

УДК 004.65

UDC 004.65

5.2.2. – Математические, статистические и инструментальные методы в экономике

5.2.2 – Mathematical, statistical and instrumental methods in economics

РАЗРАБОТКА ПРОГРАММНОГО ПРИЛОЖЕНИЯ АВТОМАТИЗАЦИИ УЧЕТА АВИАТЕХНИКИ ЛЕТНОЙ ШКОЛЫ

DEVELOPMENT OF A SOFTWARE APPLICATION FOR AUTOMATION OF ACCOUNTING OF AIRCRAFT EQUIPMENT OF A FLIGHT SCHOOL

Рыбянцева 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

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

Сетраков Никита Михайлович

студент группы ИТ2301

email: nsetrakov@inbox.ru

ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13

Setrakov Nikita Mihailovich

student of group IT2301

email: nsetrakov@inbox.ru

Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13

В качестве предметной области в данной статье использована летная школа с собственным аэродромом. База данных в разработанной информационной системе представляет собой 11 связанных между собой таблиц. Распределим таблицы по их применению в предметной области: диспетчер аэродрома: Flightlist, Aircrafts, BriefDir, TypicalBriefing; метеоданные: Flightlist, Aircrafts; стоянка авиатехники: Aircrafts, Places; обслуживание авиатехники: Aircrafts, Orders, People, склад: Basement; данные самолета: Aircrafts, Clients. Естественно, что разработанная база данных содержит ограниченный список транспорта и ограниченный вид маршрутов, который будет намного упрощен по сравнению с теми полетами, которые происходят в реальных условиях. Целью данного исследования является разработка информационной системы с целью автоматизации работы аэродрома с использованием интеграции информационных систем. Основной акцент был сделан на автоматизации учета технического обслуживания авиатехники и диспетчеризации полетов. Внедрение информационных технологий в бизнес-процессы, отвечающие за обслуживание пассажиров авиарейса и облегчение работы диспетчера, способствует уменьшению риска действия человеческого фактора и повышению безопасности полетов. При разработке данной информационной системы была использована программная платформа .NET Framework и СУБД MYSQL. Также была изучено и использовано на практике средство подключения клиентского приложения к СУБД

The subject area in this article is a flight school with its own airfield. The database in the developed information system is 11 interconnected tables. Let's distribute the tables according to their application in the subject area: – airfield dispatcher: Flightlist, Aircrafts, BriefDir, TypicalBriefing; – weather data: Flightlist, Aircrafts; – aircraft parking: Aircrafts, Places; – aircraft maintenance: Aircrafts, Orders, People, – warehouse: Basement; – aircraft data: Aircrafts, Clients. Naturally, the developed database contains a limited list of transport and a limited type of routes, which will be much simplified compared to those flights that occur in real conditions. The purpose of this study is to develop an information system for the purpose of automating the operation of an airfield using the integration of information systems. The main emphasis was placed on the automation of aircraft maintenance accounting and flight dispatching. The introduction of information technologies into business processes responsible for servicing airline passengers and facilitating the work of the dispatcher helps to reduce the risk of human factors and improve flight safety. The .NET Framework software platform and MYSQL DBMS were used in the development of this information system. The means of connecting the client application to the DBMS was also studied and used in practice

Ключевые слова: ИНФОРМАЦИОННАЯ СИСТЕМА, ЛЕТНАЯ ШКОЛА, ОКОННЫЕ ФУНКЦИИ

Keywords: INFORMATION SYSTEM, FLIGHT SCHOOL, WINDOW FUNCTIONS

<http://dx.doi.org/10.21515/1990-4665-211-017>

Летная школа предназначена для обучения основам полетов и должна располагать собственным аэродромом для осуществления практического обучения. Аэродром представляет собой отдельную территорию со своим воздушным пространством для полетов. Аэродром состоит комплекса инженерно-технических сооружений: взлетно-посадочной полосы (ВПП), командно-диспетчерского пункта (диспетчер), метеослужбы, средства радиолокационного, светотехнического и радиотехнического оборудования. Также могут располагаться инженерно-технические сооружения, обеспечивающие техническое обслуживание авиатехники. Среди них можно отметить площадки для стоянки на продолжительное время, ангары для технического обслуживания, резервуары с топливом для заправки, склад для хранения различных запчастей или же другого технического оборудования для обеспечения функционирования аэродрома.

База данных в разработанной информационной системе представляет собой 11 связанных между собой таблиц, включающий системы автоматизации ведения таблиц при помощи триггеров.

Диаграмма сущность-связь (Entity-Relational Diagram) представлена на рисунке 1.

Распределим таблицы по их применению в предметной области:

- диспетчер аэродрома: Flightlist, Aircrafts, BriefDir, TypicalBriefing;
- метеоданные: Flightlist, Aircrafts;
- стоянка авиатехники: Aircrafts, Places;
- обслуживание авиатехники: Aircrafts, Orders, People,
- склад: Basement;
- данные самолета: Aircrafts, Clients.

<http://ej.kubagro.ru/2025/07/pdf/17.pdf>


```
abs((ceiling(avg(`kurs`.`tablewithemp`.`count`) OVER () ) -
`kurs`.`tablewithemp`.`count`)) AS `count`
from
`kurs`.`tablewithemp`
where
(`kurs`.`tablewithemp`.`emp` is not null);
```

Результат представлен на рисунке 2.

	123 emp ▼	123 count ▼
1	1	0
2	2	1
3	3	1
4	4	2
5	5	2

Рисунок 2 – Результат оконной функции 1

2. Листинг:

```
CREATE OR REPLACE
ALGORITHM = UNDEFINED VIEW `kurs`.`findlazyemp` AS
select
`o`.`opportunity` AS `emp`,
count(`o`.`id`) AS `count`
from (`kurs`.`People` `p`
left join `kurs`.`Orders` `o` on
((`p`.`id` = `o`.`opportunity`)))
group by
`o`.`opportunity`
order by
`count`;
```

Результат представлен на рисунке 3.

	123 emp	123 count
1	4	1
2	5	1
3	3	2
4	1	3
5	2	4

Рисунок 3 – Результат оконной функции 2

3. Листинг:

```
CREATE OR REPLACE
```

```
ALGORITHM = UNDEFINED VIEW `kurs`.`itog` AS
```

```
Select row_number() OVER () AS `cnt`,
```

```
`kurs`.`p`.`emp` AS `emp`,
```

```
`kurs`.`p2`.`id` AS `id`
```

```
from
```

```
(`kurs`.`parts2` `p2`
```

```
join `kurs`.`parts` `p` on
```

```
((`kurs`.`p`.`place` = `kurs`.`p2`.`place`)));
```

Результат представлен на рисунке 4.

	123 cnt	123 emp	123 id
1	1	4	14
2	2	5	15
3	3	2	16
4	4	4	17
5	5	5	18
6	6	2	19
7	7	4	20

Рисунок 4 – Результат оконной функции 3

4. Листинг:

```
CREATE OR REPLACE
```

```
ALGORITHM = UNDEFINED VIEW `kurs`.`parts` AS with `sumcte` as
```

```
(
```

```
Select row_number() OVER (
```

```

ORDER BY `kurs`.`summ`.`count` desc ) AS `id`,
`kurs`.`summ`.`emp` AS `emp`
From `kurs`.`summ`),
`partscte` as (
Select row_number() OVER () AS `id`,
`kurs`.`parts2`.`place` AS `place`,
count(`kurs`.`parts2`.`id`) AS `count(id)`
from `kurs`.`parts2`
group by `kurs`.`parts2`.`place`)
select `p`.`place` AS `place`,
`s`.`emp` AS `emp`
From (`partscte` `p`
join `sumcte` `s` on
((`s`.`id` = `p`.`id`)));

```

Результат представлен на рисунке 5.

	123 place ▼	123 emp ▼
1	2	4
2	3	5
3	1	2

Рисунок 5 – Результат оконной функции 4

5. Листинг:

```

CREATE OR REPLACE
ALGORITHM = UNDEFINED VIEW `kurs`.`parts2` AS with `cte` as (
Select `kurs`.`Orders`.`id` AS `id`,
(select `kurs`.`parts4`.`count(num)`
From `kurs`.`parts4`) AS `lim`
From `kurs`.`Orders`
Where (`kurs`.`Orders`.`opportunity` is null))
Select `cte`.`id` AS `id`,

```

```
((row_number() OVER () % `cte`.`lim`) + 1) AS `place`
From `cte`;
```

Результат представлен на рисунке 6.

	123 id	123 place
1	14	2
2	15	3
3	16	1
4	17	2
5	18	3
6	19	1
7	20	2

Рисунок 6 – Результат оконной функции 5

6. Листинг:

```
CREATE OR REPLACE
```

```
ALGORITHM = UNDEFINED VIEW `kurs`.`parts4` AS with recursive
```

```
`parts3` as (Select 1 AS `num`,
```

```
(select ceiling(`kurs`.`summ`.`count`)
```

```
from `kurs`.`summ`
```

```
limit 1) AS `count`,
```

```
(select ceiling(`kurs`.`summ`.`count`)
```

```
from `kurs`.`summ`
```

```
limit 1) AS `sm`
```

```
union all
```

```
select (`p`.`num` + 1) AS `p.num+1`,
```

```
ceiling(`kurs`.`s`.`count`) AS `count`,
```

```
(`p`.`sm` + ceiling(`kurs`.`s`.`count`)) AS `sm`
```

```
From (`parts3` `p`
```

```
join `kurs`.`summ` `s` on
```

```
((`kurs`.`s`.`id` = `p`.`num`))))
```

```
Select count(`parts3`.`num`) AS `count(num)` From `parts3`
```

```
Where ((`parts3`.`count` > 0)
```

```
and (`parts3`.`sm` <= (
```

```
select count(`kurs`.`Orders`.`id`)
from `kurs`.`Orders`
where (`kurs`.`Orders`.`opportunity` is null));
```

Результат представлен на рисунке 7.

	123 count(num) ▼
1	3

Рисунок 7 – Результат оконной функции 6

7. Листинг:

```
CREATE OR REPLACE
ALGORITHM = UNDEFINED VIEW `kurs`.`summ` AS
Select row_number() OVER () AS `id`,
`kurs`.`countfind`.`emp` AS `emp`,
((select distinct count(`kurs`.`Orders`.`id`) OVER ()
From `kurs`.`Orders`
Where (`kurs`.`Orders`.`opportunity` is null)) * (`kurs`.`countfind`.`count`
/ sum(`kurs`.`countfind`.`count`) OVER ())) AS `count`
From `kurs`.`countfind`
order by `count`;
```

Результат представлен на рисунке 8.

	123 id ▼	123 emp ▼	123 count ▼
1	1	1	0
2	2	2	1.1667
3	3	3	1.1667
4	4	4	2.3333
5	5	5	2.3333

Рисунок 8 – Результат оконной функции 7

8. Листинг:

```
CREATE OR REPLACE
ALGORITHM = UNDEFINED VIEW `kurs`.`tablewithemp` AS
```

```

Select `p`.`id` AS `emp`,
count(`o`.`id`) AS `count`
from (`kurs`.`Orders` `o`
left join `kurs`.`People` `p` on
(((`p`.`id` = `o`.`opportunity`) and (`p`.`role` in (1, 2)))))
group by `p`.`id`
union
select `p`.`id` AS `id`,
count(`o`.`id`) AS `count(o.id)`
from (`kurs`.`People` `p`
left join `kurs`.`Orders` `o` on
(((`p`.`id` = `o`.`opportunity`) and (`p`.`role` in (1, 2)))))
group by `p`.`id`;

```

Результат представлен на рисунке 9.

	123 emp	123 count
1	[NULL]	7
2	1	3
3	2	4
4	3	2
5	4	1
6	5	1

Рисунок 9 – Результат оконной функции 8

В разработанном приложении клиента были реализованы все функции, обеспечивающие автоматизированное управление базой данных в предприятии. Всего в системе реализовано 10 форм. 1 – родительская, остальные – дочерние. Среди этих форм можно отметить 5 основных, так как остальные являются лишь простыми программами ведения таблиц. Рассмотрим эти 5 окон отдельно.

Главное окно имеет минималистичный интерфейс, в котором представлены кнопки вызова нужного ему окна. Вспомогательные окна не могут иметь дубликаты.

Все формы, в которых производится автоматизированное управление системой, обобщены в 4 подгруппы: Полеты, Стоянка, Обслуживание и Сотрудники.

Каждое из окон в приложении масштабируемо и способно изменять свой размер под предпочтения клиента.

Окно летного расписания вызывается при помощи кнопки «Летное расписание», расположенной во вкладке «Полеты». Данное окно представляет собой таблицу, в которой изображены необходимые сотруднику данные, и элементы управления, при помощи которых сотрудник добавляет, удаляет или корректирует необходимые ему данные.

Основными элементами управления базой данных будет являться ввод данных (кнопка со знаком – «+»), удаление (кнопка со знаком – «-») и обновление данных (кнопка с надписью «UPDT»).

Особенностью данной реализации является то, что при любом виде взаимодействия с таблицей, данные сразу же отправляются в базу данных. В некоторых формах система заранее предупреждает, что будут вноситься данные и дополнительно уведомляет пользователя о производимых изменениях. Так же система может спросить у пользователя согласие на производимые изменения.

Среди особенностей можно отметить, что данный внешний вид окна будет являться стандартным почти для всех остальных окон, что обеспечивает универсальность системы. И возможность его легкой модификации.

Окно ввода METAR обеспечивает более автоматизированное средство ввода метеоданных в систему управления полетами. Представляет оно собой конструктор итогового сообщения методики METAR, в которой необходимо лишь вставить соответствующие значения. Система сама интегрирует эти данные в выбранные пользователем позиции в летном расписании.

Данное окно вызывается при помощи кнопки «Ввод метео», находящийся во вкладке «Полеты».

Внешний вид данного окна представлен на рисунке 10.

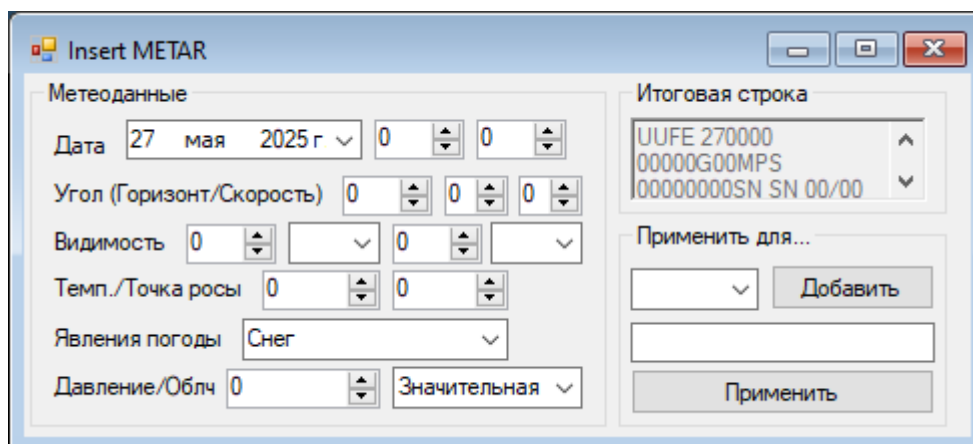


Рисунок 10 – Внешний вид окна ввода METAR

Данное окно позволяет пользователю, а точнее, диспетчеру, более наглядно следить за маршрутом, его соответствие летному плану в брифинге и общие динамические характеристики самолета в данный момент. Данное окно представляет собой развертку карты, в которой пользователь вводит координаты при помощи мыши и вводит динамические параметры в данной координате.

Это позволяет более наглядно отслеживать прогресс учебных полетов в ходе обучения.

Также точкам можно изменять данные, удалять их и отмечать как «завершенные», блокируя при этом дальнейшее редактирование.

Внешний вид окна представлен на рисунке 11.

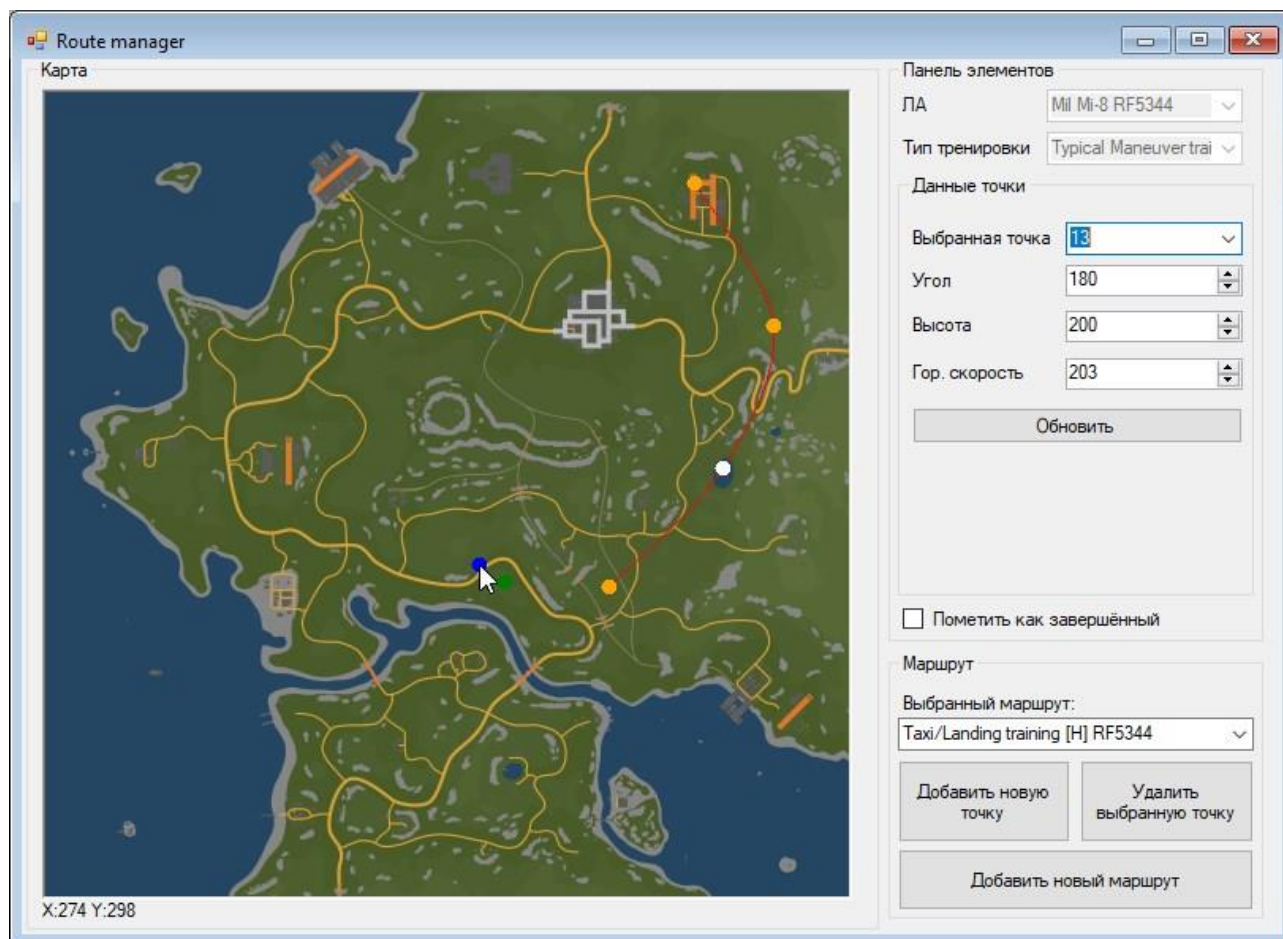


Рисунок 11 – Внешний вид окна ведения маршрутов

Окно управления заказами является более удобным способом управления заказ-нарядами, которые оформляются при техническом обслуживании авиатехники. В данном окне производится управление, как заказом, так и его отдельных элементов, поэтому в качестве компромиссного решения был реализован селектор. То есть, в случае, если пользователь желает создать новый заказ, то перед созданием системе необходимо указать, что пользователь желает создать новый заказ, а не редактировать существующий.

Данный формат ведения заказа позволяет пользователю видеть общее положение заказа. Т.е. была попытка создания некоего «монолитного приложения», в котором все функции инкапсулированы в одно окно.

Основное отличие данного окна от остальных является сложное управление целым пластом сущностей, к которому он обращается, в то время как остальные обращаются в основном к своей основной сущности, функцию автоматизации которой и производит.

В результате проведенного исследования была реализована информационная система автоматизации учета авиатехники, что доказывает, что его разработка вполне возможна и обоснована в реальных эксплуатационных условиях.

В качестве атрибутов разработки данной информационной системы была использована программная платформа .NET Framework и СУБД MYSQL. Также была изучено и использовано на практике средство подключения клиентского приложения к СУБД.

Можно сказать, что информационная система с интеграцией баз данных на данный момент является стандартной практикой, применяемых во многих компаниях, но эта «база» будет оставаться в системе довольно долго и постоянно совершенствоваться на базе данной технической архитектуры.

Список использованной литературы:

1. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. : учебник / Д. Рихтер - 4-е изд. — СПб.: Питер, 2024 — 896 с.
2. MySQL по максимуму : учебное пособие / С. Ботрос, Д. Тинли — 4-е изд. — СПб : Питер, 2023. — 432 с.
3. Грокам Алгоритмы : учебное пособие / А. Бхаргава, Е.П. Матвеев — 2-е изд. — СПб: Питер, 2025. — 352 с.
4. SQL. Pocket Guide : учебник / Э. Жао — 4-е изд. — Казахстан: Sprint Book, 2025. — 320 с.
5. C#. Основы программирования : учебное пособие / С. Наков — Москва : АСТ, 2024. — 498 с.

References:

1. CLR via C#. Programmirovaniye na platforme Microsoft .NET Framework 4.5 na jazyke S#. : uchebnik / D. Rihter - 4-e izd. — SPb.: Piter, 2024 — 896 s.
2. MySQL po maksimumu : uchebnoye posobie / S. Botros, D. Tinli — 4-e izd. — SPb : Piter, 2023. — 432 s.

3. Grokaem Algoritmy : uchebnoe posobie / A. Bhargava, E.P. Matveev – 2-e izd. – SPb: Piter, 2025. – 352 с.
4. SQL. Pocket Guide : uchebnik / Je. Zhao – 4-e izd. – Kazakhstan: Sprint Book, 2025. – 320 с.
5. C#. Osnovy programmirovaniya : uchebnoe posobie / S. Nakov – Moskva : AST, 2024. – 498 с.