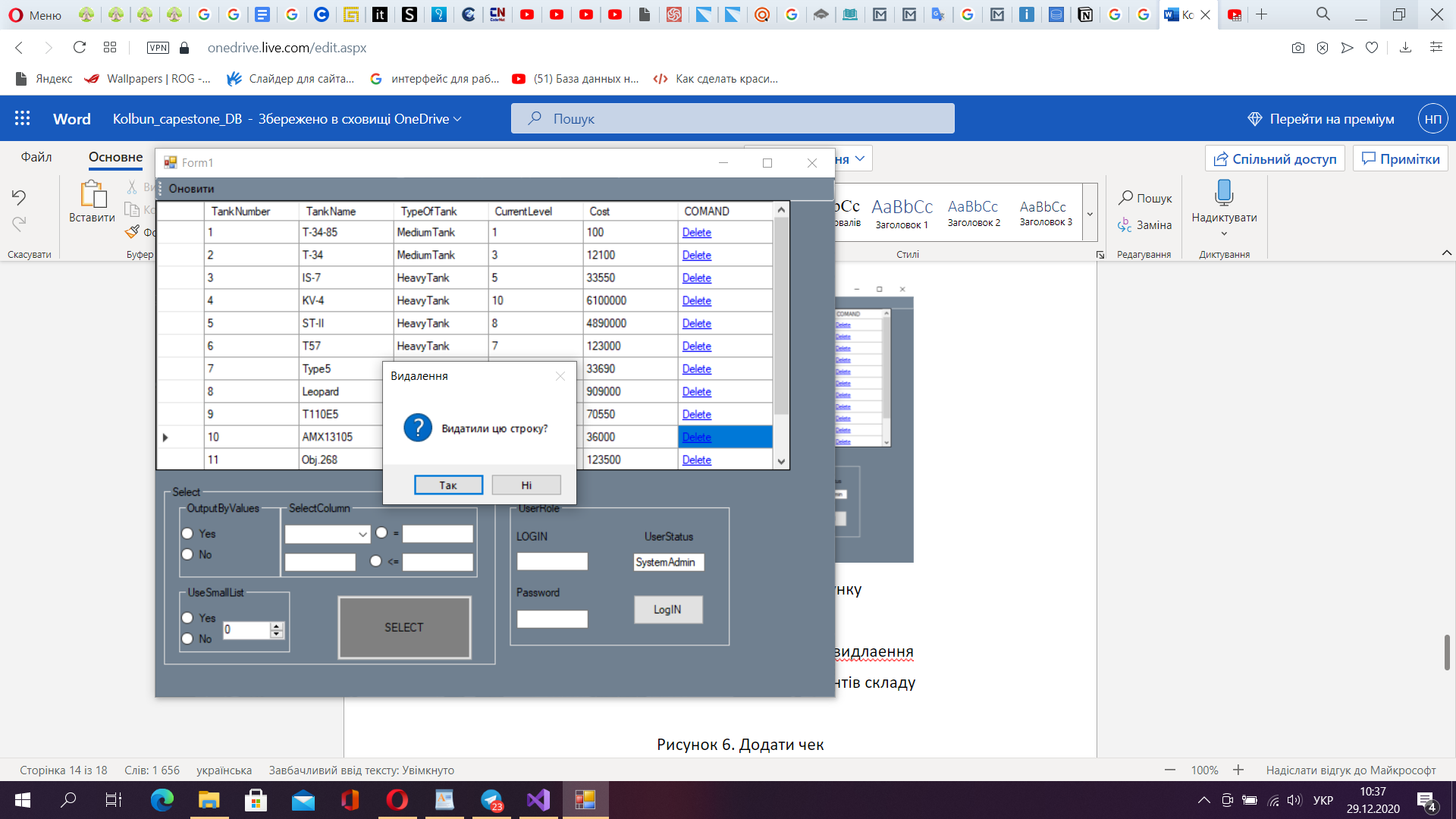


Рисунок 4. Запит на підтвердження видалення

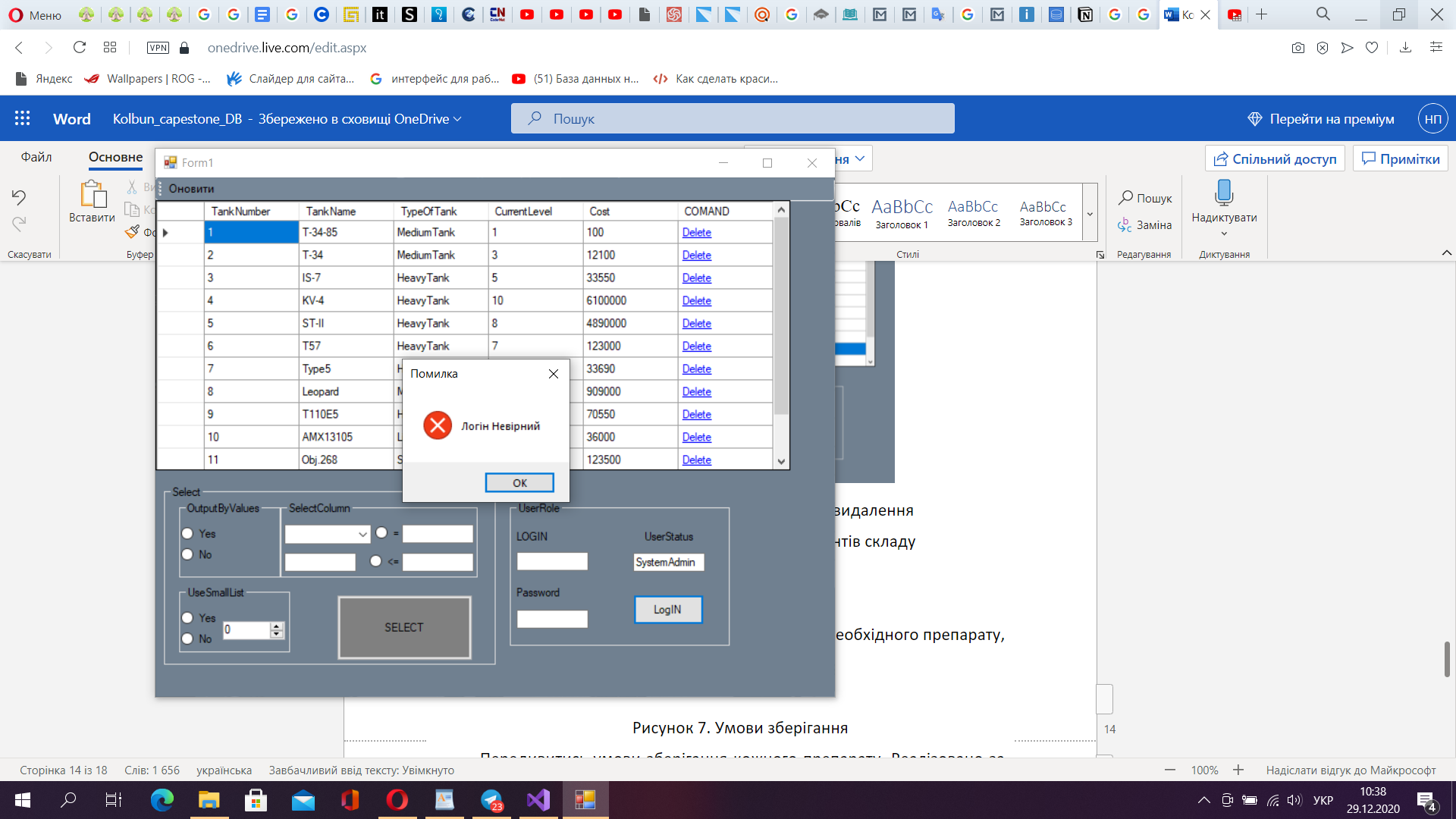


Рисунок 5. Повідомлення про невдалий вхід

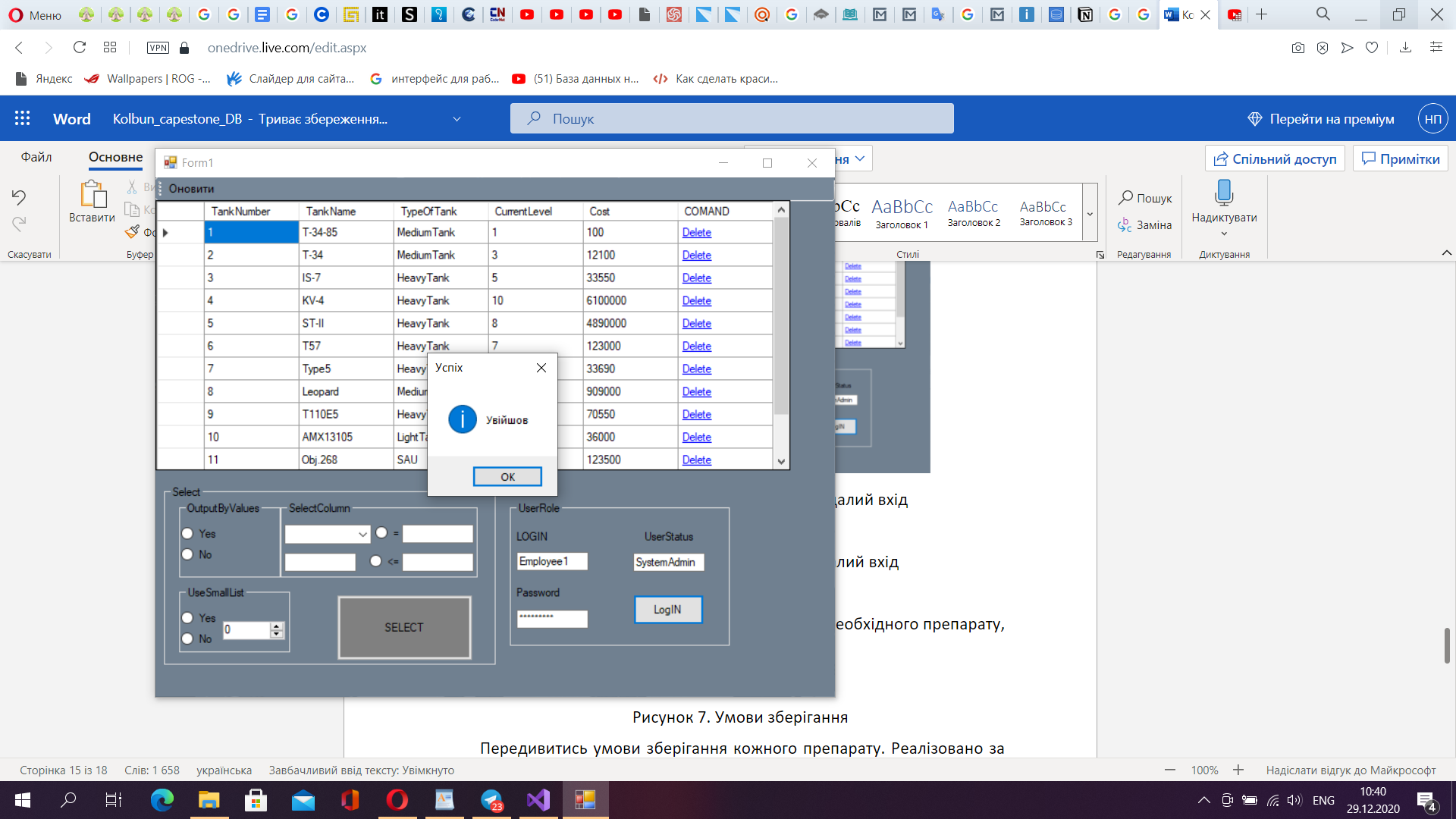


Рисунок 6. Повідомлення про вдалий вхід

# **Висновок**

У ході виконання даного курсового проекту, мною було розроблено базу даних «Військово-патріотичної гри» та застосунок для роботи із нею. Були виділені основні сутності, що містили б необхідні дані для роботи: Player, PlayerVehicle, Arnament, Suspension, Guns, TypeOfGun, CombatVehicle , Tank , що дозволяє забезпечити повний функціонал гри . Також було розроблено та узгоджено з викладачем діаграму сутностей, та діаграму фізичної бази даних.

Була створена база даних, згідно до діаграм; заповнення бази даними для презентації роботи. Було реалізовано основні функції для роботи із базою даних, які були вказані у технічному завданні. Перелік функцій: перегляд бойових одиниць, їхня фільтрація за назвою , типом, ціною; створення бойової одиниці, її видалення та редагування інформації про неї.

У майбутньому планується розширення можливостей додатку. Передбачається змога працювати з всіма таблицями, які є в базі даних. Окрім цього, передбачається збільшення списку ролей, що дозволить збільшити функціонал застосунку.

Для написання СУБД і допоміжного застосунку були використані MS SQL Server 2019 версії 15.0 та C# версії 4.4, а також бібліотека WindowsForms версії 5.0 для користувацького інтерфейсу. Такий вибір програмного забезпечення полягав у тому, що для такого списку технологій створена велика кількість прикладів та документації .

Під час написання курсової роботи я дізнався про особливості підключення бази даних до користувацького інтерфейсу, що є корисною навичкою для майбутньої роботи. Опанував проектування власної бази даних у третій нормальній формі та створення діаграм сутностей і БД.

# **Література**

1. Інформація про бойові одиниці, модулі, що встановлюються на них <https://worldoftanks.ru/ru/tankopedia/#wot&w_m=tanks>

2. Документація бібліотеки WinForms. <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-5.0>

3. Документація мови програмування С#. <https://docs.microsoft.com/en-us/dotnet/csharp/>

4. Документація мови MS SQl Server. [Електронний ресурс]. <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15>

# **Додатки**

Додаток А: код програмного застосунку

CREATE DATABASE Game

GO

USE Game

GO

----------------------------------

--Створення БД

----------------------------------

CREATE TABLE [TypeOfTank] (

TypeId int NOT NULL,

GunTypeName varchar(255) NOT NULL,

CONSTRAINT [PK\_TYPEOFTANK] PRIMARY KEY CLUSTERED

(

[TypeId] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [Guns] (

GunId int NOT NULL,

GunTypeId int NOT NULL,

GunName varchar(255) NOT NULL UNIQUE,

Caliber int NOT NULL,

Damage int NOT NULL,

CONSTRAINT [PK\_GUN] PRIMARY KEY CLUSTERED

(

[GunId] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [Armament] (

TowerId int NOT NULL,

Tower varchar(255) NOT NULL,

CurrentGunId int NOT NULL,

CONSTRAINT [PK\_ARMAMENT] PRIMARY KEY CLUSTERED

(

[TowerId] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

--значення за замовчуванням

CREATE TABLE [Player] (

PlayerId int NOT NULL,

NickName varchar(255) NOT NULL UNIQUE,

Email varchar(255) NOT NULL UNIQUE,

Passwords varchar(255) NOT NULL,

Expir int NOT NULL DEFAULT 0,

CurrencyAmount int NOT NULL DEFAULT 0,

CONSTRAINT [PK\_PLAYER] PRIMARY KEY CLUSTERED

(

[PlayerId] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [PlayerVehicle] (

CurrentPlayerId int NOT NULL,

CurrentVehicleId int NOT NULL,

CONSTRAINT [PK\_PLAYERVEHICLE] PRIMARY KEY CLUSTERED

(

[CurrentPlayerId] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [CombatVehicle] (

CombatVehicleId int NOT NULL,

CurrentTankId int NOT NULL,

Cost int NOT NULL,

CurrentLevel int NOT NULL,

CurrentArmamentId int NOT NULL,

CurrentSuspensiontId int NOT NULL,

CONSTRAINT [PK\_COMBATVEHICLE] PRIMARY KEY CLUSTERED

(

[CombatVehicleId] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

----------

CREATE TABLE [Info] (

TankNumber int NOT NULL ,

Cost int NOT NULL,

CurrentLevel int NOT NULL,

TankName varchar(255) NOT NULL,

TypeOfTank varchar(255) NOT NULL

CONSTRAINT [PK\_INFO] PRIMARY KEY CLUSTERED

(

[TankNumber] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

------

CREATE TABLE History

(

Id INT IDENTITY PRIMARY KEY,

NickName nvarchar(255) NOT NULL,

DeleteAt DATETIME NOT NULL DEFAULT GETDATE(),

);

CREATE TABLE [Tank] (

TankName varchar(255) NOT NULL,

TankId int NOT NULL,

TypeOfTank varchar(255) NOT NULL,

Nation varchar(255) NOT NULL,

CreationDate date NOT NULL,

CONSTRAINT [PK\_TANK] PRIMARY KEY CLUSTERED

(

[TankName] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

CREATE TABLE [Suspension] (

SuspensionId int NOT NULL,

TankId int NOT NULL,

TypeOfSuspension varchar(255) NOT NULL,

LiftingСapacity int NOT NULL,

CONSTRAINT [PK\_SUSPENSION] PRIMARY KEY CLUSTERED

(

[SuspensionId] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

--показує башні встановлені на танки

--view , join

CREATE VIEW ArmamentMountedOnTanks AS

SELECT dbo.Tank.TankName AS TankName,

dbo.Armament.Tower AS Tower

FROM dbo.Tank INNER JOIN dbo.CombatVehicle ON dbo.Tank.TankId = dbo.CombatVehicle.CurrentTankId

INNER JOIN dbo.Armament ON dbo.CombatVehicle.CurrentArmamentId = dbo.Armament.TowerId

GO

select \* from Player

go

CREATE TRIGGER Player\_Delete

ON Player

AFTER DELETE

AS

INSERT INTO History (NickName)

SELECT NickName

FROM DELETED

GO

--показує всі модулі ,які можна встановити на танки

--union

CREATE VIEW Moduls

AS

SELECT DISTINCT dbo.Armament.Tower FROM dbo.Armament

UNION ALL

SELECT DISTINCT dbo.Guns.GunName FROM dbo.Guns

UNION ALL

SELECT DISTINCT dbo.Suspension.TypeOfSuspension FROM dbo.Suspension

GO

--виводить дані про пушку,встановлену на танк

--підзапит

CREATE FUNCTION GunIformation

(

@TankIds int=1

)

RETURNS TABLE

AS

RETURN(

SELECT \* FROM dbo.Guns

WHERE dbo.Guns.GunId IN(SELECT dbo.Armament.CurrentGunId

FROM dbo.Armament

WHERE dbo.Armament.CurrentGunId IN(SELECT dbo.CombatVehicle.CurrentArmamentId FROM

dbo.CombatVehicle WHERE dbo.CombatVehicle.CurrentTankId= @Tankids))

)

GO

--таблична функція

CREATE FUNCTION TanksInRange

(

@beg int,

@ending int

)

RETURNS TABLE

AS

RETURN(

SELECT \* FROM dbo.Info

WHERE dbo.Info.TankNumber BETWEEN @beg And @ending

)

GO

SELECT \* FROM TanksInRange(5,9)

GO

--процедура

CREATE PROCEDURE Add\_Player

@Player int,

@Nick nvarchar(255),

@Email nvarchar(255),

@Pass nvarchar(255)='12345'

AS

INSERT INTO dbo.Player(PlayerId, NickName,Email, Passwords, Expir ,CurrencyAmount)

VALUES(@Player,@Nick,@Email,@Pass,0,0)

GO

--скалярна функція,case

CREATE FUNCTION ExperienceForTankResearch

(

@Tankid int

)

RETURNS int

AS

BEGIN

DECLARE @exper int = 0;

DECLARE @type nvarchar(255);

DECLARE @pursh int;

DECLARE @lvl int;

SET @pursh=(SELECT Cost FROM Info WHERE TankNumber=@Tankid)

SET @lvl=(SELECT CurrentLevel FROM Info WHERE TankNumber=@Tankid)

SET @type=(SELECT TypeOfTank FROM Info WHERE TankNumber=@Tankid)

SET @exper= CASE @type

WHEN 'MediumTank' THEN @pursh\*@lvl\*1

WHEN 'HeavyTank' THEN @pursh\*@lvl\*2

WHEN 'LightTank' THEN @pursh\*@lvl\*2

WHEN 'SAU' THEN @pursh\*@lvl\*15

END

RETURN(@exper)

END

GO

SELECT dbo.ExperienceForTankResearch(3) AS Experience

GO

ALTER TABLE [Guns] WITH CHECK ADD CONSTRAINT [Gun\_fk0] FOREIGN KEY ([GunTypeId]) REFERENCES [TypeOfTank]([TypeId])

ON UPDATE CASCADE

GO

ALTER TABLE [Guns] CHECK CONSTRAINT [Gun\_fk0]

GO

ALTER TABLE [Armament] WITH CHECK ADD CONSTRAINT [Armament\_fk0] FOREIGN KEY ([CurrentGunId]) REFERENCES [Guns]([GunId])

ON UPDATE CASCADE

GO

ALTER TABLE [Armament] CHECK CONSTRAINT [Armament\_fk0]

GO

ALTER TABLE [PlayerVehicle] WITH CHECK ADD CONSTRAINT [PlayerVehicle\_fk0] FOREIGN KEY ([CurrentPlayerId]) REFERENCES [Player]([PlayerId])

ON UPDATE CASCADE

GO

ALTER TABLE [PlayerVehicle] CHECK CONSTRAINT [PlayerVehicle\_fk0]

GO

ALTER TABLE [PlayerVehicle] WITH CHECK ADD CONSTRAINT [PlayerVehicle\_fk1] FOREIGN KEY ([CurrentVehicleId]) REFERENCES [CombatVehicle]([CombatVehicleId])

ON UPDATE CASCADE

GO

ALTER TABLE [PlayerVehicle] CHECK CONSTRAINT [PlayerVehicle\_fk1]

GO

-----

ALTER TABLE [CombatVehicle] WITH CHECK ADD CONSTRAINT [CombatVehicle\_fk0] FOREIGN KEY ([CurrentTankId]) REFERENCES [Tank]([TankId])

ON UPDATE CASCADE

GO

ALTER TABLE [CombatVehicle] CHECK CONSTRAINT [CombatVehicle\_fk0]

GO

-----

-----

-----

ALTER TABLE [CombatVehicle] WITH CHECK ADD CONSTRAINT [CombatVehicle\_fk1] FOREIGN KEY ([CurrentArmamentId]) REFERENCES [Armament]([TowerId])

ON UPDATE CASCADE

GO

ALTER TABLE [CombatVehicle] CHECK CONSTRAINT [CombatVehicle\_fk1]

GO

ALTER TABLE [CombatVehicle] WITH CHECK ADD CONSTRAINT [CombatVehicle\_fk2] FOREIGN KEY ([CurrentSuspensiontId]) REFERENCES [Suspension]([SuspensionId])

ON UPDATE CASCADE

GO

ALTER TABLE [CombatVehicle] CHECK CONSTRAINT [CombatVehicle\_fk2]

GO

----

----

----

ALTER TABLE [Tank] WITH CHECK ADD CONSTRAINT [Tank\_fk0] FOREIGN KEY ([TankId]) REFERENCES [Suspension]([TankId])

ON UPDATE CASCADE

GO

ALTER TABLE [Tank] CHECK CONSTRAINT [Tank\_fk0]

GO

----

----

----

ALTER TABLE [Suspension] WITH CHECK ADD CONSTRAINT [Suspension\_fk0] FOREIGN KEY ([TankId]) REFERENCES [Tank]([TankId])

ON UPDATE CASCADE

GO

ALTER TABLE [Suspension] CHECK CONSTRAINT [Suspension\_fk0]

GO

---------------

---------------

--Наповнення Таблиць

---------------

---------------

INSERT dbo.TypeOfTank(TypeId,GunTypeName)

VALUES (1, 'common')

GO

INSERT dbo.TypeOfTank(TypeId,GunTypeName)

VALUES (2, 'swinging'),

(3, 'artsau'),

(4, 'sau'),

(5, 'double-barreled'),

(6, 'semi-sau'),

(7, 'semi-artsau'),

(8, 'hydraulic'),

(9, 'motionless'),

(10, 'terrestrial')

GO

INSERT dbo.Guns(GunId,GunTypeId,GunName,Caliber,Damage)

VALUES (1,1,'BL-9',152,750)

GO

INSERT dbo.Guns(GunId,GunTypeId,GunName,Caliber,Damage)

VALUES

(2,2,'BL-10',152,750),

(3,3,'Zis-10',100,300),

(5,5,'Zis-15',76,180),

(6,1,'T4',57,75),

(7,8,'Pzkfc-1',122,320),

(8,2,'DGKN',88,240),

(9,6,'Plot',75,100),

(10,8,'Zeves',107,240),

(11,10,'Link-1',15,50),

(12,9,'BL-7',192,1090),

(13,3,'Pzcfc-5',750,10000)

GO

INSERT dbo.Armament(TowerId,Tower,CurrentGunId)

VALUES (1,'T34',1)

GO

INSERT dbo.Armament(TowerId,Tower,CurrentGunId)

VALUES

(2,'T34',2),

(3,'T34',3),

(4,'T-34-85',2),

(5,'T4',2),

(6,'T5',5),

(7,'T4',1),

(8,'T5',6),

(9,'Tiger',7),

(10,'Maus',1),

(11,'Kv-4',3),

(12,'IS-7',2),

(13,'T-54',8)

GO

INSERT dbo.Player(PlayerId,NickName,Email,Passwords,Expir,CurrencyAmount)

VALUES (1,'NP','NP@gmail.com','12356',123400,7000)

GO

INSERT dbo.Player(PlayerId,NickName,Email,Passwords,Expir,CurrencyAmount)

VALUES

(2,'Jove','Jove@gmail.com','1212356',12300,600),

(3,'MP','MP@gmail.com','1qwr356',12445,75000),

(4,'Porh','Porh@gmail.com','121256',123124400,3000),

(5,'Qwerty','Qwerty@gmail.com','12123356',120,7200),

(6,'JamesBond','JamesBond@gmail.com','1qw2356',3400,1000),

(7,'Kulia','Kulia@gmail.com','12123356',1279,70000),

(8,'NazckiSubaru','NazckiSubaru@gmail.com','14q2356',6457,17000),

(9,'KanekckiKen','KanekckiKen@gmail.com','12das356',23547,72000),

(10,'Naruto','Naruto@gmail.com','1212as356',346347,70700),

(11,'Sasuke','Sasuke@gmail.com','12asd356',34784,70001),

(12,'Meliodas','Meliodas@gmail.com','1212a356',34637,71000),

(13,'WinsonKley','WinsonKley@gmail.com','123asf56',347488,67000),

(14,'Rouch','Rouch@gmail.com','1asd2356',232352,7000),

(15,'GeraltIzRIVIII','GeraltIzRIVIII@gmail.com','12ad356',4567547,997000)

GO

INSERT dbo.Suspension(SuspensionId,TankId,TypeOfSuspension,LiftingСapacity)

VALUES (1,1,'ordinary',720)

GO

INSERT dbo.Suspension(SuspensionId,TankId,TypeOfSuspension,LiftingСapacity)

VALUES

(2,1,'wheeled',1000),

(3,2,'special',1240),

(4,1,'ordinary',560),

(5,3,'ordinary',10000),

(6,4,'special',5400),

(7,1,'wheeled',12415),

(8,5,'ordinary',12414),

(9,6,'special',12456),

(10,7,'ordinary',12523),

(11,8,'special',72)

GO

INSERT dbo.Tank(TankName,TankId,TypeOfTank,Nation,CreationDate)

VALUES ('T-34-85',1,'MediumTank','USSR','1942-01-01')

GO

INSERT dbo.Tank(TankName,TankId,TypeOfTank,Nation,CreationDate)

VALUES

('T-34',2,'MediumTank','USSR','1941-10-01'),

('IS-7',3,'HeavyTank','USSR','1943-01-10'),

('KV-4',4,'HeavyTank','USSR','1940-09-01'),

('ST-II',5,'HeavyTank','USSR','1947-01-09'),

('T57',6,'HeavyTank','USA','1960-11-01'),

('Type5',7,'HeavyTank','Japan','1939-01-11'),

('Leopard',8,'MediumTank','Germany','1970-08-01'),

('T110E5',9,'HeavyTank','USA','1960-01-08'),

('AMX13105',10,'LightTank','France','1960-07-01'),

('Obj.777',11,'HeavyTank','USSR','1946-01-07'),

('Obj.268',11,'SAU','USSR','1948-11-11'),

('Obj.268/4',11,'SAU','USSR','1945-12-12'),

('Obj.257F',11,'HeavyTank','USSR','1999-10-10')

GO

INSERT dbo.CombatVehicle(CombatVehicleId,CurrentTankId,Cost,CurrentLevel,CurrentArmamentId,CurrentSuspensiontId)

VALUES (1,1,100,1,1,1)

GO

INSERT dbo.CombatVehicle(CombatVehicleId,CurrentTankId,Cost,CurrentLevel,CurrentArmamentId,CurrentSuspensiontId)

VALUES

(2,3,12100,3,7,1),

(3,4,33550,5,5,2),

(4,7,6100000,10,3,3),

(5,3,4890000,9,2,2),

(6,5,123000,7,6,5),

(7,6,33690,5,7,6),

(8,4,909000,6,8,2),

(9,6,70550,3,9,5),

(10,10,36000,2,10,7),

(11,3,0,1,2,5),

(12,4,230000,8,5,4)

GO

INSERT dbo.PlayerVehicle(CurrentPlayerId,CurrentVehicleId)

VALUES (1,1)

GO

INSERT dbo.PlayerVehicle(CurrentPlayerId,CurrentVehicleId)

VALUES

(2,2),

(3,3),

(4,5),

(5,6),

(6,7),

(7,8),

(8,8),

(9,9),

(10,4),

(11,7),

(12,5),

(13,7)

GO

INSERT dbo.Info(TankNumber,Cost,CurrentLevel,TankName,TypeOfTank)

VALUES (10,36000,2,'AMX13105','LightTank'),

(11,123500,6,'Obj.268','SAU'),

(12,102350,6,'Obj.268/4','SAU'),

(13,102670,7,'Obj.777','HeavyTank')

GO

INSERT dbo.Info(TankNumber,Cost,CurrentLevel,TankName,TypeOfTank)

VALUES

(2,12100,3,'T-34','MediumTank'),

(3,33550,5,'IS-7','HeavyTank'),

(4,6100000,10,'KV-4','HeavyTank'),

(5,4890000,9,'ST-II','HeavyTank'),

(6,123000,7,'T57','HeavyTank'),

(7,33690,5,'Type5','HeavyTank'),

(8,909000,6,'Leopard','MediumTank'),

(9,70550,3,'T110E5','HeavyTank'),

(10,36000,2,'AMX13105','LightTank'),

(11,123500,6,'Obj.268','SAU'),

(12,102350,6,'Obj.268/4','SAU'),

(13,102670,7,'Obj.777','HeavyTank')

GO

select \* from dbo.Armament

select\* from dbo.CombatVehicle

-- користувачі

CREATE ROLE [OnlyRead]

GRANT SELECT ON [dbo].[Info] TO [OnlyRead];

DENY DELETE ON [dbo].[Info] TO [OnlyRead];

DENY UPDATE ON [dbo].[Info] TO [OnlyRead];

DENY Insert ON [dbo].[Info] TO [OnlyRead];

CREATE LOGIN Employee1 WITH PASSWORD = 'qwerty123';

CREATE USER Employee1 FOR LOGIN Employee1;

ALTER ROLE [OnlyRead] ADD MEMBER Employee1

CREATE ROLE [SystemAdmin]

GRANT SELECT ON [dbo].[Info] TO [SystemAdmin];

GRANT DELETE ON [dbo].[Info] TO [SystemAdmin];

GRANT UPDATE ON [dbo].[Info] TO [SystemAdmin];

GRANT INSERT ON [dbo].[Info] TO [SystemAdmin];

CREATE LOGIN Employee2 WITH PASSWORD = 'qwerty123';

CREATE USER Employee2 FOR LOGIN Employee2;

ALTER ROLE [SystemAdmin] ADD MEMBER Employee2

SELECT @@version