

КУРСОВА РОБОТА

з Баз даних
(назва дисципліни)
на тему: Розробка БД і застосунку для роботи з БД

Студента 3 курсу В31 групи
галузі знань

12. Інформаційні технології
спеціальності

121. Інженерія програмного забезпечення

Геока Євгенія Ілліча
(прізвище та ініціали)

Керівник Матвійчук Т.М.
(прізвище та ініціали)

Національна шкала _____

Члени комісії _____
(підпис) (прізвище та ініціали)

(підпис) (прізвище та ініціали)

Ізмаїльський Фаховий Коледж Економіки і Права

Рецензія про якість курсової роботи з дисципліни

« Бази даних »

Тема курсової роботи «Розробка БД і застосунку для роботи з БД»

Автор курсової роботи Геока Євгеній Ілліч

Спеціальність 121. Інженерія програмного забезпечення курс 3 група B31

Курсова робота виконана _____

Зауваження викладача _____

Критерії оцінки курсової роботи

1. Ступінь розробки теми _____
2. Повнота використання літератури _____
3. Використання спеціальної літератури _____
4. Творчий підхід до написання курсової роботи _____
5. Оформлення курсової роботи - _____

Попередня оцінка _____

Захист курсової роботи _____

Загальна оцінка _____

Керівник _____

(підпис)

(прізвище, ініціали)

« ____ » _____ 20__ р.

Зміст

Вступ	4
Опис обладнання та ПЗ	5
Конфігурація системи	5
ПЗ, яке було застосовано	5
Постановка задачі	6
Проектування БД	7
Проектування застосунку.....	10
Загальний опис вигляду форм.....	10
Система найменувань компонентів.....	11
Заходи для стійкості програми	11
Діаграма класів	12
Опис методів	15
Тестування роботи додатка.....	16
Тестові дані.....	16
Скріншоти роботи програми	18
Висновки	27
Перелік використаних інформаційних джерел	28
Додаток 1. Код програми	29
Info.cs	29
Main.cs	29

12.121.B31.05.KP.ПЗ

					12.121.B31.05.KP.ПЗ							
Изм.	Лист	№ Документа	Подпись	Дата								
Разработал					Розробка БД і застосунку для роботи з БД			Лит.	Лист	Листов		
Руковод.										3	43	
Н.Контр.								ІФКЕП				
Зав. Каф.												

Вступ

Використання різноманітних баз даних для збереження даних у наші часи становиться дедалі популярнішим. Базы даних дозволяють структуровано зберігати інформацію, що спрощує процес її обробки. Однак, БД не призначені для відображення даних у зручному вигляді. Тому існують такі застосунки, що звуться СУБД. Вони дозволяють створювати БД, здійснювати різноманітні операції з даними, які знаходяться в БД у вигляді таблиць; деякі дозволяють навіть створювати різноманітні засоби зручного відображення даних (тут прикладом є СУБД MS Access). Але у деяких випадках є потреба у застосунках, що створені спеціально для роботи із конкретною БД. Вони повинні повністю задовольняти функціональні потреби, але, так як вони створені спеціально для конкретної БД, можуть мати покращені показники комфорту роботи та деякі додаткові можливості, які неможливо чи недоцільно реалізувати в універсальних СУБД.

Завданням цієї курсової роботи як раз і є створення такого спеціалізованого застосунку.

Інколи мені доводилось звертатися за додатковою інформацією до різноманітних джерел для усвідомлення деяких теоретичних моментів або для пошуку цікавих практичних прийомів.

Я б виділив дві основні мети цієї курсової роботи:

- Виконати завдання у повному обсязі;
- Покращити мої навички та знання шляхом застосування якомога більшої кількості різноманітних прийомів з використанням пройдених протягом семестру теоретичних даних.

Опис обладнання та ПЗ

Конфігурація системи

Розробка й тестування застосунку здійснювалися на такій апаратній платформі:

- Процесор: AMD E-450 APU with Radeon(tm) HD Graphics 1.65 GHz;
- Оперативна пам'ять: 3,00 ГБ (+ 5,5ГБ файл підкачки);
- Відеоадаптер інтегрований у процесор.
- Операційна система: Windows 10 Pro

ПЗ, яке було застосовано

- База даних була спроектована, створена та заповнена тестовими даними у СУБД MS Access 2013
- Застосунок був розроблений у середовищі MS Visual Studio 2013 Common v12.0.40629.0 на платформі .NET v4.5. Мова програмування C#.

					12.121.B31.05.КР.ПЗ	Лист
						5
Изм.	Лист	№ Документа	Подпись	Дат		

Постановка задачі

Завдання №5: БД Радіостанції.

Таблиці:

- 1) Працівники;
- 2) Посади;
- 3) Виконавці;
- 4) Жанри;
- 5) Записи;
- 6) Графік роботи.

Виконати такі запити:

- 1) Відділ кадрів (Поєднує таблиці «Працівники» та «Посади» по полю «Код посади»);
- 2) Музичний архів (Поєднує таблиці «Записи», «Виконавці», «Жанри» по полям «Код виконавця» та «Код жанру»
- 3) Сітка мовлення (Поєднує таблиці «Графік роботи», «Працівники», «Записи» по полям «Код працівника», «Код запису», «Код запису1», «Код запису2», «Код запису3»).

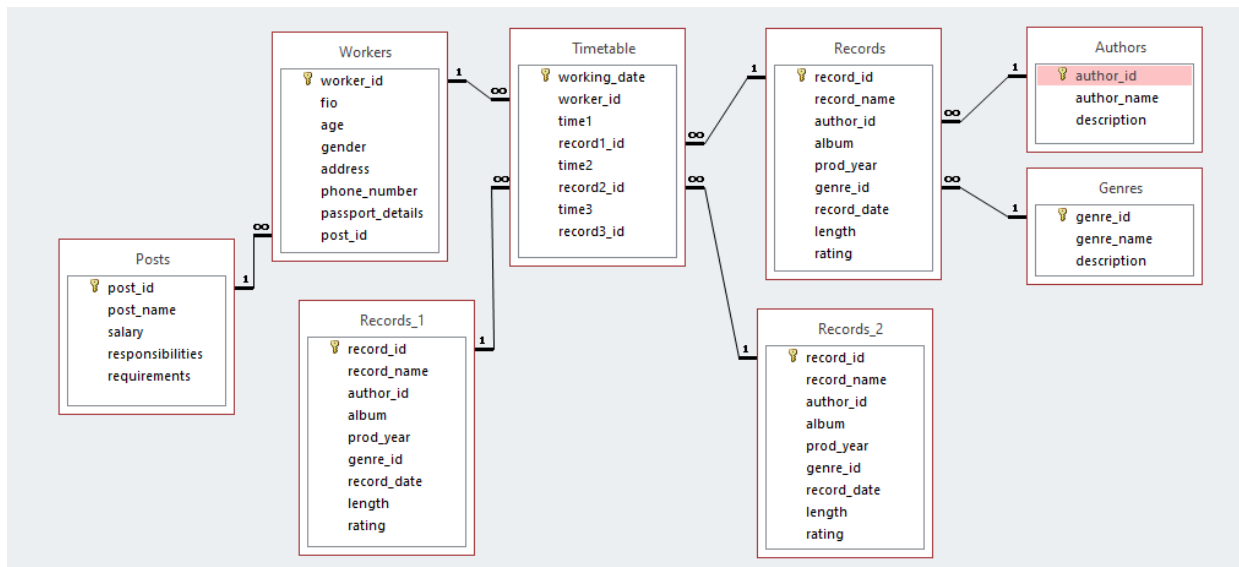
Передбачити наступні фільтри:

- 1) Фільтр для відображення працівників певних посад (На основі запиту «Відділ кадрів»);
- 2) Фільтр для відображення записів певного виконавця (На основі запиту «Музикальний архів»);
- 3) Фільтр для відображення записів певного жанру (На основі запиту «Музикальний архів»);
- 4) Фільтр сітки мовлення по окремим датам (На основі запиту «Сітка мовлення»);
- 5) Фільтр сітки мовлення по окремим працівникам (На основі запиту «Сітка мовлення»);

Розробити застосунок для роботи з цією БД.

Проектування БД

Так як я вирішив обрати англійську мову у якості мови інтерфейсу, назви таблиць та запитів теж англійськомовні.



Малюнок 1. Схема даних БД.

Records_1 та Records_2 є копіями Records. (Тільки шляхом створення цих копій зміг виконати запит «Сітка мовлення», де є досить складні зв'язки).

Працівники - Workers;
 Посади - Posts;
 Виконавці - Authors;
 Жанри - Genres;
 Записи - Records;
 Графік роботи – Timetable;
 Відділ кадрів – Personnel department;
 Музикальний архів – Music archive;;
 Сітка мовлення – Broadcast grid.

Таблиця Workers

Найменування	Ідентифікатор	Тип даних	Довжина	Порожні рядки	Ключ
Код працівника	worker_id	лічильник		-	+
ПІБ	fio	короткий текст	60	-	-
Вік	age	чисельний	байт	-	-

Стать	gender	короткий текст	6	-	-
Адрес	address	короткий текст	255	-	-
Номер телефону	phone_number	короткий текст	18	-	-
Паспортні дані	passport_details	короткий текст	255	-	-
Код посади	post_id	чисельний	довге ціле	-	-

Таблиця Posts

Найменування	Ідентифікатор	Тип даних	Довжина	Порожні рядки	Ключ
Код посади	post_id	лічильник		-	+
Назва	post_name	короткий текст	60	-	-
Зарплата	salary	чисельний	довге ціле	-	-
Обов'язки	responsibilities	короткий текст	255	-	-
Вимоги	requirements	короткий текст	255	-	-

Таблиця Authors

Найменування	Ідентифікатор	Тип даних	Довжина	Порожні рядки	Ключ
Код виконавця	author_id	лічильник		-	+
Назва	author_name	короткий текст	60	-	-
Опис	description	короткий текст	255	-	-

Таблиця Genres

Найменування	Ідентифікатор	Тип даних	Довжина	Порожні рядки	Ключ
Код жанру	genre_id	лічильник		-	+
Назва	genre_name	короткий текст	60	-	-

Опис	description	короткий текст	255	-	-
------	-------------	----------------	-----	---	---

Таблиця Records

Найменування	Ідентифікатор	Тип даних	Довжина	Порожні рядки	Ключ
Код запису	record_id	лічильник		-	+
Назва	record_name	короткий текст	60	-	-
Код автора	author_id	чисельний	довге ціле	-	-
Альбом	album	короткий текст	60	-	-
Рік релізу	prod_year	чисельний	ціле	-	-
Код жанру	genre_id	чисельний	довге ціле	-	-
Дата релізу(запису)	record_date	дата й час		-	-
Тривалість	length	дата й час		-	-
Рейтинг	rating	чисельний	double	-	-

Таблиця Timetable

Найменування	Ідентифікатор	Тип даних	Довжина	Порожні рядки	Ключ
Робоча дата	working_date	дата й час		-	+
Код працівника	worker_id	чисельний	довге ціле	-	-
Час1	time1	дата й час		-	-
Код запису1	record1_id	чисельний	довге ціле	-	-
Час2	time2	дата й час		-	-
Код запису2	record2_id	чисельний	довге ціле	-	-
Час3	time3	дата й час		-	-
Код запису3	record3_id	чисельний	довге ціле	-	-

Проектування застосунку

Загальний опис вигляду форм

Я вирішив реалізувати застосунок за допомогою якомога меншої кількості форм (вікон). В результаті маю форму Main, яка є основною формою додатку. Увесь функціонал реалізовано на ній, окрім відображення текстової інформації про додаток. Для цього була створена форма Info. Для відображення інформації про додаток використовуються компоненти Label та GroupBox. Більше нічого там немає.

Загальний вигляд головної форми під час перегляду таблиць має вигляд:

- 1) Заголовок таблиці; (один Label, текст якого відповідає назві поточної таблиці)
- 2) Компонент DataGridView; (Один на формі, заповнюється даними поточної таблиці)
- 3) Кнопки додавання/видалення/збереження даних
- 4) Компонент GroupBox, в якому знаходяться компоненти, що потрібні для додавання нових записів для кожної таблиці. (Таких груп 6, одночасно доступна тільки одна, відповідно до поточної таблиці)

Загальний вигляд головної форми під час перегляду запитів має вигляд:

- 1) Заголовок таблиці; (той же Label)
- 2) Компонент DataGridView;
- 3) Компонент GroupBox для фільтрів (містить 3 RadioButton та 2 Combobox)

Також у верхній частині вікна розташований компонент MenuStrip, який дозволяє зручно здійснювати навігацію між таблицями і запитами. Також там є можливість переглянути інформацію про додаток.

Під час перегляду запитів із активованим фільтром, пункт меню, в якому можна обрати запит для перегляду, заблокований. (Таке рішення зумовлене попередженням критичних та некритичних помилок під час виконання.) Для розблокування достатньо вимкнути фільтр.

Система найменувань компонентів

Зважаючи на те, що компонентів на одній формі багато, потрібно було запровадити зрозумілу систему найменувань компонентів, щоб не заплутатися.

Спеціальні компоненти на головній формі, які виконують одразу декілька задач чи є універсальними для різних таблиць і запитів, мають спеціальні назви. Вони є досить унікальними та легко запам'ятовуються.

Компоненти, що належать до груп додавання даних у певну таблицю, мають таку систему назв, яка забезпечує легкий пошук та навігацію.

Загальний вигляд імені такого компонента має такий вигляд:

<перша літера таблиці>_<призначення>_<скорочена назва компонента>

Наприклад, A_id_LB – компонент Label, який використовується у якості надписа для іншого компонента, що відповідає за введення даних в поле id таблиці Authors

W - Workers

P - Posts

T - Timetable

A - Authors

G- Genres

R – Records

LB – Label.

TB - TextBox

CB - ComboBox

RB – RadioButton

NUD – Numeric Up Down

DTP - DateTimePicker

Заходи для стійкості програми

Програма повинна бути стійкою, тому потрібна реалізація різноманітного захисту від речей, що призводять до критичних і некритичних помилок під час виконання.

З цією метою було реалізовано:

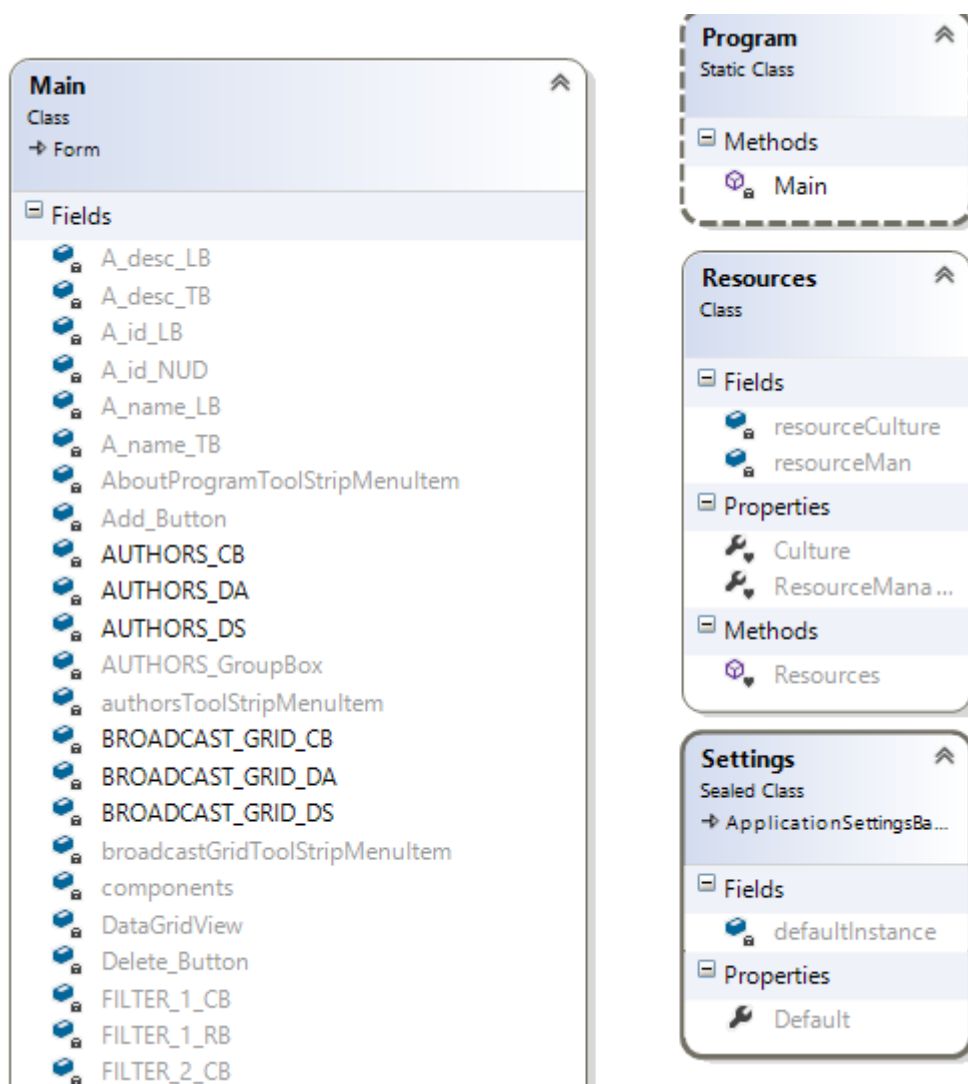
- Попередження, якщо файл БД не знаходиться за певним шляхом;
- Повідомлення про некоректність даних під час їх додавання у таблицю;
- Відключив можливість редагування тексту компонентів ComboBox в розділі фільтрів;
- Заборонив повторне відображення запитів, якщо програма зараз відображає запит із активованим фільтром. (Потрібно спочатку вимкнути фільтр);






























Додатково:






























Редагування тексту компонентів ComboBox під час додавання нових записів у БД не призводить до критичної помилки (Відповідне повідомлення відображається);

Якщо під час додавання нового запису в таблицю ключ буде збігатися із вже існуючим, то критичної помилки не виникне. Натомість, ключ самостійно отримає унікальне значення, завдяки типу даних «лічильник», який використовується в усіх таблицях для ключа.

Діаграма класів



-  FILTER_2_RB
-  FILTERS_GroupBox
-  G_desc_LB
-  G_desc_TB
-  G_id_LB
-  G_id_NUD
-  G_name_LB
-  G_name_TB
-  GENRES_CB
-  GENRES_DA
-  GENRES_DS
-  GENRES_GroupBox
-  genresToolStripMenuItem
-  Hint_LBL
-  menuStrip1
-  MUSIC_ARCHIVE_CB
-  MUSIC_ARCHIVE_DA
-  MUSIC_ARCHIVE_DS
-  musicArchiveToolStripMenuItem
-  NO_FILTERS_RB
-  P_id_LB
-  P_id_NUD
-  P_name_LB
-  P_name_TB
-  P_req_LB
-  P_req_TB
-  P_resp_LB
-  P_resp_TB
-  P_salary_LB

-  P_salary_NUD
-  path
-  PERSONNEL_DEPARTMENT_CB
-  PERSONNEL_DEPARTMENT_DA
-  PERSONNEL_DEPARTMENT_DS
-  personnelDepartmentToolStripMenuItem
-  POSTS_CB
-  POSTS_DA
-  POSTS_DS
-  POSTS_GroupBox
-  postsToolStripMenuItem
-  QueriesToolStripMenuItem
-  R_album_LB
-  R_album_TB
-  R_author_CB
-  R_author_LB
-  R_date_DTP
-  R_date_LB
-  R_genre_CB
-  R_genre_LB
-  R_id_LB
-  R_id_NUD
-  R_length_DTP
-  R_length_LB
-  R_name_LB
-  R_name_TB
-  R_rating_LB
-  R_rating_NUD
-  RECORDS_CB

- RECORDS_DA
- RECORDS_DS
- RECORDS_GroupBox
- recordsToolStripMenuItem
- RS_DB_Connection
- Save_Button
- T_date_DTP
- T_date_LB
- T_record1_CB
- T_record1_LB
- T_record2_CB
- T_record2_LB
- T_record3_CB
- T_record3_LB
- T_time1_DTP
- T_time1_LB
- T_time2_DTP
- T_time2_LB
- T_time3_DTP
- T_time3_LB
- T_worker_CB
- T_worker_LB
- Table_Name_Label
- TablesToolStripMenuItem
- TIMETABLE_CB
- TIMETABLE_DA
- TIMETABLE_DS
- TIMETABLE_GroupBox
- timetableToolStripMenuItem

- W_address_LB
- W_address_TB
- W_age_LB
- W_age_NUD
- W_fio_LB
- W_fio_TB
- W_gender_CB
- W_gender_LB
- W_id_LB
- W_id_NUD
- W_number_TB
- W_passport_LB
- W_passport_TB
- W_phone_LB
- W_post_CB
- W_post_LB
- WORKERS_CB
- WORKERS_DA
- WORKERS_DS
- WORKERS_GroupBox
- workersToolStripMenuItem

Methods

- AboutProgramToolStripMenuItem_Click
- Add_Button_Click
- authorsToolStripMenuItem_Click
- broadcastGridToolStripMenuItem_Click
- ComboBox_Fill
- comboBox1_SelectedIndexChanged
- comboBox2_SelectedIndexChanged

- DataGrid_Fill
- Delete_Button_Click
- Dispose
- DT_Cell_Date_Truncate
- DT_Cell_Time_Truncate
- FILTER_1_CB_KeyPress
- FILTER_1_RB_CheckedChanged
- FILTER_2_CB_KeyPress
- FILTER_2_RB_CheckedChanged
- genresToolStripMenuItem_Click
- Get_id
- InitializeComponent
- Main
- Main_Load
- musicArchiveToolStripMenuItem_Click
- NO_FILTERS_RB_CheckedChanged
- personnelDepartmentToolStripMenuItem_...
- postsToolStripMenuItem_Click
- recordsToolStripMenuItem_Click
- Reset
- Save_Button_Click
- Table_Remove
- timetableToolStripMenuItem_Click
- workersToolStripMenuItem_Click

Info

Class

→ Form

Малюнок 2. Діаграма класів.

На жаль, мій підхід до виконання на одній формі також має недоліки. ВЕЛИЧЕЗНА діаграма класів, яку ну просто неможливо вставити в документ не по частинах.

Опис методів

Параметри пропущені.

Reset() – встановлює початкові значення компонентам і змінним.

DataGrid_Fill() – дозволяє заповнити DataGridView даними із DataSet.

DT_Cell_Date_Truncate() – дозволяє видалити непотрібну дату, залишивши тільки час (01.01.2022 17:00:00 -> 17:00:00).

DT_Cell_Time_Truncate() – дозволяє видалити непотрібний час, залишивши тільки дату (01.01.2022 17:00:00 -> 01.01.2022).

ComboBox_Fill() – дозволяє заповнити ComboBox даними певного стовпчика певної таблиці із DataSet.

Get_id() – дозволяє «витягти» id по значенню поля (назви чогось).

Table_Remove() – дозволяє видалити всі виділені рядки із DataSet і DataGridView.

USB-накопитель (F:) > Учебные документы Жени > Курсовая 3 курс 2 семестр > RS_DB_Editor > RS_DB_Editor				
Имя	Дата изменения	Тип	Размер	
bin	01.05.2022 22:11	Папка с файлами		
obj	01.05.2022 22:11	Папка с файлами		
Properties	01.05.2022 22:11	Папка с файлами		
App	01.05.2022 22:11	XML Configuratio...	1 КБ	
ClassDiagram1.cd	09.05.2022 19:47	Class Diagram file	2 КБ	
Info.cs	09.05.2022 20:25	Visual C# Source f...	1 КБ	
Info.Designer.cs	09.05.2022 20:25	Visual C# Source f...	11 КБ	
Info	09.05.2022 20:25	.NET Managed Re...	6 КБ	
Main.cs	09.05.2022 21:09	Visual C# Source f...	39 КБ	
Main.Designer.cs	09.05.2022 19:34	Visual C# Source f...	66 КБ	
Main	09.05.2022 19:32	.NET Managed Re...	7 КБ	
Program.cs	07.05.2022 2:35	Visual C# Source f...	1 КБ	
RS_DB_Editor	09.05.2022 20:24	Visual C# Project f...	5 КБ	

Малюнок 3. Каталог проекту.

USB-накопитель (F:) > Учебные документы Жени > Курсовая 3 курс 2 семестр > RS_DB_Editor > RS_DB_Editor > bin > Debug

Имя	Дата изменения	Тип	Размер
RS_DB	10.05.2022 6:34	Microsoft Access ...	1 280 КБ
RS_DB_Editor	09.05.2022 21:09	Приложение	54 КБ
RS_DB_Editor.exe	01.05.2022 22:11	XML Configuratio...	1 КБ
RS_DB_Editor	09.05.2022 21:09	Program Debug D...	58 КБ
RS_DB_Editor.vshost	09.05.2022 21:10	Приложение	23 КБ
RS_DB_Editor.vshost.exe	01.05.2022 22:11	XML Configuratio...	1 КБ

Малюнок 4. Місцезнаходження БД.

Тестування роботи додатка

Тестові дані

worker_id	fio	age	gender	address	phone_num	passport_details	post_id
1	Myroslav Adamovych Hryshko	25	Male	Zhitomirskaya	+380-273-855-5	S34240-SFSDF-234-DF	5
2	Oleh Kuzmich Demydenko	19	Male	Kharkovskaya	+380-395-558-2	OOSDJNK-DKKKS-DDG	5
3	Ruslana Valentynivna Tarasyuk	24	Female	Lvovskaya obl	+380-661-555-5	V28902-JKKJSX0-0288	5
4	Illya Vitalijovych Chaban	34	Male	Dnepropetrov	+380-395-557-5	N389798-3489D-DLCG	1
5	Alina Leontiyivna Zakharchenko	27	Female	Donetskaya ob	+380-505-553-2	M299-CJJS0-2382NS-S	4
6	Mykyta Ruslanovych Slyusar	36	Male	Kharkovskaya	+380-505-554-1	X200-28SK-288D-KJCJ	2
7	Stefan Pavlovych Havrylenko	20	Male	Odesskaya obl	+380-505-551-2	KJK7-DJHJ-SJH3-JHSX4	2
8	Fedir Denysovych Lytovchenko	35	Male	Ternopolskaya	+380-505-551-5	S475-SB23K-DCJ-SKJD	3
9	Serhij Vasylovych Kulish	26	Male	Nikolaevskaya	+380-365-955-5	N239-382-CJD8S-SKN	3
10	Yana Eduardivna Nikolenko	28	Female	Kievskaya obl	+380-635-551-7	S827-SKDD-SDJ-SKDJ	3

Малюнок 5. Таблица Workers.

post_id	post_name	salary	responsibilities	requirements
1	Music Director	2000	Sifting through music submitted by independent musicians. Coordinating with the program director on the shows.	Responsibility, Communicability, Management skills.
2	Program Director	2000	Managing the station's programming teams. Making sure that radio station content appeals to its audience.	Responsibility, Communicability, Management skills.
3	Radio Presenter	1600	Developing content for the station's website & social media pages. Attending the station's events or hosting remote	Responsibility, Communicability, Media skills
4	Producer	2200	Protecting the radio station's license from invocation by ensuring that the programs don't feature obscenity, indecency or	Responsibility, Communicability, Management skills.
5	Broadcast Engineer	1800	Radio broadcast engineers handle all technical issues revolving around the broadcast equipment.	Responsibility, Technical qualification, Practical skills in solving issues.

Малюнок 6. Таблица Posts.

Схема данных	Workers	Timetable	Posts	Authors	Genres	Records
author_id	author_name	description				
+	1 Brothers Of Metal	One of Scandinavia's most promising new bands. The 8-piece, hailing from Falun, Sweden consists of 3 vocalists, 3 guitar players, a bass player and a drummer.				
+	2 The Neighborhood	An American alternative rock/dark pop band formed in Newbury Park, California, in 2011. The band is also known as "THE NBHD".				
+	3 PhemieC	A singer and a songwriter from Manitoba, Canada (previously Toronto). Their nickname pronounces as fem-ee-see. Has a beautiful voice and nice guitar skills.				
+	4 Vivienne Mort	A Ukrainian indie rock band formed by Daniela Zaiushkina in 2007.				
+	5 Fall Out Boy	An American rock band formed in Wilmette, Illinois, a suburb of Chicago, in 2001.				
+	6 Juniore	A French indie pop band formed in Paris in 2013. Their musical style is influenced by the music of the 1960s.				
+	7 Two Steps From Hell	An American production music company based in Los Angeles, California, founded by Thomas Bergersen and Nick Phoenix in 2006.				

Малюнок 7. Таблица Authors.

Схема данных	Workers	Timetable	Posts	Authors	Genres	Records
genre_id	genre_name	description				
+	2 Indie Rock	A subgenre of rock music that originated in the United States and United Kingdom in the 1970s. Originally used to describe independent record labels, the term became associated with the music they produced.				
+	3 Alternative Rock	A category of rock music that emerged from the independent music underground of the 1970s and became widely popular in the 1990s. "Alternative" refers to the genre's distinction from mainstream or commercial rock or pop music.				
+	4 Power Metal	Power metal is a subgenre of heavy metal combining characteristics of traditional heavy metal with speed metal, often within symphonic context. Generally, power metal is characterized by a faster, lighter, and more uplifting sound.				
+	5 Indie Folk	Indie folk is a music genre that arose in the 1990s among musicians from indie rock scenes influenced by folk music. Indie folk hybridizes the acoustic guitar melodies of traditional folk music with contemporary instrumentation.				
+	6 Orchestral	Orchestral music originated in the late 16th century stage spectacles and 17th century opera orchestras. The word "orchestra" came into common use in the 18th century, when music of the Baroque and Classical eras often was composed for specific occasions.				
+	7 Indie Pop	A music genre and subculture that combines guitar pop with DIY ethic in opposition to the style and tone of mainstream pop music.				

Малюнок 8. Таблица Genres.

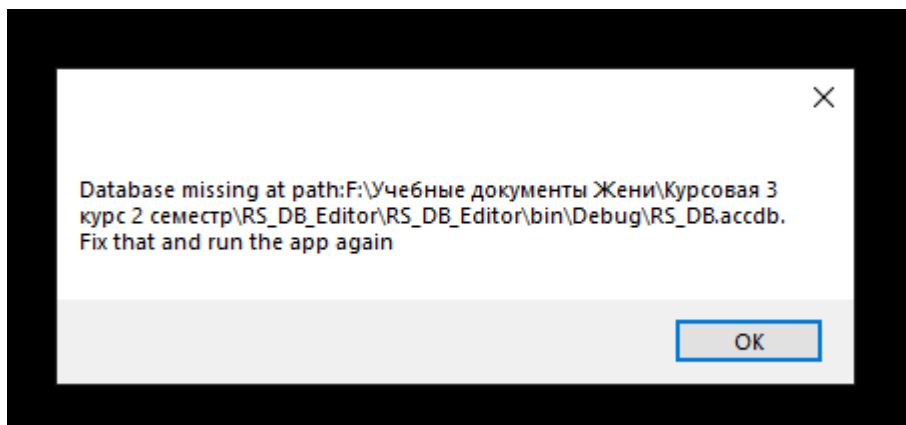
record_id	record_name	author_id	album	prod_year	genre_id	record_date	length	rating
1	Hel	1	Emblas Saga	2020	4	10.02.2020	0:04:41	8,7
2	Prophecy Of Ragnarok	1	Prophecy of Ragnarok	2017	4	01.04.2017	0:03:43	8,8
3	Weaver Of fate	1	Emblas Saga	2020	4	10.02.2020	0:04:46	8,4
4	How	2	I Love You	2013	3	23.04.2013	0:05:15	9,1
5	Alleyways	2	I Love You	2013	3	23.04.2013	0:04:28	8,6
6	Flawless	2	I Love You	2013	3	23.04.2013	0:04:07	8,8
7	Victory	7	Battlecry	2015	6	28.04.2015	0:05:20	8,5
8	Never Give Up	7	Unleashed	2017	6	22.09.2017	0:05:20	8,3
9	Star Sky	7	Battlecry	2015	6	28.04.2015	0:05:30	8,6
10	Ugly Story	3	Songs For Sad Trolls	2015	5	05.06.2015	0:03:20	9,1
11	An Apology	3	Songs For Sad Trolls	2015	5	05.06.2015	0:04:14	8,5
12	My Nightmare	3	Songs For A Doomed Time	2015	5	05.06.2015	0:03:33	8,7
13	Іній	4	Single	2017	2	01.02.2017	0:03:31	9,2
14	Хто Ти Такий	4	Rosa	2016	2	15.03.2016	0:03:27	8,9
15	Готика	4	Готика	2014	2	03.10.2014	0:03:43	8,7
16	Marabout	6	Marabout	2015	7	20.11.2015	0:03:47	7,8
17	Cavalier Solitaire	6	Marabout	2015	7	20.11.2015	0:03:43	8,1
18	A La Plage	6	Marabout	2015	7	20.11.2015	0:03:39	7,9
19	Centuries	5	American Beauty/American	2015	1	16.02.2015	0:03:52	8,8
20	Twin Skeleton	5	American Beauty/American	2015	1	16.02.2015	0:03:41	8,6
21	Fourth Of July	5	American Beauty/American	2015	1	16.02.2015	0:03:45	8,2

Малюнок 9. Таблица Records.

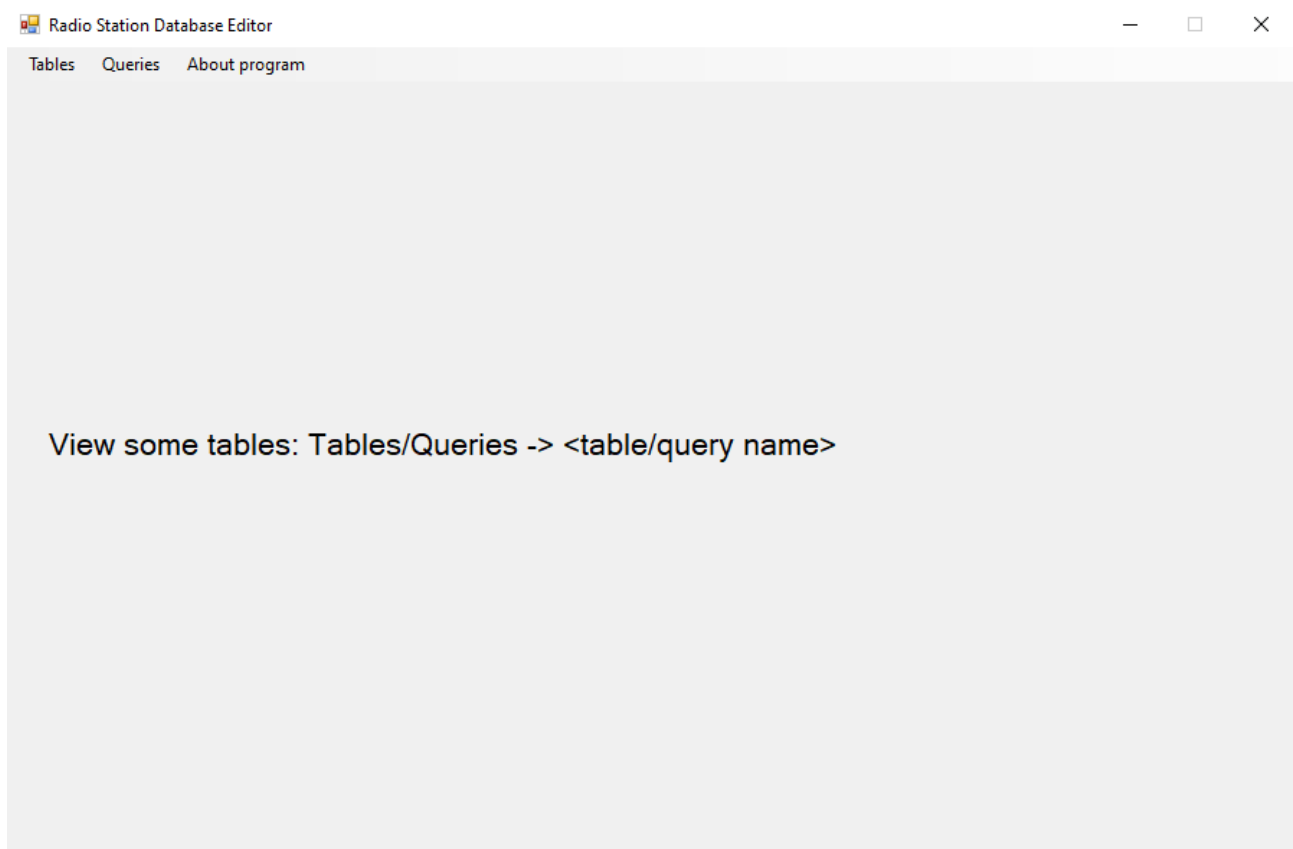
working_date	worker_id	time1	record1_id	time2	record2_id	time3	record3_id
16.05.2022	6	17:00:00	1	17:05:00	2	17:10:00	3
17.05.2022	4	14:30:00	4	14:35:00	5	14:40:00	6
18.05.2022	2	16:45:00	7	16:50:00	8	16:55:00	9
19.05.2022	3	17:00:00	10	17:05:00	11	17:10:00	12
20.05.2022	1	14:30:00	13	14:35:00	14	14:40:00	15
21.05.2022	7	16:45:00	16	16:50:00	17	16:55:00	18
22.05.2022	5	17:00:00	19	17:05:00	20	17:10:00	21
23.05.2022	9	14:30:00	1	14:35:00	2	14:40:00	3
24.05.2022	10	16:45:00	4	16:50:00	5	16:55:00	6
25.05.2022	8	17:00:00	7	17:05:00	8	17:10:00	9

Малюнок 10. Таблица Timetable.

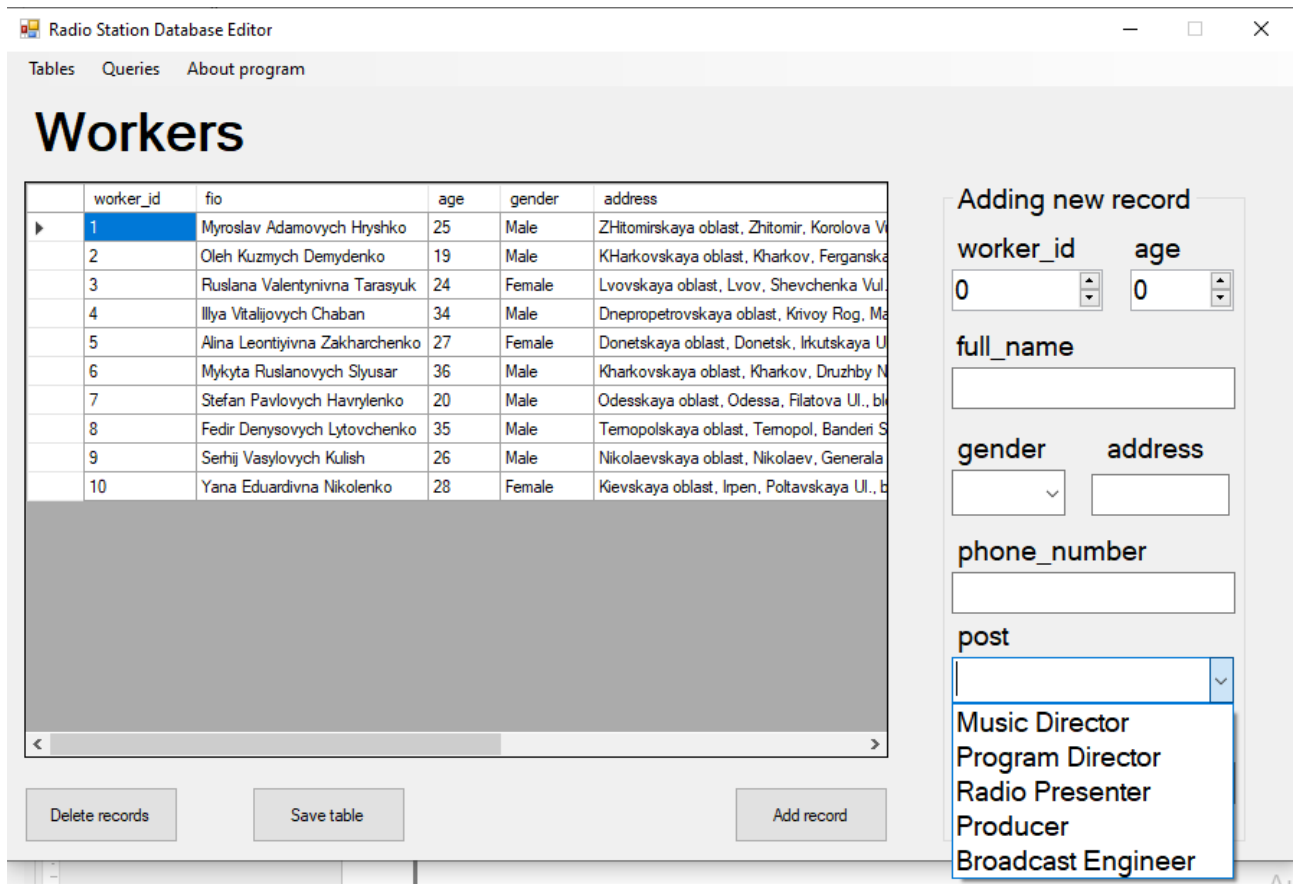
Скріншоти роботи програми



Малюнок 11. Повідомлення про відсутність БД.



Малюнок 12.Старт.



Малюнок 13.

Radio Station Database Editor

Tables

Queries

About program

Posts

	post_id	post_name	salary	responsibilities
▶	1	Music Director	2000	Sifting through music submitted by independent musicians. Coordinating
	2	Program Director	2000	Managing the station's programming teams. Making sure that radio sta
	3	Radio Presenter	1600	Developing content for the station's website & social media pages. Att
	4	Producer	2200	Protecting the radio station's license from invocation by ensuring that t
	5	Broadcast Engineer	1800	Radio broadcast engineers handle all technical issues revolving aroun

Adding new record

post_id

0

salary

0

post_name

responsibilities

requirements

Delete records

Save table

Add record

Малюнок 14.

Radio Station Database Editor

Tables

Queries

About program

Timetable

	working_date	worker_id	time1	record1_id	time2	record2_id	ti
▶	16.05.2022	6	17:00:00	1	17:05:00	2	1
	17.05.2022	4	14:30:00	4	14:35:00	5	1
	18.05.2022	2	16:45:00	7	16:50:00	8	1
	19.05.2022	3	17:00:00	10	17:05:00	11	1
	20.05.2022	1	14:30:00	13	14:35:00	14	1
	21.05.2022	7	16:45:00	16	16:50:00	17	1
	22.05.2022	5	17:00:00	19	17:05:00	20	1
	23.05.2022	9	14:30:00	1	14:35:00	2	1
	24.05.2022	10	16:45:00	4	16:50:00	5	1
	25.05.2022	8	17:00:00	7	17:05:00	8	1
	10.05.2022 7:12:03	3	10.05.2022 7:12:03	20	10.05.2022 7:12:03	14	10

Adding new record

working_date

10 травня 2022 р

worker

Ruslana Valentynivna Tarasyi

time1

7:12:03

time2

7:12:03

time3

7:12:03

record1

Twin Skeletons (Hote

record2

Хто Ти Такий

record3

An Apology

Delete records

Save table

Add record

Малюнок 15. Додав новий запис.

Radio Station Database Editor

Tables

Queries

About program

Authors

	author_id	author_name	description
▶	1	Brothers Of Metal	One of Scandinavia's most promising new bands. The 8-piece, hailing from Falu
	2	The Neighborhood	An American alternative rock/dark pop band formed in Newbury Park, Californi
	3	PhemieC	A singer and a songwriter from Manitoba, Canada (previously Toronto). Their nic
	4	Vivienne Mort	A Ukrainian indie rock band formed by Daniela Zaiushkina in 2007.
	5	Fall Out Boy	An American rock band formed in Wilmette, Illinois, a suburb of Chicago, in 200
	6	Juniore	A French indie pop band formed in Paris in 2013. Their musical style is influence
	7	Two Steps From Hell	An American production music company based in Los Angeles, California, foun

Adding new record

author_id

0

author_name

description

Delete records

Save table

Add record

Малюнок 16.

Radio Station Database Editor

Tables

Queries

About program

Genres

	genre_id	genre_name	description
▶	1	Rock	A broad genre of popular music that originated as "rock and roll" in the United State
	2	Indie Rock	A subgenre of rock music that originated in the United States and United Kingdom i
	3	Alternative Rock	A category of rock music that emerged from the independent music underground of
	4	Power Metal	Power metal is a subgenre of heavy metal combining characteristics of traditional he
	5	Indie Folk	Indie folk is a music genre that arose in the 1990s among musicians from indie rock
	6	Orchestral	Orchestral music originated in the late 16th century stage spectacles and 17th cent
	7	Indie Pop	A music genre and subculture that combines guitar pop with DIY ethic in opposition

Adding new record

genre_id

0

genre_name

description

Delete records

Save table

Add record

Малюнок 17.

Radio Station Database Editor

Tables

Queries

About program

Records

	record_id	record_name	author_id	album	prod_year
	4	How	2	I Love You	2013
	5	Alleyways	2	I Love You	2013
	6	Flawless	2	I Love You	2013
	7	Victory	7	Battlecry	2015
	8	Never Give Up On Your Dreams	7	Unleashed	2017
	9	Star Sky	7	Battlecry	2015
	10	Ugly Story	3	Songs For Sad Trolls	2015
	11	An Apology	3	Songs For Sad Trolls	2015
	12	My Nightmare	3	Songs For A Doomed Timeline	2015
	13	Іній	4	Single	2017
	14	Хто Ти Такий	4	Rósa	2016
	15	Готика	4	Готика	2014
	16	Marabout	6	Marabout	2015
	17	Cavalier Solitaire	6	Marabout	2015
	18	A La Plage	6	Marabout	2015
	19	Centuries	5	American Beauty/American Psycho	2015
	20	Twin Skeletons (Hotel in NYC)	5	American Beauty/American Psycho	2015
	21	Fourth Of July	5	American Beauty/American psycho	2015

Adding new record

record_id

rating

0

0,0

record_name

album

genre

length

author

record_date

10 травня 2022 р

Delete records

Save table

Add record

Малюнок 18.

Radio Station Database Editor

Tables

Queries

About program

Personnel department

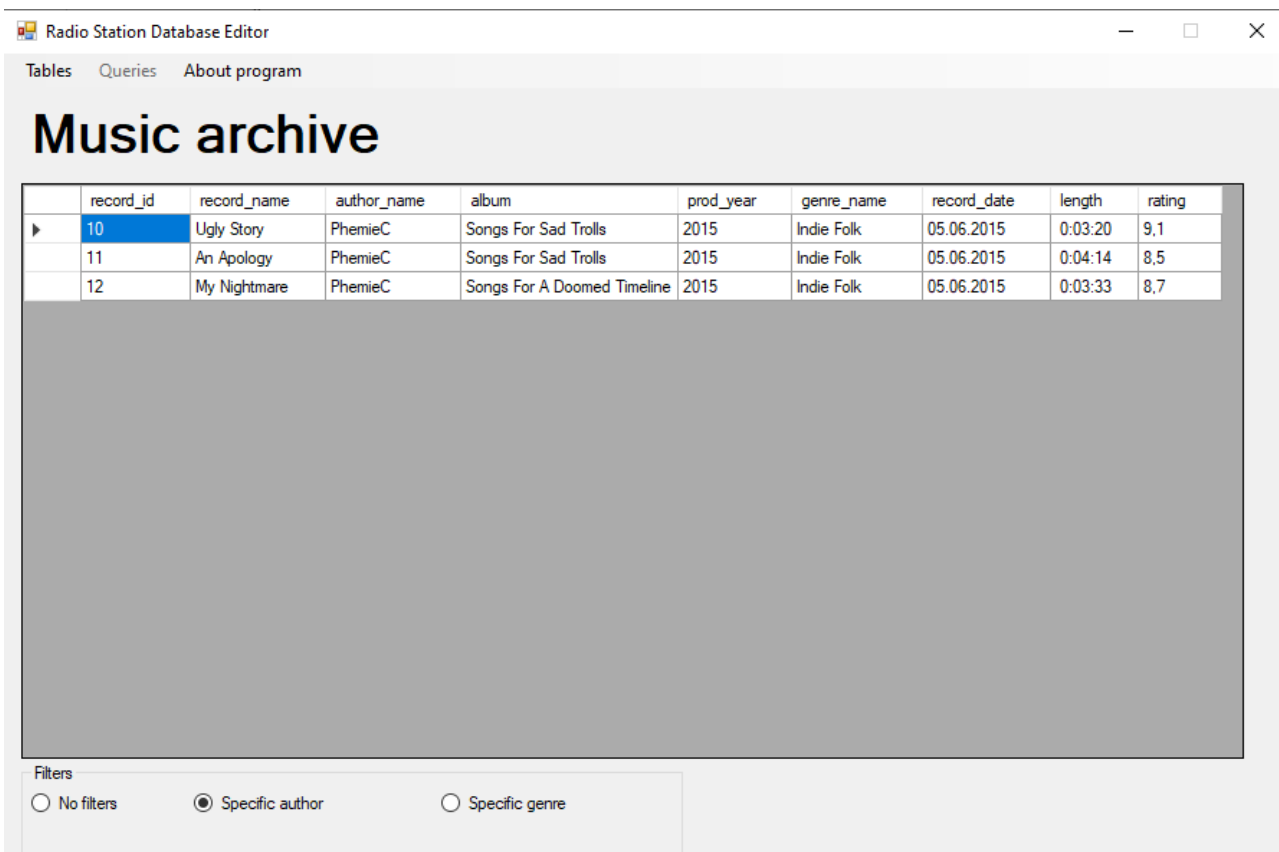
	age	gender	address	phone_number	passport_details	post_name
	25	Male	Zhitomirskaya oblast, Zhitomir, Korolova Vul., bld. 46/Б, appt. 84	+380-273-855-581	S34240-SF5DF-234-DF	Broadcast Engineer
	19	Male	Kharkovskaya oblast, Kharkov, Ferganskaya Ul., bld. 33/Б, appt. 28	+380-395-558-234	00SDJNK-DKKKS-DDG	Broadcast Engineer
	24	Female	Lvovskaya oblast, Lvov, Shevchenka Vul., bld. 370, appt. 34	+380-661-555-524	V28902-JKKJSX0-0288	Broadcast Engineer
	34	Male	Dnepropetrovskaya oblast, Krivoy Rog, Marshaka Ul., bld. 16, appt. 30	+380-395-557-592	N389798-3489D-DLCG	Music Director
	27	Female	Donetskaya oblast, Donetsk, Irkutskaya Ul., bld. 37	+380-505-553-232	M299-CJJS0-2382NS-S	Producer
	36	Male	Kharkovskaya oblast, Kharkov, Druzhby Narodov Ul., bld. 238, appt. 209	+380-505-554-152	X200-28SK-288D-KJCJ	Program Director
	20	Male	Odesskaya oblast, Odessa, Filatova Ul., bld. 10/A, appt. 36	+380-505-551-242	KJK7-DJHJ-SJH3-JHSX4	Program Director
	35	Male	Temopolskaya oblast, Temopol, Banderi S. Pr., bld. 84, appt. 61	+380-505-551-590	S475-SB23K-DCJ-SKJD	Radio Presenter
	26	Male	Nikolaevskaya oblast, Nikolaev, Generala Karpenko, bld. 59/A, appt. 63	+380-365-955-535	N239-382-CJD8S-SKN	Radio Presenter
	28	Female	Kievskaya oblast, Ipen, Poltavskaya Ul., bld. 63, appt. 64	+380-635-551-743	S827-SKDD-SDJ-SKDJ	Radio Presenter

Filters

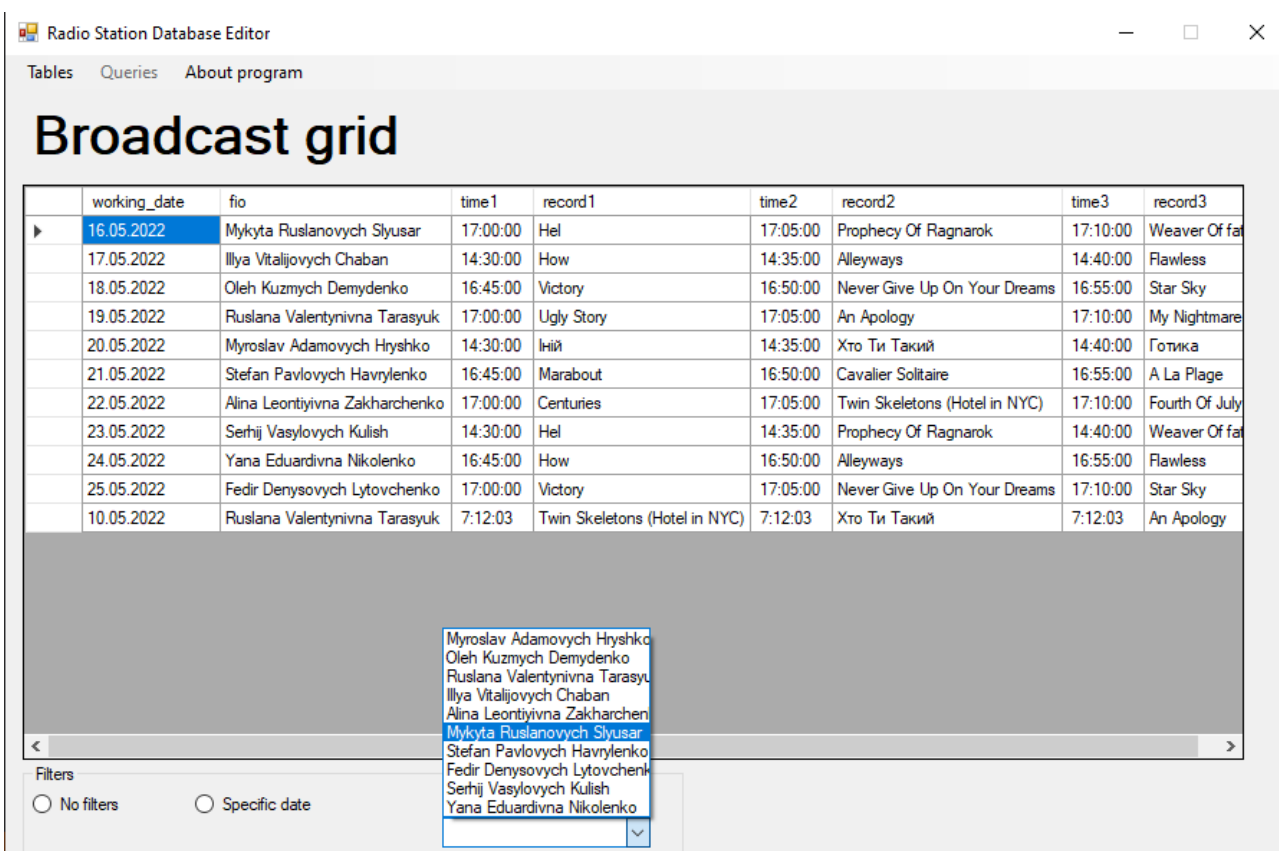
No filters

Specific post

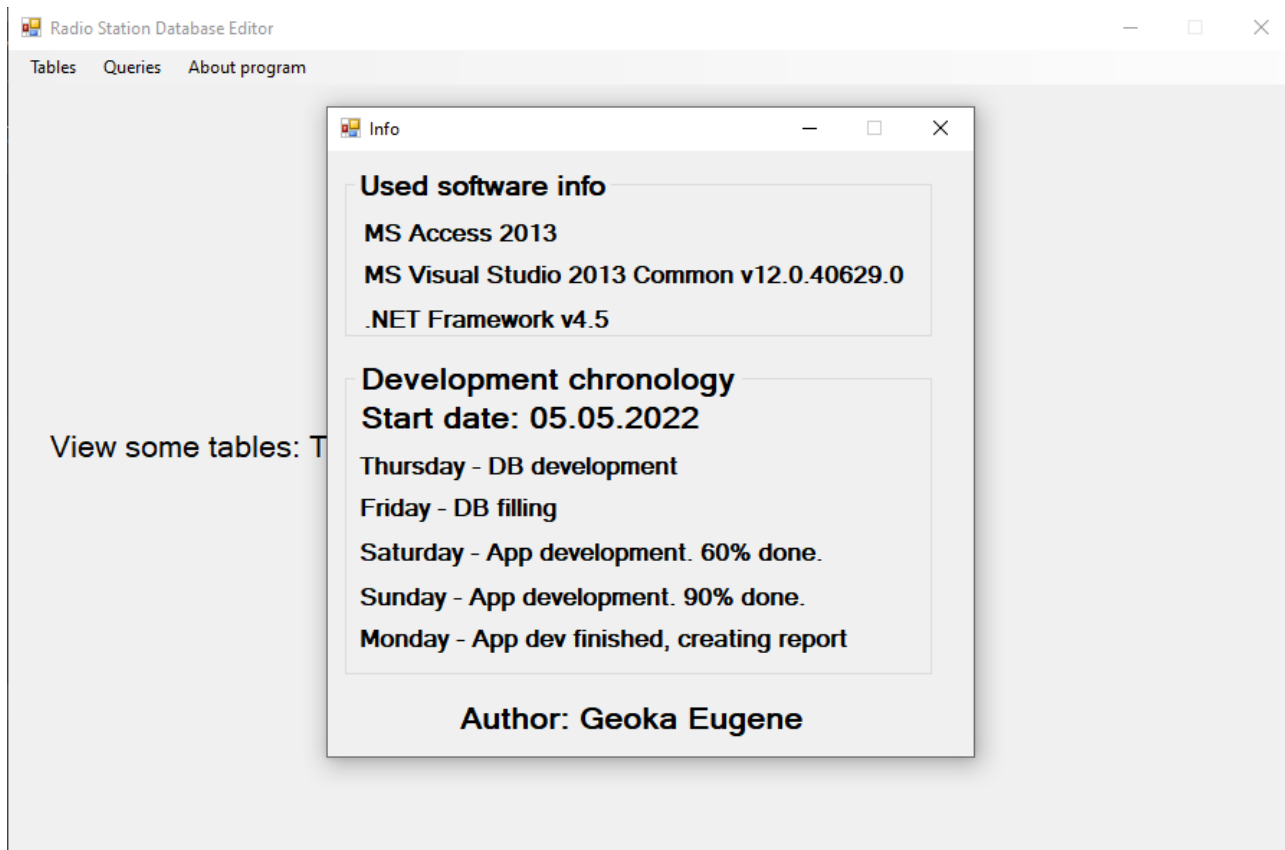
Малюнок 19. Запит Відділ кадрів



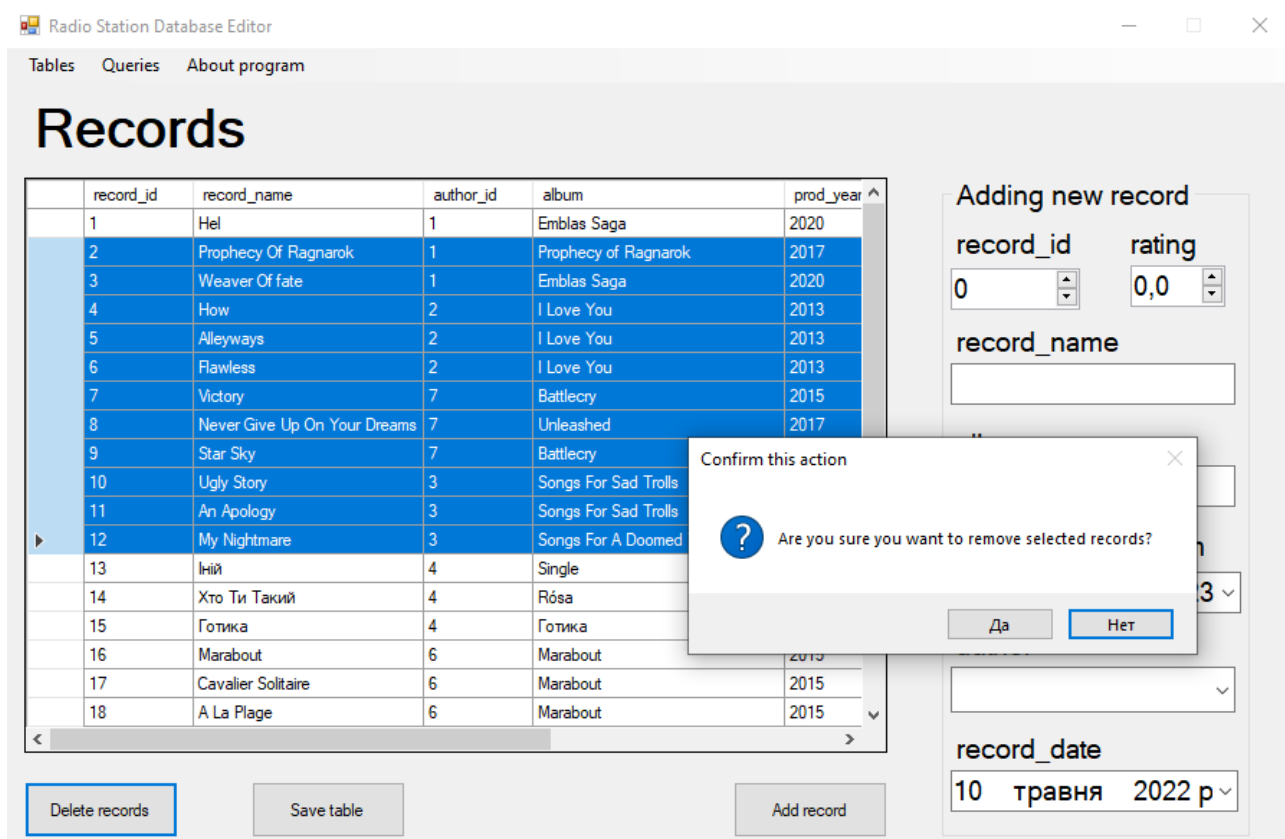
Малюнок 22. Запит Музичний архів + фільтр по виконавцю (PhemieC)



Малюнок 23. Запит Сітка мовлення + вибір ПІБ для фільтру по працівнику



Малюнок 24. Інформація про додаток



Малюнок 25. Повідомлення про підтвердження видалення записів

Radio Station Database Editor

Tables
Queries
About program

Records

	record_id	record_name	author_id	album	prod_year
	1	Hel	1	Emblas Saga	2020
▶	13	Іній	4	Single	2017
	14	Хто Ти Такий	4	Rósa	2016
	15	Готика	4	Готика	2014
	16	Marabout	6	Marabout	2015
	17	Cavalier Solitaire	6	Marabout	2015
	18	A La Plage	6	Marabout	2015
	19	Centuries	5	American Beauty/American Psycho	2015
	20	Twin Skeletons (Hotel in NYC)	5	American Beauty/American Psycho	2015
	21	Fourth Of July	5	American Beauty/American psycho	2015

Adding new record

record_id

rating

0

0,0

record_name

album

genre

length

author

record_date

10 травня 2022 р

Delete records

Save table

Add record

Малюнок 26. Записи видалені

Висновки

Під час виконання завдання цієї курсової роботи мені довелося шукати додаткову інформацію, вивчати нові практичні прийоми, тому обидві мети були досягнуті:

- Я покращив свої навички і знання;
- Вимоги до завдань були виконані у повному обсязі і, навіть, були реалізовані додаткові функції та перевірки.

Але все ж таки моє виконання не є ідеальним. Можна додати функцію редагування даних, можна трохи змінити структуру таблиці Timetable, додавши ключове поле лічильника, що дозволить зняти ключове поле із `working_date`, що зробить можливим мати різні записи на одну й ту ж дату. Також в процесі створення цього документа я знайшов декілька нюансів, які можна виправити, щоб вдосконалити додаток:

- Якщо назва пісні містить символ ‘ то така назва не проходить перевірку на коректність даних.
- Якщо спочатку вибрати фільтр, відкрити ComboBox і нічого не вибрати, після чого зняти фокус з ComboBox, виникне критична помилка в процесі виконання програми. (Індекс обраного елемента не існує, а в подальшому він використовується).

Не зміг я залишити критичну помилку, про яку знаю. Виправив. Що стосується символів ‘ в назвах, то я просто їх видалив, і все працює.

Перелік використаних інформаційних джерел

- 1) Герберт Шилдт – «С# 4.0. Повне керівництво».
- 2) <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/>.
- 3) С. Макконнел – «Досконалий код». (рос. «Совершенный код»)
- 4) В. Пасічник «Організація баз даних та знань».

					12.121.B31.05.KP.ПЗ	Лист
Изм.	Лист	№ Документа	Подпись	Дат		28

Додаток 1. Код програми

Info.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace RS_DB_Editor
{
    public partial class Info : Form
    {
        public Info()
        {
            InitializeComponent();
        }
    }
}
```

Main.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
using System.IO;
```

```
namespace RS_DB_Editor
{
    public partial class Main : Form
    {
        static string path = @"F:\Учебные документы Жени\Курсовая 3 курс 2
семестр\RS_DB_Editor\RS_DB_Editor\bin\Debug\RS_DB.accdb";
        static OleDbConnection RS_DB_Connection = new
OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + path);
```

```
        DataSet WORKERS_DS = new DataSet();
        DataSet POSTS_DS = new DataSet();
        DataSet TIMETABLE_DS = new DataSet();
        DataSet AUTHORS_DS = new DataSet();
        DataSet GENRES_DS = new DataSet();
        DataSet RECORDS_DS = new DataSet();
```

```
        DataSet PERSONNEL_DEPARTMENT_DS = new DataSet();
        DataSet MUSIC_ARCHIVE_DS = new DataSet();
```

					12.121.B31.05.KP.ПЗ	Лист
Изм.	Лист	№ Документа	Подпись	Дат		29

```

DataSet BROADCAST_GRID_DS = new DataSet();

OleDbDataAdapter WORKERS_DA = new OleDbDataAdapter("SELECT * FROM Workers",
RS_DB_Connection);
OleDbDataAdapter POSTS_DA = new OleDbDataAdapter("SELECT * FROM Posts", RS_DB_Connection);
OleDbDataAdapter TIMETABLE_DA = new OleDbDataAdapter("SELECT * FROM Timetable",
RS_DB_Connection);
OleDbDataAdapter AUTHORS_DA = new OleDbDataAdapter("SELECT * FROM Authors",
RS_DB_Connection);
OleDbDataAdapter GENRES_DA = new OleDbDataAdapter("SELECT * FROM Genres", RS_DB_Connection);
OleDbDataAdapter RECORDS_DA = new OleDbDataAdapter("SELECT * FROM Records",
RS_DB_Connection);

OleDbDataAdapter PERSONNEL_DEPARTMENT_DA = new OleDbDataAdapter("SELECT
Workers.worker_id, Workers.fio, Workers.age, Workers.gender, Workers.address, Workers.phone_number,
Workers.passport_details, Posts.post_name FROM Workers INNER JOIN Posts ON Workers.post_id = Posts.post_id",
RS_DB_Connection);
OleDbDataAdapter MUSIC_ARCHIVE_DA = new OleDbDataAdapter("SELECT Records.record_id,
Records.record_name, Authors.author_name, Records.album, Records.prod_year, Genres.genre_name,
Records.record_date, Records.length, Records.rating FROM Genres INNER JOIN (Authors INNER JOIN Records ON
Authors.author_id = Records.author_id) ON Genres.genre_id = Records.genre_id", RS_DB_Connection);
OleDbDataAdapter BROADCAST_GRID_DA = new OleDbDataAdapter("SELECT Timetable.working_date,
Workers.fio, Timetable.time1, Records.record_name AS record1, Timetable.time2, Records_1.record_name AS
record2, Timetable.time3, Records_2.record_name AS record3 FROM Records AS Records_2 INNER JOIN (Records
AS Records_1 INNER JOIN (Records INNER JOIN (Workers INNER JOIN Timetable ON Workers.worker_id =
Timetable.worker_id) ON Records.record_id = Timetable.record1_id) ON Records_1.record_id =
Timetable.record2_id) ON Records_2.record_id = Timetable.record3_id", RS_DB_Connection);

OleDbCommandBuilder WORKERS_CB = new OleDbCommandBuilder();
OleDbCommandBuilder POSTS_CB = new OleDbCommandBuilder();
OleDbCommandBuilder TIMETABLE_CB = new OleDbCommandBuilder();
OleDbCommandBuilder AUTHORS_CB = new OleDbCommandBuilder();
OleDbCommandBuilder GENRES_CB = new OleDbCommandBuilder();
OleDbCommandBuilder RECORDS_CB = new OleDbCommandBuilder();

OleDbCommandBuilder PERSONNEL_DEPARTMENT_CB = new OleDbCommandBuilder();
OleDbCommandBuilder MUSIC_ARCHIVE_CB = new OleDbCommandBuilder();
OleDbCommandBuilder BROADCAST_GRID_CB = new OleDbCommandBuilder();

public Main()
{
    InitializeComponent();

    // Reset method for application
    public static void Reset(DataGridView DataGridView, Label Table_Name_Label, GroupBox
WORKERS_GroupBox, GroupBox POSTS_GroupBox, GroupBox TIMETABLE_GroupBox, GroupBox
AUTHORS_GroupBox, GroupBox GENRES_GroupBox, GroupBox RECORDS_GroupBox, GroupBox
FILTERS_GroupBox, Button Add_Button, Button Delete_Button, Button Save_Button, RadioButton F0_RB,
RadioButton F1_RB, RadioButton F2_RB, ComboBox F1_CB, ComboBox F2_CB)
    {
        DataGridView.Visible = false;
        Table_Name_Label.Visible = false;

        DataGridView.Rows.Clear();
        DataGridView.Columns.Clear();
        DataGridView.Width = 600;

        WORKERS_GroupBox.Visible = false;
        POSTS_GroupBox.Visible = false;
        TIMETABLE_GroupBox.Visible = false;
        AUTHORS_GroupBox.Visible = false;
        GENRES_GroupBox.Visible = false;
    }

```

```

RECORDS_GroupBox.Visible = false;
FILTERS_GroupBox.Visible = false;
FILTERS_GroupBox.Width = 460;

F0_RB.Text = "No filters";
F1_RB.Text = "f1";
F2_RB.Text = "f2";
F1_CB.Items.Clear();
F2_CB.Items.Clear();
F1_CB.Visible = false;
F2_CB.Visible = false;

Add_Button.Visible = false;
Delete_Button.Visible = false;
Save_Button.Visible = false;
}

//Fill method for DataGridView (1 DataGridView for any DataSet.Tables[0])
public static void DataGrid_Fill(DataGridView DT, DataSet DS)
{
    for (int i = 0; i < DS.Tables[0].Columns.Count; i++)
    {
        DT.Columns.Add(DS.Tables[0].Columns[i].Caption, DS.Tables[0].Columns[i].Caption);
    }

    for (int m = 0; m < DS.Tables[0].Rows.Count; m++)
    {
        DT.Rows.Add();
        for (int n = 0; n < DS.Tables[0].Columns.Count; n++)
        {
            DT.Rows[m].Cells[n].Value = DS.Tables[0].Rows[m].ItemArray[n];
        }
    }
}

//DateTime truncate method (01.01.2022 17:00:00 -> 17:00:00)
public static void DT_Cell_Date_Truncate(DataGridView DT, int id)
{
    string new_value;
    for (int i = 0; i < DT.Rows.Count; i++)
    {
        new_value = DT.Rows[i].Cells[id].Value.ToString().Remove(0, 10);
        DT.Rows[i].Cells[id].Value = new_value;
    }
}

//DateTime truncate method (01.01.2022 17:00:00 -> 01.01.2022)
public static void DT_Cell_Time_Truncate(DataGridView DT, int id)
{
    string new_value;
    for (int i = 0; i < DT.Rows.Count; i++)
    {
        new_value = DT.Rows[i].Cells[id].Value.ToString().Remove(10);
        DT.Rows[i].Cells[id].Value = new_value;
    }
}

//Fill method for ComboBox (1 ComboBox to display any Column of any DataSet.Tables[0])
public static void ComboBox_Fill(ComboBox CB, DataSet DS, int id)
{
    CB.Items.Clear();
    for (int i = 0; i < DS.Tables[0].Rows.Count; i++)
    {

```

```

        CB.Items.Add(DS.Tables[0].Rows[i].ItemArray[id]);
    }
}

//A quite tricky method to get the id of selected item in Combobox
public static string Get_id(string input, string id_column, string name_column, string table)
{
    RS_DB_Connection.Open();
    OleDbCommand select_id = RS_DB_Connection.CreateCommand();

    select_id.Parameters.Add("@ID_COLUMN", OleDbType.VarChar);
    select_id.Parameters.Add("@TABLE", OleDbType.VarChar);
    select_id.Parameters.Add("@NAME_COLUMN", OleDbType.VarChar);
    select_id.Parameters.Add("@INPUT", OleDbType.VarChar);

    select_id.Parameters["@ID_COLUMN"].Value = id_column;
    select_id.Parameters["@TABLE"].Value = table;
    select_id.Parameters["@NAME_COLUMN"].Value = name_column;
    select_id.Parameters["@INPUT"].Value = input;

    select_id.CommandText = "SELECT " + id_column + " FROM " + table + " WHERE " + name_column + " = " +
    input + """";

    string result = select_id.ExecuteScalar().ToString();
    RS_DB_Connection.Close();
    return result;
}

//A method that removes all selected records from DataGridView and current DataSet
public static void Table_Remove(DataGridView DT, DataSet DS, OleDbDataAdapter DA)
{
    if (MessageBox.Show("Are you sure you want to remove selected records?", "Confirm this action",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question, MessageBoxDefaultButton.Button2) == DialogResult.Yes)
    {
        int temp;
        for (int i = DT.SelectedRows.Count - 1; i >= 0; i--)
        {
            temp = DT.SelectedRows[i].Index;
            DT.Rows.RemoveAt(temp);
            DS.Tables[0].Rows[temp].Delete();
            DA.Update(DS);
        }
    }
}

private void Main_Load(object sender, EventArgs e)
{
    Width = 910;
    CenterToScreen();
    Reset(DataGridView, Table_Name_Label, WORKERS_GroupBox, POSTS_GroupBox,
    TIMETABLE_GroupBox, AUTHORS_GroupBox, GENRES_GroupBox, RECORDS_GroupBox,
    FILTERS_GroupBox, Add_Button, Delete_Button, Save_Button, NO_FILTERS_RB, FILTER_1_RB, FILTER_2_RB,
    FILTER_1_CB, FILTER_2_CB);

    //Checking whether the database file exists
    if (File.Exists(path) == false)
    {
        MessageBox.Show("Database missing at path:" + path + ". Fix that and run the app again");
        Environment.Exit(0);
    }

    WORKERS_CB.DataAdapter = WORKERS_DA;
    WORKERS_DA.InsertCommand = WORKERS_CB.GetInsertCommand();
}

```



```

WORKERS_DA.DeleteCommand = WORKERS_CB.GetDeleteCommand();
WORKERS_DA.UpdateCommand = WORKERS_CB.GetUpdateCommand();
WORKERS_DA.Fill(WORKERS_DS);

POSTS_CB.DataAdapter = POSTS_DA;
POSTS_DA.InsertCommand = POSTS_CB.GetInsertCommand();
POSTS_DA.DeleteCommand = POSTS_CB.GetDeleteCommand();
POSTS_DA.UpdateCommand = POSTS_CB.GetUpdateCommand();
POSTS_DA.Fill(POSTS_DS);

TIMETABLE_CB.DataAdapter = TIMETABLE_DA;
TIMETABLE_DA.InsertCommand = TIMETABLE_CB.GetInsertCommand();
TIMETABLE_DA.DeleteCommand = TIMETABLE_CB.GetDeleteCommand();
TIMETABLE_DA.UpdateCommand = TIMETABLE_CB.GetUpdateCommand();
TIMETABLE_DA.Fill(TIMETABLE_DS);

AUTHORS_CB.DataAdapter = AUTHORS_DA;
AUTHORS_DA.InsertCommand = AUTHORS_CB.GetInsertCommand();
AUTHORS_DA.DeleteCommand = AUTHORS_CB.GetDeleteCommand();
AUTHORS_DA.UpdateCommand = AUTHORS_CB.GetUpdateCommand();
AUTHORS_DA.Fill(AUTHORS_DS);

GENRES_CB.DataAdapter = GENRES_DA;
GENRES_DA.InsertCommand = GENRES_CB.GetInsertCommand();
GENRES_DA.DeleteCommand = GENRES_CB.GetDeleteCommand();
GENRES_DA.UpdateCommand = GENRES_CB.GetUpdateCommand();
GENRES_DA.Fill(GENRES_DS);

RECORDS_CB.DataAdapter = RECORDS_DA;
RECORDS_DA.InsertCommand = RECORDS_CB.GetInsertCommand();
RECORDS_DA.DeleteCommand = RECORDS_CB.GetDeleteCommand();
RECORDS_DA.UpdateCommand = RECORDS_CB.GetUpdateCommand();
RECORDS_DA.Fill(RECORDS_DS);
}

```

```

private void workersToolStripMenuItem_Click(object sender, EventArgs e)
{
    NO_FILTERS_RB.Checked = true;
    QueriesToolStripMenuItem.Enabled = true;
    //workers table
    WORKERS_DS.Clear();
    DataGridView.Rows.Clear();
    DataGridView.Columns.Clear();
    Reset(DataGridView, Table_Name_Label, WORKERS_GroupBox, POSTS_GroupBox,
TIMETABLE_GroupBox, AUTHORS_GroupBox, GENRES_GroupBox, RECORDS_GroupBox,
FILTERS_GroupBox, Add_Button, Delete_Button, Save_Button, NO_FILTERS_RB, FILTER_1_RB, FILTER_2_RB,
FILTER_1_CB, FILTER_2_CB);
    WORKERS_CB.DataAdapter = WORKERS_DA;
    WORKERS_DA.Fill(WORKERS_DS);
    DataGrid_Fill(DataGridView, WORKERS_DS);

    W_gender_CB.Items.Clear();
    W_gender_CB.Items.Add("Male");
    W_gender_CB.Items.Add("Female");
    ComboBox_Fill(W_post_CB, POSTS_DS, 1);

    DataGridView.Visible = true;
    Table_Name_Label.Text = "Workers";
    Table_Name_Label.Visible = true;
    WORKERS_GroupBox.Visible = true;
    Add_Button.Visible = true;
    Delete_Button.Visible = true;
}

```

					12.121.B31.05.KP.ПЗ	Лист
Изм.	Лист	№ Документа	Подпись	Дат		33

```

        Save_Button.Visible = true;
    }

private void postsToolStripMenuItem_Click(object sender, EventArgs e)
{
    NO_FILTERS_RB.Checked = true;
    QueriesToolStripMenuItem.Enabled = true;
    //posts table
    POSTS_DS.Clear();
    Reset(DataGridView, Table_Name_Label, WORKERS_GroupBox, POSTS_GroupBox,
TIMETABLE_GroupBox, AUTHORS_GroupBox, GENRES_GroupBox, RECORDS_GroupBox,
FILTERS_GroupBox, Add_Button, Delete_Button, Save_Button, NO_FILTERS_RB, FILTER_1_RB, FILTER_2_RB,
FILTER_1_CB, FILTER_2_CB);
    POSTS_CB.DataAdapter = POSTS_DA;
    POSTS_DA.Fill(POSTS_DS);
    DataGrid_Fill(DataGridView, POSTS_DS);

    DataGridView.Visible = true;
    Table_Name_Label.Text = "Posts";
    Table_Name_Label.Visible = true;
    POSTS_GroupBox.Visible = true;
    Add_Button.Visible = true;
    Delete_Button.Visible = true;
    Save_Button.Visible = true;
}

private void timetableToolStripMenuItem_Click(object sender, EventArgs e)
{
    NO_FILTERS_RB.Checked = true;
    QueriesToolStripMenuItem.Enabled = true;
    //timetable table
    TIMETABLE_DS.Clear();
    Reset(DataGridView, Table_Name_Label, WORKERS_GroupBox, POSTS_GroupBox,
TIMETABLE_GroupBox, AUTHORS_GroupBox, GENRES_GroupBox, RECORDS_GroupBox,
FILTERS_GroupBox, Add_Button, Delete_Button, Save_Button, NO_FILTERS_RB, FILTER_1_RB, FILTER_2_RB,
FILTER_1_CB, FILTER_2_CB);
    TIMETABLE_CB.DataAdapter = TIMETABLE_DA;
    TIMETABLE_DA.Fill(TIMETABLE_DS);
    DataGrid_Fill(DataGridView, TIMETABLE_DS);

    DT_Cell_Time_Truncate(DataGridView, 0);
    DT_Cell_Date_Truncate(DataGridView, 2);
    DT_Cell_Date_Truncate(DataGridView, 4);
    DT_Cell_Date_Truncate(DataGridView, 6);

    ComboBox_Fill(T_worker_CB, WORKERS_DS, 1);
    ComboBox_Fill(T_record1_CB, RECORDS_DS, 1);
    ComboBox_Fill(T_record2_CB, RECORDS_DS, 1);
    ComboBox_Fill(T_record3_CB, RECORDS_DS, 1);

    DataGridView.Visible = true;
    Table_Name_Label.Text = "Timetable";
    Table_Name_Label.Visible = true;
    TIMETABLE_GroupBox.Visible = true;
    Add_Button.Visible = true;
    Delete_Button.Visible = true;
    Save_Button.Visible = true;
}

private void authorsToolStripMenuItem_Click(object sender, EventArgs e)
{
    NO_FILTERS_RB.Checked = true;
    QueriesToolStripMenuItem.Enabled = true;

```

```

//authors table
AUTHORS_DS.Clear();
Reset(DataGridView, Table_Name_Label, WORKERS_GroupBox, POSTS_GroupBox,
TIMETABLE_GroupBox, AUTHORS_GroupBox, GENRES_GroupBox, RECORDS_GroupBox,
FILTERS_GroupBox, Add_Button, Delete_Button, Save_Button, NO_FILTERS_RB, FILTER_1_RB, FILTER_2_RB,
FILTER_1_CB, FILTER_2_CB);
AUTHORS_CB.DataAdapter = AUTHORS_DA;
AUTHORS_DA.Fill(AUTHORS_DS);
DataGrid_Fill(DataGridView, AUTHORS_DS);

DataGridView.Visible = true;
Table_Name_Label.Text = "Authors";
Table_Name_Label.Visible = true;
AUTHORS_GroupBox.Visible = true;
Add_Button.Visible = true;
Delete_Button.Visible = true;
Save_Button.Visible = true;
}

private void genresToolStripMenuItem_Click(object sender, EventArgs e)
{
    NO_FILTERS_RB.Checked = true;
    QueriesToolStripMenuItem.Enabled = true;
    //genres table
    GENRES_DS.Clear();
    Reset(DataGridView, Table_Name_Label, WORKERS_GroupBox, POSTS_GroupBox,
TIMETABLE_GroupBox, AUTHORS_GroupBox, GENRES_GroupBox, RECORDS_GroupBox,
FILTERS_GroupBox, Add_Button, Delete_Button, Save_Button, NO_FILTERS_RB, FILTER_1_RB, FILTER_2_RB,
FILTER_1_CB, FILTER_2_CB);
    GENRES_CB.DataAdapter = GENRES_DA;
    GENRES_DA.Fill(GENRES_DS);
    DataGrid_Fill(DataGridView, GENRES_DS);

    DataGridView.Visible = true;
    Table_Name_Label.Text = "Genres";
    Table_Name_Label.Visible = true;
    GENRES_GroupBox.Visible = true;
    Add_Button.Visible = true;
    Delete_Button.Visible = true;
    Save_Button.Visible = true;
}

private void recordsToolStripMenuItem_Click(object sender, EventArgs e)
{
    NO_FILTERS_RB.Checked = true;
    QueriesToolStripMenuItem.Enabled = true;
    //records table
    RECORDS_DS.Clear();
    Reset(DataGridView, Table_Name_Label, WORKERS_GroupBox, POSTS_GroupBox,
TIMETABLE_GroupBox, AUTHORS_GroupBox, GENRES_GroupBox, RECORDS_GroupBox,
FILTERS_GroupBox, Add_Button, Delete_Button, Save_Button, NO_FILTERS_RB, FILTER_1_RB, FILTER_2_RB,
FILTER_1_CB, FILTER_2_CB);
    RECORDS_CB.DataAdapter = RECORDS_DA;
    RECORDS_DA.Fill(RECORDS_DS);
    DataGrid_Fill(DataGridView, RECORDS_DS);

    DT_Cell_Time_Truncate(DataGridView, 6);
    DT_Cell_Date_Truncate(DataGridView, 7);

    ComboBox_Fill(R_genre_CB, GENRES_DS, 1);
    ComboBox_Fill(R_author_CB, AUTHORS_DS, 1);

    DataGridView.Visible = true;

```

```

        Table_Name_Label.Text = "Records";
        Table_Name_Label.Visible = true;
        RECORDS_GroupBox.Visible = true;
        Add_Button.Visible = true;
        Delete_Button.Visible = true;
        Save_Button.Visible = true;
    }

    private void personnelDepartmentToolStripMenuItem_Click(object sender, EventArgs e)
    {
        // query 1
        PERSONNEL_DEPARTMENT_DS.Clear();
        Reset(DataGridView, Table_Name_Label, WORKERS_GroupBox, POSTS_GroupBox,
TIMETABLE_GroupBox, AUTHORS_GroupBox, GENRES_GroupBox, RECORDS_GroupBox,
FILTERS_GroupBox, Add_Button, Delete_Button, Save_Button, NO_FILTERS_RB, FILTER_1_RB, FILTER_2_RB,
FILTER_1_CB, FILTER_2_CB);
        PERSONNEL_DEPARTMENT_CB.DataAdapter = PERSONNEL_DEPARTMENT_DA;
        PERSONNEL_DEPARTMENT_DA.Fill(PERSONNEL_DEPARTMENT_DS);
        DataGridView_Fill(DataGridView, PERSONNEL_DEPARTMENT_DS);
        DataGridView.Width = 850;

        DataGridView.Visible = true;
        Table_Name_Label.Text = "Personnel department";
        Table_Name_Label.Visible = true;

        FILTER_1_RB.Text = "Specific post";
        FILTER_2_RB.Visible = false;
        FILTERS_GroupBox.Width = 275;
        FILTERS_GroupBox.Visible = true;
    }

    private void musicArchiveToolStripMenuItem_Click(object sender, EventArgs e)
    {
        // query 2
        MUSIC_ARCHIVE_DS.Clear();
        Reset(DataGridView, Table_Name_Label, WORKERS_GroupBox, POSTS_GroupBox,
TIMETABLE_GroupBox, AUTHORS_GroupBox, GENRES_GroupBox, RECORDS_GroupBox,
FILTERS_GroupBox, Add_Button, Delete_Button, Save_Button, NO_FILTERS_RB, FILTER_1_RB, FILTER_2_RB,
FILTER_1_CB, FILTER_2_CB);
        MUSIC_ARCHIVE_CB.DataAdapter = MUSIC_ARCHIVE_DA;
        MUSIC_ARCHIVE_DA.Fill(MUSIC_ARCHIVE_DS);
        DataGridView_Fill(DataGridView, MUSIC_ARCHIVE_DS);
        DataGridView.Width = 850;

        DT_Cell_Time_Truncate(DataGridView, 6);
        DT_Cell_Date_Truncate(DataGridView, 7);

        DataGridView.Visible = true;
        Table_Name_Label.Text = "Music archive";
        Table_Name_Label.Visible = true;

        FILTER_1_RB.Text = "Specific author";
        FILTER_2_RB.Text = "Specific genre";
        FILTER_1_RB.Visible = true;
        FILTER_2_RB.Visible = true;
        FILTERS_GroupBox.Visible = true;
    }

    private void broadcastGridToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //query 3
        BROADCAST_GRID_DS.Clear();

```

```

Reset(DataGridView, Table_Name_Label, WORKERS_GroupBox, POSTS_GroupBox,
TIMETABLE_GroupBox, AUTHORS_GroupBox, GENRES_GroupBox, RECORDS_GroupBox,
FILTERS_GroupBox, Add_Button, Delete_Button, Save_Button, NO_FILTERS_RB, FILTER_1_RB, FILTER_2_RB,
FILTER_1_CB, FILTER_2_CB);
BROADCAST_GRID_CB.DataAdapter = BROADCAST_GRID_DA;
BROADCAST_GRID_DA.Fill(BROADCAST_GRID_DS);
DataGridView.Fill(DataGridView, BROADCAST_GRID_DS);
DataGridView.Width = 850;

DT_Cell_Time_Truncate(DataGridView, 0);
DT_Cell_Date_Truncate(DataGridView, 2);
DT_Cell_Date_Truncate(DataGridView, 4);
DT_Cell_Date_Truncate(DataGridView, 6);

DataGridView.Visible = true;
Table_Name_Label.Text = "Broadcast grid";
Table_Name_Label.Visible = true;

FILTER_1_RB.Text = "Specific date";
FILTER_2_RB.Text = "Specific worker";
FILTER_1_RB.Visible = true;
FILTER_2_RB.Visible = true;
FILTERS_GroupBox.Visible = true;
}

private void AboutProgramToolStripMenuItem_Click(object sender, EventArgs e)
{
    QueriesToolStripMenuItem.Enabled = true;
    // about program
    Info info = new Info();
    info.ShowDialog();
}

private void Add_Button_Click(object sender, EventArgs e)
{
    switch (Table_Name_Label.Text)
    {
        case "Workers":
        {
            try
            {
                WORKERS_DS.Tables[0].Rows.Add(W_id_NUD.Value.ToString(), W_fio_TB.Text,
                W_age_NUD.Value.ToString(), W_gender_CB.SelectedItem.ToString(), W_address_TB.Text, W_number_TB.Text,
                W_passport_TB.Text, Get_id(W_post_CB.SelectedItem.ToString(), "post_id", "post_name", "Posts"));
                DataGridView.Rows.Add(W_id_NUD.Value.ToString(), W_fio_TB.Text,
                W_age_NUD.Value.ToString(), W_gender_CB.SelectedItem.ToString(), W_address_TB.Text, W_number_TB.Text,
                W_passport_TB.Text, Get_id(W_post_CB.SelectedItem.ToString(), "post_id", "post_name", "Posts"));
                WORKERS_DA.Update(WORKERS_DS);
            }
            catch
            {
                MessageBox.Show("Data input error! Make sure the data is correct and primary key is unique!");
            }
        }
        break;

        case "Posts":
        {
            try
            {
                POSTS_DS.Tables[0].Rows.Add(P_id_NUD.Value.ToString(), P_name_TB.Text,
                P_salary_NUD.Value.ToString(), P_resp_TB.Text, P_req_TB.Text);
            }
            catch
            {
                MessageBox.Show("Data input error! Make sure the data is correct and primary key is unique!");
            }
        }
        break;
    }
}

```

```

        DataGridView.Rows.Add(P_id_NUD.Value.ToString(), P_name_TB.Text,
P_salary_NUD.Value.ToString(), P_resp_TB.Text, P_req_TB.Text);
        POSTS_DA.Update(POSTS_DS);
    }
    catch
    {
        MessageBox.Show("Data input error! Make sure the data is correct and primary key is unique!");
    }
}
break;

case "Timetable":
{
    try
    {
        TIMETABLE_DS.Tables[0].Rows.Add(T_date_DTP.Value.ToString(),
Get_id(T_worker_CB.SelectedItem.ToString(), "worker_id", "fio", "Workers"), T_time1_DTP.Value.ToString(),
Get_id(T_record1_CB.SelectedItem.ToString(), "record_id", "record_name", "Records"),
T_time2_DTP.Value.ToString(), Get_id(T_record2_CB.SelectedItem.ToString(), "record_id", "record_name",
"Records"), T_time3_DTP.Value.ToString(), Get_id(T_record3_CB.SelectedItem.ToString(), "record_id",
"record_name", "Records"));
        DataGridView.Rows.Add(T_date_DTP.Value.ToString(),
Get_id(T_worker_CB.SelectedItem.ToString(), "worker_id", "fio", "Workers"), T_time1_DTP.Value.ToString(),
Get_id(T_record1_CB.SelectedItem.ToString(), "record_id", "record_name", "Records"),
T_time2_DTP.Value.ToString(), Get_id(T_record2_CB.SelectedItem.ToString(), "record_id", "record_name",
"Records"), T_time3_DTP.Value.ToString(), Get_id(T_record3_CB.SelectedItem.ToString(), "record_id",
"record_name", "Records"));
        TIMETABLE_DA.Update(TIMETABLE_DS);
    }
    catch
    {
        MessageBox.Show("Data input error! Make sure the data is correct and primary key is unique!");
    }
}
break;

case "Authors":
{
    try
    {
        AUTHORS_DS.Tables[0].Rows.Add(A_id_NUD.Value.ToString(), A_name_TB.Text,
A_desc_TB.Text);
        DataGridView.Rows.Add(A_id_NUD.Value.ToString(), A_name_TB.Text, A_desc_TB.Text);
        AUTHORS_DA.Update(AUTHORS_DS);
    }
    catch
    {
        MessageBox.Show("Data input error! Make sure the data is correct and primary key is unique!");
    }
}
break;

case "Genres":
{
    try
    {
        GENRES_DS.Tables[0].Rows.Add(G_id_NUD.Value.ToString(), G_name_TB.Text, G_desc_TB.Text);
        DataGridView.Rows.Add(G_id_NUD.Value.ToString(), G_name_TB.Text, G_desc_TB.Text);
        GENRES_DA.Update(GENRES_DS);
    }
    catch
    {
        MessageBox.Show("Data input error! Make sure the data is correct and primary key is unique!");
    }
}

```

					12.121.B31.05.KP.ПЗ	Лист
Изм.	Лист	№ Документа	Подпись	Дат		38

```

    }
}
break;

case "Records":
{
    try
    {
        string prod_year = R_date_DTP.Value.ToString().Remove(0, 6);
        prod_year = prod_year.Remove(4);
        RECORDS_DS.Tables[0].Rows.Add(R_id_NUD.Value.ToString(), R_name_TB.Text,
Get_id(R_author_CB.SelectedItem.ToString(), "author_id", "author_name", "Authors"), R_album_TB.Text, prod_year,
Get_id(R_genre_CB.SelectedItem.ToString(), "genre_id", "genre_name", "Genres"), R_date_DTP.Value.ToString(),
R_length_DTP.Value.ToString(), R_rating_NUD.Value.ToString());
        DataGridView.Rows.Add(R_id_NUD.Value.ToString(), R_name_TB.Text,
Get_id(R_author_CB.SelectedItem.ToString(), "author_id", "author_name", "Authors"), R_album_TB.Text, prod_year,
Get_id(R_genre_CB.SelectedItem.ToString(), "genre_id", "genre_name", "Genres"), R_date_DTP.Value.ToString(),
R_length_DTP.Value.ToString(), R_rating_NUD.Value.ToString());
        RECORDS_DA.Update(RECORDS_DS);
    }
    catch
    {
        MessageBox.Show("Data input error! Make sure the data is correct and primary key is unique!");
    }
}
break;
}
}

private void Delete_Button_Click(object sender, EventArgs e)
{
    switch (Table_Name_Label.Text)
    {
        case "Workers":
        {
            Table_Remove(DataGridView, WORKERS_DS, WORKERS_DA);
        }
        break;

        case "Posts":
        {
            Table_Remove(DataGridView, POSTS_DS, POSTS_DA);
        }
        break;

        case "Timetable":
        {
            Table_Remove(DataGridView, TIMETABLE_DS, TIMETABLE_DA);
        }
        break;

        case "Authors":
        {
            Table_Remove(DataGridView, AUTHORS_DS, AUTHORS_DA);
        }
        break;

        case "Genres":
        {
            Table_Remove(DataGridView, GENRES_DS, GENRES_DA);
        }
        break;
    }
}

```

```

        case "Records":
        {
            Table_Remove(DataGridView, RECORDS_DS, RECORDS_DA);
        }
        break;
    }
}

private void Save_Button_Click(object sender, EventArgs e)
{
    WORKERS_DA.Update(WORKERS_DS);
    POSTS_DA.Update(POSTS_DS);
    TIMETABLE_DA.Update(TIMETABLE_DS);
    AUTHORS_DA.Update(AUTHORS_DS);
    GENRES_DA.Update(GENRES_DS);
    RECORDS_DA.Update(RECORDS_DS);
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (FILTER_1_CB.SelectedIndex < 0)
    {
        return;
    }
    switch (Table_Name_Label.Text)
    {
        case "Personnel department":
        {
            PERSONNEL_DEPARTMENT_DA.SelectCommand.CommandText = "SELECT Workers.worker_id,
Workers.fio, Workers.age, Workers.gender, Workers.address, Workers.phone_number, Workers.passport_details,
Posts.post_name FROM Workers INNER JOIN Posts ON Workers.post_id = Posts.post_id WHERE Posts.post_name = '" +
FILTER_1_CB.SelectedItem.ToString() + "'";
            personnelDepartmentToolStripMenuItem_Click(sender, e);
        }
        break;

        case "Music archive":
        {
            MUSIC_ARCHIVE_DA.SelectCommand.CommandText = "SELECT Records.record_id,
Records.record_name, Authors.author_name, Records.album, Records.prod_year, Genres.genre_name,
Records.record_date, Records.length, Records.rating FROM Genres INNER JOIN (Authors INNER JOIN Records ON
Authors.author_id = Records.author_id) ON Genres.genre_id = Records.genre_id WHERE Authors.author_name = '" +
FILTER_1_CB.SelectedItem.ToString() + "'";
            musicArchiveToolStripMenuItem_Click(sender, e);
        }
        break;

        case "Broadcast grid":
        {
            // "date" is a converted input of type 01.01.2022 -> #01/01/2022#
            string date = "#" + FILTER_1_CB.SelectedItem.ToString().Remove(2) + "/" +
FILTER_1_CB.SelectedItem.ToString().Remove(0, 3).Remove(2) + "/" +
FILTER_1_CB.SelectedItem.ToString().Remove(0, 6).Remove(4) + "#";
            BROADCAST_GRID_DA.SelectCommand.CommandText = "SELECT Timetable.working_date,
Workers.fio, Timetable.time1, Records.record_name AS record1, Timetable.time2, Records_1.record_name AS
record2, Timetable.time3, Records_2.record_name AS record3 FROM Records AS Records_2 INNER JOIN (Records
AS Records_1 INNER JOIN (Records INNER JOIN (Workers INNER JOIN Timetable ON Workers.worker_id =
Timetable.worker_id) ON Records.record_id = Timetable.record1_id) ON Records_1.record_id =
Timetable.record2_id) ON Records_2.record_id = Timetable.record3_id WHERE Timetable.working_date = " + date +
";";
            broadcastGridToolStripMenuItem_Click(sender, e);
        }
        break;
    }
}

```

					12.121.B31.05.KP.ПЗ	Лист
Изм.	Лист	№ Документа	Подпись	Дат		40


```

    }
}

private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    if (FILTER_2_CB.SelectedIndex < 0)
    {
        return;
    }
    switch (Table_Name_Label.Text)
    {
        case "Personnel department":
        {
            //nothing
        }
        break;

        case "Music archive":
        {
            MUSIC_ARCHIVE_DA.SelectCommand.CommandText = "SELECT Records.record_id,
Records.record_name, Authors.author_name, Records.album, Records.prod_year, Genres.genre_name,
Records.record_date, Records.length, Records.rating FROM Genres INNER JOIN (Authors INNER JOIN Records ON
Authors.author_id = Records.author_id) ON Genres.genre_id = Records.genre_id WHERE Genres.genre_name = '" +
FILTER_2_CB.SelectedItem.ToString() + "'";
            musicArchiveToolStripMenuItem_Click(sender, e);
        }
        break;

        case "Broadcast grid":
        {
            BROADCAST_GRID_DA.SelectCommand.CommandText = "SELECT Timetable.working_date,
Workers.fio, Timetable.time1, Records.record_name AS record1, Timetable.time2, Records_1.record_name AS
record2, Timetable.time3, Records_2.record_name AS record3 FROM Records AS Records_2 INNER JOIN (Records
AS Records_1 INNER JOIN (Records INNER JOIN (Workers INNER JOIN Timetable ON Workers.worker_id =
Timetable.worker_id) ON Records.record_id = Timetable.record1_id) ON Records_1.record_id =
Timetable.record2_id) ON Records_2.record_id = Timetable.record3_id WHERE Workers.fio = '" +
FILTER_2_CB.SelectedItem.ToString() + "'";
            broadcastGridToolStripMenuItem_Click(sender, e);
        }
        break;
    }
}

//preventing editing of ComboBoxes in unprotected area of code
private void FILTER_1_CB_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = true;
}

private void FILTER_2_CB_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = true;
}

private void NO_FILTERS_RB_CheckedChanged(object sender, EventArgs e)
{
    if (NO_FILTERS_RB.Checked)
    {
        switch (Table_Name_Label.Text)
        {
            case "Personnel department":
            {

```

```

PERSONNEL_DEPARTMENT_DA.SelectCommand.CommandText = "SELECT Workers.worker_id,
Workers.fio, Workers.age, Workers.gender, Workers.address, Workers.phone_number, Workers.passport_details,
Posts.post_name FROM Workers INNER JOIN Posts ON Workers.post_id = Posts.post_id";
FILTER_1_CB.Visible = false;
FILTER_2_CB.Visible = false;
personnelDepartmentToolStripMenuItem_Click(sender, e);
}
break;

case "Music archive":
{
MUSIC_ARCHIVE_DA.SelectCommand.CommandText = "SELECT Records.record_id,
Records.record_name, Authors.author_name, Records.album, Records.prod_year, Genres.genre_name,
Records.record_date, Records.length, Records.rating FROM Genres INNER JOIN (Authors INNER JOIN Records ON
Authors.author_id = Records.author_id) ON Genres.genre_id = Records.genre_id";
FILTER_1_CB.Visible = false;
FILTER_2_CB.Visible = false;
musicArchiveToolStripMenuItem_Click(sender, e);
}
break;

case "Broadcast grid":
{
BROADCAST_GRID_DA.SelectCommand.CommandText = "SELECT Timetable.working_date,
Workers.fio, Timetable.time1, Records.record_name AS record1, Timetable.time2, Records_1.record_name AS
record2, Timetable.time3, Records_2.record_name AS record3 FROM Records AS Records_2 INNER JOIN (Records
AS Records_1 INNER JOIN (Records INNER JOIN (Workers INNER JOIN Timetable ON Workers.worker_id =
Timetable.worker_id) ON Records.record_id = Timetable.record1_id) ON Records_1.record_id =
Timetable.record2_id) ON Records_2.record_id = Timetable.record3_id";
FILTER_1_CB.Visible = false;
FILTER_2_CB.Visible = false;
broadcastGridToolStripMenuItem_Click(sender, e);
}
break;
}
QueriesToolStripMenuItem.Enabled = true;
}
}

private void FILTER_1_RB_CheckedChanged(object sender, EventArgs e)
{
if (FILTER_1_RB.Checked)
{
switch (Table_Name_Label.Text)
{
case "Personnel department":
{
FILTER_1_CB.Visible = true;
FILTER_2_CB.Visible = false;
ComboBox_Fill(FILTER_1_CB, POSTS_DS, 1);
}
break;

case "Music archive":
{
FILTER_1_CB.Visible = true;
FILTER_2_CB.Visible = false;
ComboBox_Fill(FILTER_1_CB, AUTHORS_DS, 1);
}
break;

case "Broadcast grid":
{

```

```

        FILTER_1_CB.Visible = true;
        FILTER_2_CB.Visible = false;
        ComboBox_Fill(FILTER_1_CB, TIMETABLE_DS, 0);
    }
    break;
}
QueriesToolStripMenuItem.Enabled = false;
}
}

private void FILTER_2_RB_CheckedChanged(object sender, EventArgs e)
{
    if (FILTER_2_RB.Checked)
    {
        switch (Table_Name_Label.Text)
        {
            case "Personnel department":
            {
                // nothing
            }
            break;

            case "Music archive":
            {
                FILTER_1_CB.Visible = false;
                FILTER_2_CB.Visible = true;
                ComboBox_Fill(FILTER_2_CB, GENRES_DS, 1);
            }
            break;

            case "Broadcast grid":
            {
                FILTER_1_CB.Visible = false;
                FILTER_2_CB.Visible = true;
                ComboBox_Fill(FILTER_2_CB, WORKERS_DS, 1);
            }
            break;
        }
        QueriesToolStripMenuItem.Enabled = false;
    }
}
}
}

```