

УДК 004.65

5.2.2. Математические, статистические и инструментальные методы экономики (физико-математические науки, экономические науки)

РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ МОРСКОГО ПОРТА

Рыбьянцева Мария Сергеевна
кандидат экономических наук, доцент
РИНЦ SPIN-код: 7874-8981
email: Riban1@mail.ru
ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13

Авакимян Наталья Николаевна
К.ф.-м.н., доцент
РИНЦ SPIN-код: 7020-2847
email: avnatali@mail.ru
ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13

Тараненко Вадим Витальевич
студент группы ИТ2201
email: vadimtaranenko325@gmail.com
ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13

Морские порты играют ключевую роль в экспорте и импорте товаров. Они формируют рабочие места, помогают развитию туризма и международной торговли. С увеличением объемов перевозок и усложнением логистических процессов, эффективность управления портовыми операциями снижается. Соответственно, возрастает необходимость разработки более совершенной системы учета операций для оптимизации грузового контроля и уменьшения затрат путем повышения производительности. Проектируемая система должна будет обеспечить надежное управление ключевыми аспектами порта, а именно: регистрацию и отслеживание судов, учет грузов и управление портовыми складами. В данной статье рассмотрены важнейшие аспекты разработки информационной системы грузового порта. Описана ER-диаграмма (выделены следующие сущности: Суда, Грузы, Рейсы, Персонал, Портовые и Финансовые операции). Между таблицами сформированы связи, что создает полноценное представление о движении информационных потоков. В работе проведена визуализация бизнес-процессов, позволяющая выделить важнейшие информационные потоки, необходимые для создания эффективной системы управления.

UDC 004.65

5.2.2. Mathematical, statistical and instrumental methods of economics (physical and mathematical sciences, economic sciences)

DEVELOPMENT AND DESIGN OF A SEAPORT INFORMATION SYSTEM

Rybyantseva Maria Sergeevna
Candidate of Economic Sciences associate professor
RSCI SPIN-code: 7874-8981
email: Riban1@mail.ru
Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13

Avakimyan Natalia Nikolaevna
Cand.Phys.-Math.Sci., associate professor,
RSCI SPIN-code: 7020-2847
email: avnatali@mail.ru
Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13

Taranenko Vadim Vitalyevich
student of IT2201
email: vadimtaranenko325@gmail.com
Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13

Seaports play a key role in the export and import of goods. They create jobs and help develop tourism and international trade. As traffic volumes increase and logistics processes become more complex, the efficiency of port operations management decreases. Accordingly, there is an increasing need to develop a more advanced accounting system to optimize cargo control and reduce costs by increasing productivity. The designed system will need to provide reliable management of key aspects of the port, namely: vessel registration and tracking, cargo accounting and port warehouse management. This article discusses the most important aspects of developing a cargo port information system. An ER diagram is described (the following entities are highlighted: Vessels, Cargo, Flights, Personnel, Port and Financial operations. Connections are formed between the tables, which creates a complete picture of the movement of information flows. The work visualizes business processes, allowing us to highlight the most important information flows necessary to create an effective management system. The information capabilities and codes of the following forms are described: – employee accounting form (reflects the increase in wages and calculation of average accruals to the employee); – a form for registering ships (reflects the possibility of registering ships, generating reports on

Описаны информационные возможности и коды следующих форм: форма учета сотрудников (отражает увеличение заработной платы и расчет средних начислений сотруднику); форма учета судов (отражает возможность учета судов, формирование отчетности по кораблям, отражение маршрутов следования, ранжирования кораблей по тоннажу)

ships, reflecting routes, ranking ships by tonnage).

Ключевые слова: БАЗЫ ДАННЫХ, СУБД, МОРСКОЙ ПОРТ, VISUAL STUDIO, ER-МОДЕЛЬ, СУЩНОСТИ, SQL-ЗАПРОСЫ, ВИЗУАЛИЗАЦИЯ БИЗНЕС-ПРОЦЕССОВ

Keywords: DATABASES, DBMS, SEAPORT, VISUAL STUDIO, ER-MODEL, ENTITIES, SQL-QUESTIONS, VISUALIZATION OF BUSINESS PROCESSES

<http://dx.doi.org/10.21515/1990-4665-200-016>

Введение. Морские порты играют ключевую роль в экспорте и импорте товаров. Они формируют рабочие места, помогают развитию туризма и международной торговли. Деятельность морских портов для некоторых стран, являются условием их экономического процветания.

С увеличением объемов перевозок и усложнением логистических процессов, эффективность управления портовыми операциями снижается. Соответственно, возрастает необходимость разработки более совершенной системы учета операций для оптимизации грузового контроля и уменьшения затрат путем повышения производительности.

Проектируемая система должна будет обеспечить надежное управление ключевыми аспектами порта, а именно: регистрацию и отслеживание судов, учет грузов и управление портовыми складами.

Разработка такой системы требует тщательного анализа специфики устройства морских портов, использования всех возможных источников данных и безопасности на должном уровне.

Материалы и методы. Для разработки информационной системы, функционала были выбраны C#, .NET Framework - кроссплатформенная среда разработки для создания приложений с поддержкой большого количества языков программирования, таких как C#, C++, Java, и другие. К достоинствам выбранной среды разработки в том, что Visual Studio обеспечивает:

<http://ej.kubagro.ru/2024/06/pdf/16.pdf>

- команде разработчиков мощные инструменты для создания высококачественных приложений;
- строгие меры безопасности, в том числе и управление правами доступа и защиту кода.

Для начала работы была создана ER-модель, описывающая структуру созданной базы данных (рисунок 1).

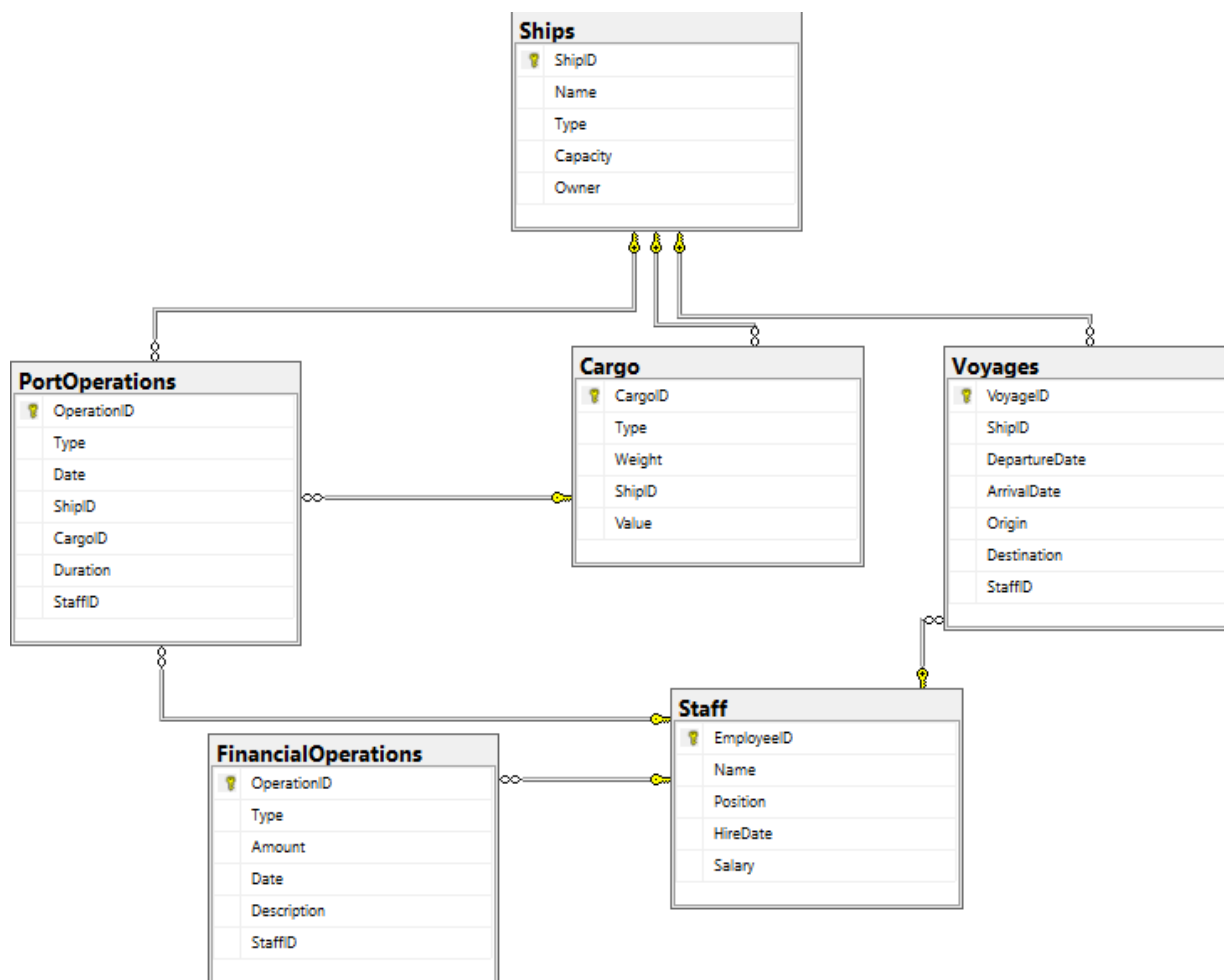


Рис. 1 – ER- модель базы данных

В ходе формирования ER-диаграммы основное внимание уделено созданию структурированной и комплексной системы, которая должна отразить ключевые аспекты работы порта.

Разработанная модель базы данных включает в себя следующие сущности (таблицы): Суда, Грузы, Рейсы, Персонал, Портовые и Финансовые операции. Между таблицами сформированы связи, что

создает полноценное представление о движении информационных потоков.

Для управления данными необходимо детально представлять бизнес-процессы грузового порта, что предусматривает их детализацию (рисунок 2).

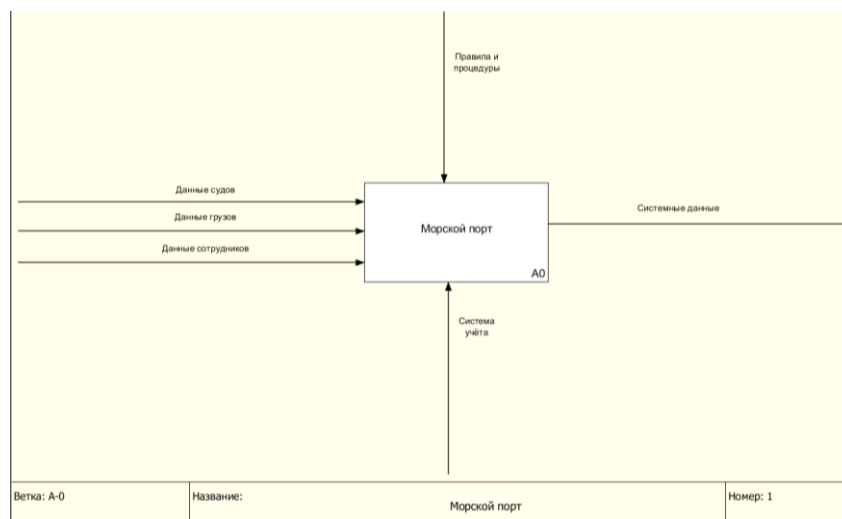


Рис. 2 – Первый уровень детализации

На первом уровне детализации бизнес-процессов определяется, с каким данными предстоит работать во время разработки. Рассматриваемой областью является описание бизнес-процессов работы морского порта, разрабатываемая информационная система будет обрабатывать следующие наборы данных: грузы, суда, сотрудников (рисунок 3).

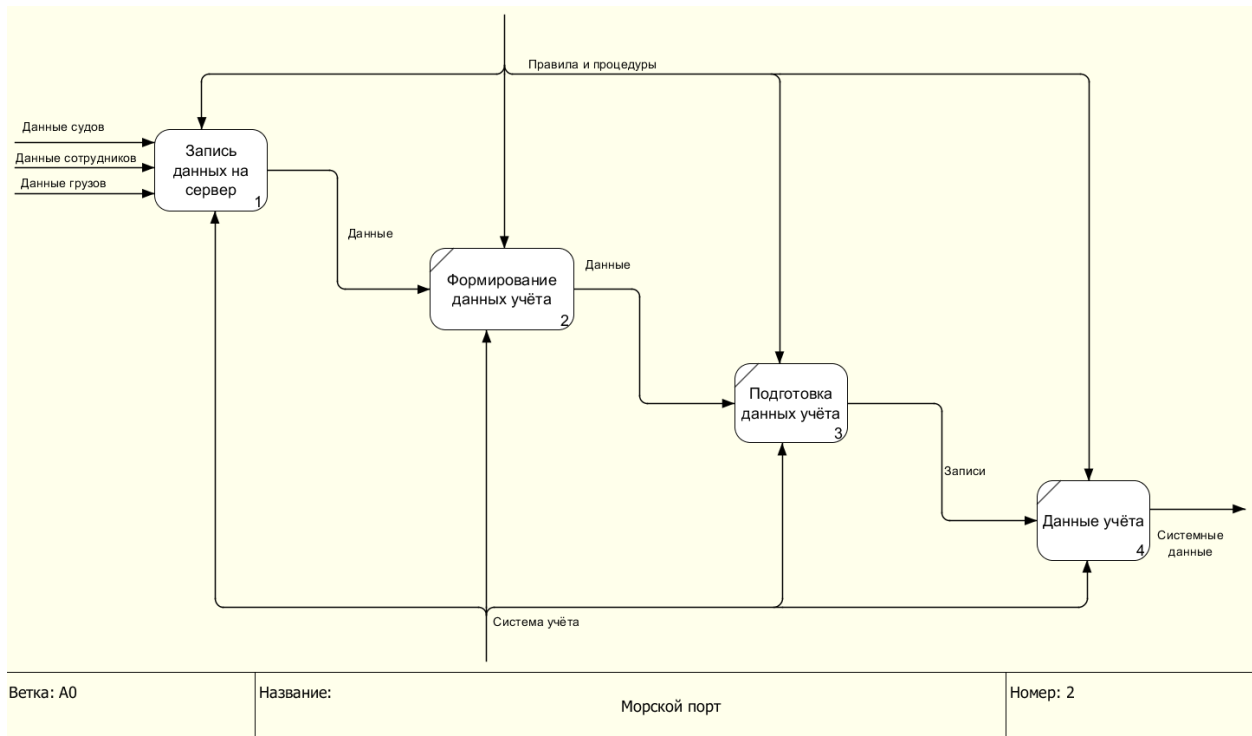


Рис. 3 – Второй уровень детализации

На втором уровне рассматривается более подробная модель системы, возникает необходимость запись данных на сервер базы данных, формируются учетные показатели и отчетные формы.

Третий уровень детализации приведен на рисунке 4.

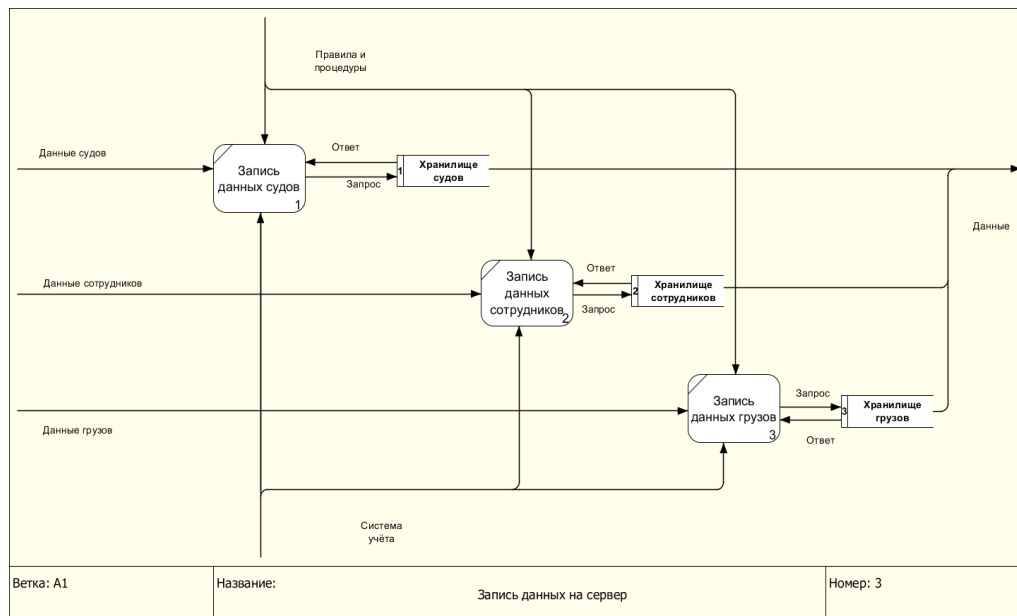


Рис. 4 – Третий уровень детализации (запись данных на сервер)

На рисунке 4 представлено детальное описание процесса записи данных на сервер базы данных; для разных типов данных существуют разные хранилища. Итоговые данные объединены в один поток данных.

На следующем шаге разработки информационной системы рассматриваются концептуальные наработки приложения, которые помогут вести систему учета работы морского порта. Например, форма учета сотрудников приведена на рисунке 5.

Рис. 5 – Форма учета сотрудников

В данной форме разработана функция прогнозирования увеличения заработной платы сотрудников. Администратор может выбрать из выпадающего списка имя сотрудника, ввести процент надбавки и указать промежуток времени, за которое будет осуществлено увеличение начисленных сумм. Для удобства пользователей создана функция поиска по должности. Также, осуществлена возможность отчетности о средней зарплате сотрудников по каждой из должности, ниже приведен код, с помощью которого подгружаются данные из таблицы сотрудников.

```
private void Form2_Load(object sender, EventArgs e)
{
    SqlCommand command = new SqlCommand("SELECT
DISTINCT Name FROM Staff", database.getConnection());
    database.openConnection();
    SqlDataReader reader = command.ExecuteReader();
    while (reader.Read())
    {
        comboBox1.Items.Add(reader["Name"]);
    }
}
```

```
    }  
    database.closeConnection();  
}
```

Принцип состоит в том, что при каждой загрузке страницы данные автоматически подгружаются из базы данных. При добавлении новых должностей в таблицу сотрудники, они также будут загружены.

```
private void button2_Click(object sender, EventArgs e)  
{  
    DataTable dataTable = new DataTable();  
    dataTable.Columns.Add("Должность", typeof(string));  
    dataTable.Columns.Add("Средняя зарплата", typeof(decimal));  
    string queryString = @"SELECT Position, AVG(Salary) AS  
AverageSalary FROM Staff GROUP BY Position";  
    SqlCommand command = new SqlCommand(queryString,  
database.getConnection());  
    database.openConnection();  
    SqlDataReader reader = command.ExecuteReader();  
    while (reader.Read())  
    {  
        DataRow row = dataTable.NewRow();  
        row["Должность"] = reader["Position"];  
        row["Средняя зарплата"] = reader["AverageSalary"];  
        dataTable.Rows.Add(row);  
    }  
    dataGridView1.DataSource = dataTable;  
    database.closeConnection();  
}
```

Форма позволяет отразить расчет средней заработной платы по каждой должности.

Рассмотрим расчет итоговой суммы выплат за период с учетом увеличения зарплаты для конкретного сотрудника, выбранного из списка.

```
private void button3_Click(object sender, EventArgs e)  
{  
    decimal incrementRate;  
    if (!decimal.TryParse(textBox2.Text, out incrementRate))  
    {  
        MessageBox.Show("Введите корректное значение прибавки!");  
        return;  
    }  
}
```

```
string selectedName = comboBox1.SelectedItem?.ToString();
if (string.IsNullOrEmpty(selectedName))
{
    MessageBox.Show("Выберите сотрудника!");
    return;
}
DateTime startDate = dateTimePickerStart.Value.Date;
DateTime endDate = dateTimePickerEnd.Value.Date;
if (endDate < startDate)
{
    MessageBox.Show("Дата окончания должна быть позже даты
начала!");
    return;
}
int months = ((endDate.Year - startDate.Year) * 12) + endDate.Month -
startDate.Month;
DataTable dataTable = new DataTable();
dataTable.Columns.Add("Имя", typeof(string));
dataTable.Columns.Add("Должность", typeof(string));
dataTable.Columns.Add("Текущая зарплата", typeof(decimal));
dataTable.Columns.Add("Прибавка", typeof(decimal));
dataTable.Columns.Add("Итоговая сумма выплат за период",
typeof(decimal));
string queryString = @"SELECT Name, Position, Salary, Salary *
@IncrementRate AS Increment,
(Salary + (Salary * @IncrementRate)) * @Months AS
TotalPaymentForPeriod
FROM Staff
WHERE Name = @Name";
SqlCommand command = new SqlCommand(queryString,
database.getConnection());
command.Parameters.AddWithValue("@IncrementRate", incrementRate
/ 100);
command.Parameters.AddWithValue("@Name", selectedName);
command.Parameters.AddWithValue("@Months", months);
database.openConnection();
SqlDataReader reader = command.ExecuteReader();
while (reader.Read())
{
    DataRow row = dataTable.NewRow();
    row["Имя"] = reader["Name"];
    row["Должность"] = reader["Position"];
    row["Текущая зарплата"] = reader["Salary"];
```



```

        row["Прибавка"] = reader["Increment"];
        row["Итоговая сумма выплат за период"] =
reader["TotalPaymentForPeriod"];
        dataTable.Rows.Add(row);
    }
    dataGridView1.DataSource = dataTable;
    database.closeConnection();
}

```

Код реализует функциональную работу с данными сотрудников.

Форма учета судов приведена на рисунке 6.

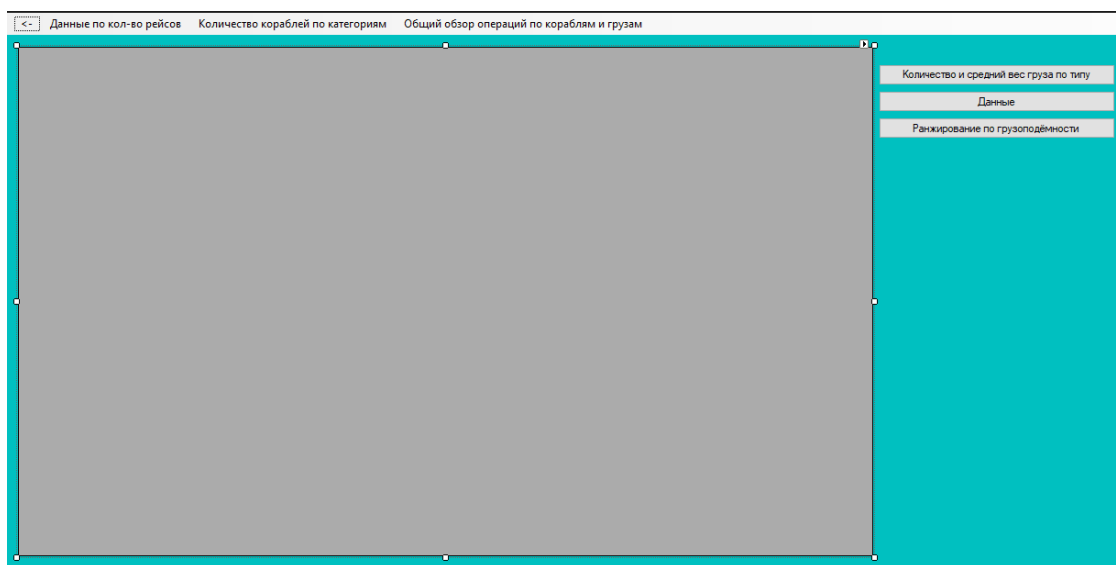


Рис. 6 – Форма учета судов

Представленный макет окна отражает возможность учета судов, формирование отчетности по кораблям, отражение маршрутов следования, ранжирования кораблей по тоннажу.

```

private void button2_Click(object sender, EventArgs e)
{
    DataTable dataTable = new DataTable();
    dataTable.Columns.Add("ShipName", typeof(string));
    dataTable.Columns.Add("ShipType", typeof(string));
    dataTable.Columns.Add("Capacity", typeof(int));
    dataTable.Columns.Add("Owner", typeof(string));
    dataTable.Columns.Add("DepartureDate", typeof(DateTime));
    dataTable.Columns.Add("ArrivalDate", typeof(DateTime));
    dataTable.Columns.Add("Origin", typeof(string));
    dataTable.Columns.Add("Destination", typeof(string));
}

```

```

string query = @"SELECT s.Name AS ShipName, s.Type AS ShipType,
s.Capacity, s.Owner,
    v.DepartureDate, v.ArrivalDate, v.Origin, v.Destination
FROM Ships s
JOIN Voyages v ON s.ShipID = v.ShipID";
SqlCommand command = new SqlCommand(query,
database.getConnection());
database.openConnection();
SqlDataReader reader = command.ExecuteReader();
while (reader.Read())
{
    DataRow row = dataTable.NewRow();
    row["ShipName"] = reader["ShipName"];
    row["ShipType"] = reader["ShipType"];
    row["Capacity"] = reader["Capacity"];
    row["Owner"] = reader["Owner"];
    row["DepartureDate"] = reader["DepartureDate"];
    row["ArrivalDate"] = reader["ArrivalDate"];
    row["Origin"] = reader["Origin"];
    row["Destination"] = reader["Destination"];
    dataTable.Rows.Add(row);
}
dataGridView1.DataSource = dataTable;
database.closeConnection();
}

```

Код направлен на получение данных о рейсах, включая данные о кораблях.

```

private void button3_Click(object sender, EventArgs e)
{
    DataTable dataTable = new DataTable();
    dataTable.Columns.Add("ShipType", typeof(string));
    dataTable.Columns.Add("TotalCargoes", typeof(int));
    dataTable.Columns.Add("AverageCargoWeight", typeof(decimal))
    string query = @"
        SELECT
            s.Type AS ShipType,
            COUNT(c.CargoID) AS TotalCargoes,
            AVG(c.Weight) AS AverageCargoWeight
        FROM Ships s
        JOIN Cargo c ON s.ShipID = c.ShipID
        GROUP BY s.Type;";
}

```

```

using (SqlCommand command = new SqlCommand(query,
database.getConnection()))
{
    try
    {
        database.openConnection();
        SqlDataReader reader = command.ExecuteReader();
        while (reader.Read())
        {
            DataRow row = dataTable.NewRow();
            row["ShipType"] = reader["ShipType"].ToString();
            row["TotalCargoes"] = Convert.ToInt32(reader["TotalCargoes"]);
            row["AverageCargoWeight"] =
            Convert.ToDecimal(reader["AverageCargoWeight"]);
            dataTable.Rows.Add(row);
        }
        reader.Close();
        dataGridView1.DataSource = dataTable;
    }
    catch (Exception ex)
    {
        MessageBox.Show("Ошибка при выполнении запроса: " +
ex.Message);
    }
    finally
    {
        database.closeConnection();
    }
}
}

```

Запрос получает информацию о типах грузах из соответствующей таблице в базе данных.

```

string query = @"
SELECT
    s.Type AS ShipType,
    COUNT(c.CargoID) AS TotalCargoes,
    AVG(c.Weight) AS AverageCargoWeight
FROM Ships s
JOIN Cargo c ON s.ShipID = c.ShipID
GROUP BY s.Type;";

```

Рассмотрим запрос, позволяющий получать данные об общем количестве груза, сгруппированных по типу корабля.

```
using (SqlCommand command = new SqlCommand(query,
database.getConnection()))
{
    try
    {
        database.openConnection();
        SqlDataReader reader = command.ExecuteReader();

        while (reader.Read())
        {
            DataRow row = dataTable.NewRow();
            row["ShipType"] = reader["ShipType"].ToString();
            row["TotalCargoes"] = Convert.ToInt32(reader["TotalCargoes"]);
            row["AverageCargoWeight"] =
            Convert.ToDecimal(reader["AverageCargoWeight"]);
            dataTable.Rows.Add(row);
        }
        reader.Close();
        dataGridView1.DataSource = dataTable;
    }
    catch (Exception ex)
    {
        MessageBox.Show("Ошибка при выполнении запроса: " + ex.Message);
    }
    finally
    {
        database.closeConnection();
    }
}
```

Приведенный код отражает создание соединения и выполнение запроса, в цикле считываются строки данных, после чтения данных подключение закрывается. Сам код обрабатывает нажатие кнопки, выполняя тем самым запрос к базе данных для получения из нее информации о грузах.

Заключение. Создание базы данных для управления морским портом с использованием SQL Server Management Studio (SSMS)

представляет собой масштабный и многоаспектный проект, который требует глубоких знаний в области баз данных и умения интегрировать различные типы информации в единую систему.

В ходе разработки и проектирования информационной системы были определены ключевые сущности, такие как суда, грузы, рейсы, персонал, портовые и финансовые операции, и разработаны соответствующие таблицы для их учета. Уделено внимание оптимизации производительности запросов, что является важным для систем с большим объемом данных и высокими требованиями к времени отклика.

Ключевой частью проекта стала реализация внешних ключей и связей между таблицами, что позволило обеспечить целостность данных и их последовательное обновление. Индексация выбранных столбцов способствовала ускорению поиска и сортировки данных, что повысило общую производительность системы.

В процессе разработки были использованы такие возможности SQL Server, как процедуры, триггеры и функции для автоматизации рутинных операций и минимизации возможности ошибок. Эти элементы значительно упрощают поддержку и масштабирование системы, делая ее гибкой и адаптивной к изменяющимся условиям эксплуатации.

Таким образом, созданная информационная система является мощным инструментом управления ресурсами морского порта, способным обеспечить его эффективное и бесперебойное функционирование.

Список использованной литературы:

1. SQL. Сборник рецептов. – 2-е изд. : Пер. с англ. / Э. Молинаро, Р. де Грааф. – СПб. : БХВ-Петербург, 2023. – 592 с.
2. Изучаем SQL. Генерация, выборка и обработка данных / А. Болье, 3-е изд.: Пер. с англ. – СПб. : ООО «Диалектика», 2021. – 400 с.
3. Настройка производительности MySQL : Пер. с англ. / Д. Нихтер. – СПб.: БХВ-Петербург, 2023. – 336 с.

References

1. SQL. Sbornik receptov. – 2-e izd. : Per. s angl. / Je. Molinaro, R. de Graaf. – SPb. : BHV-Peterburg, 2023. – 592 s.
2. Izuchaem SQL. Generacija, vyborka i obrabotka dannyh / A. Bol'e, 3-e izd.: Per. s angl. – SPb. : OOO «Dialektika», 2021. – 400 s.
3. Nastrojka proizvoditel'nosti MySQL : Per. s angl. / D. Nihter. – SPb.: BHV-Peterburg, 2023. – 336 s.