

Data oddania	22.06.2022
Nazwa kursu	Bazy danych
Prowadzący	Dr inż. Roman Ptak
Dzień i godzina	środa 9.45
Kod grupy zajęciowej	Y00-37j
Nr grupy	6
Temat projektu	Sklep z grami komputerowymi
Skład grupy	
Imię i nazwisko	Nr indeksu
Mikołaj Saramonowicz	252964
Michał Wieczorek	252872
Marcin Młodziński	252902

1 Założenia ogólne

Realizowana baza danych będzie wykorzystywana do obsługi sklepu internetowego z grami komputerowymi.

Baza umożliwi przeglądanie produktów wyróżniając:

- nośnik produktu (pudełkowa/cyfrowa)
- platformę (PC, PS4, PS5, Xbox One, Xbox Series...)
- gatunek gry (FPS, RPG, symulator, strategia, przygodowa...)
- ocenę użytkowników
- cenę

Przewidywany orientacyjny rozmiar bazy danych to do stu pozycji w ofercie, każda dostępna w ilości do kilkuset kopii fizycznych lub cyfrowych, a liczbę pracowników sklepu szacuje się na od 3 do 10-20. (ogólność zastosowanych rozwiązań pozwoli na ewentualne zwiększenie tych limitów w razie potrzeby). Każdy produkt dysponuje własnym numerem katalogowym, dla każdego artykułu jest też przewidziany możliwy typ wysyłki (w zależności od typu produktu: email lub kurier/paczkomat) oraz przechowywana jest informacja o jego liczebności.

Spis treści

1	Założenia ogólne	1
2	Cel i zakres projektu	3
3	Funkcjonalności aplikacji	4
3.1	Zwykły pracownik	4
3.2	Administrator	4
4	Wykorzystywane rozwiązania	5
5	Modele projektowe bazy danych	6
5.1	Model konceptualny	6
5.2	Model logiczno-fizyczny	7
5.3	Diagram przypadków użycia	8
6	Elementy związane z implementacją bazy danych	9
6.1	Tabele - wersja opisowa	9
6.2	Tabele - wersja fizyczna	13
6.2.1	Pola - wygenerowane przez dbdiagram.io	13
6.2.2	Klucze obce - wygenerowane przez dbdiagram.io	13
6.2.3	Indeksy - dodane ręcznie w programie MySQL Workbench	14
6.3	Widoki - wersja opisowa	14
6.4	Widoki - wersja fizyczna	16
6.4.1	Implementacja widoku - kod napisany ręcznie	17
6.5	Prawa użytkowników w bazie danych - wersja opisowa	17
6.5.1	Klient	17
6.5.2	Pracownik sklepu	17
6.5.3	Administrator	17
6.6	Prawa użytkowników w bazie danych - wersja fizyczna	18
6.6.1	Implementacja dodania ograniczonych uprawnień - pracownik	18
6.6.2	Implementacja dodania pełnych uprawnień - administrator	18
6.7	Procedury składowane - wersja opisowa	19
6.7.1	Procedury klienta (nie dostępne w aplikacji)	19
6.7.2	Procedury pracownika (dostępne w aplikacji)	21
6.7.3	Procedury administratora (dostępne w aplikacji)	22
6.8	Wyzwalacze	24
6.9	Procedury składowane i wyzwalacze - testy implementacji	25

7	Elementy związane z implementacją aplikacji	41
7.1	Model aplikacji dostępowej	41
7.2	Bezpieczeństwo aplikacji oraz bazy danych	43
7.3	Instrukcja obsługi i instalacji aplikacji	44
7.4	Testy poprawności działania - od strony użytkownika	47
7.5	Testy poprawności działania - realizacja w kodzie źródłowym	50
8	Wykorzystane materiały	52
9	Spis rysunków z etapu testowania procedur i aplikacji	52
10	Spis modeli i opisów tabel	54
11	Podsumowanie	54

2 Cel i zakres projektu

Projekt zakłada stworzenie bazy danych na potrzeby sklepu internetowego w zakresie umożliwiającym pracownikom sklepu wewnętrzną obsługę tej bazy za pośrednictwem specjalnej aplikacji. Zakładamy współpracę z firmą trzecią w całości odpowiadającą za stronę internetową, przez którą do bazy danych sklepu będą mieli dostęp klienci chcący dokonać transakcji. W momencie dokonania transakcji na stronie, do magazynu zostanie wysłana informacja o zamówieniu, które ma zostać przygotowane przez pracowników. Projekt kładzie nacisk na poprawnie działającą bazę danych oraz aplikację desktopową dla pracowników. Przewidywano możliwość rozszerzenia projektu o aplikację webową lub mobilną, jednak z powodu ograniczonego czasu nie udało się tego wykonać w ramach obecnie realizowanego kursu.

3 Funkcjonalności aplikacji

W przyjętym rozwiązaniu planowane funkcjonalności podzielono ze względu na poziom uprawnień zalogowanego użytkownika.

3.1 Zwykły pracownik

- Przeglądanie produktów
- Filtrowanie i sortowanie produktów
- Usuwanie i dodawanie nowych produktów do oferty
- Ręczna edycja danych przypisanych produktowi
- Obsługa zamówień i wysyłek

3.2 Administrator

- Wszystkie uprawnienia zwykłego pracownika
- Dodawanie i usuwanie kategorii
- Przeglądanie, dodawanie, usuwanie i modyfikacja kont pracowników i innych administratorów

4 Wykorzystywane rozwiązania

Jako typ struktury wybraliśmy relacyjną bazę danych. Takie bazy danych są oparte na modelu relacyjnym — jest to prosty i intuicyjny sposób przedstawiania danych w tabelach. Tabele zawierają określone informacje, które składowane są w dedykowanych kolumnach. Na kolumnie ID założony jest unikatowy klucz, który spaja dwie tabele, tworząc między nimi swoistą relację. Na przykład gdyby w naszym sklepie internetowym została wykonana transakcja, to dzięki relacjom dane klienta zostałyby powiązane z zamówionymi produktami.

Relacyjna baza danych zbudowana jest przy użyciu standardowego języka zapytań SQL. Takie podejście zapewnia matematyczną spójność oraz wydajność systemu dzięki algebrze relacyjnej. Projekt realizowany będzie w środowisku MySQL z wykorzystaniem programu MySQL Workbench. Zaletami MySQL jest możliwość obsługi bardzo dużej ilości zapytań na sekundę czy uniwersalność, która gwarantuje, że strona będzie obsługiwana na różnych systemach.

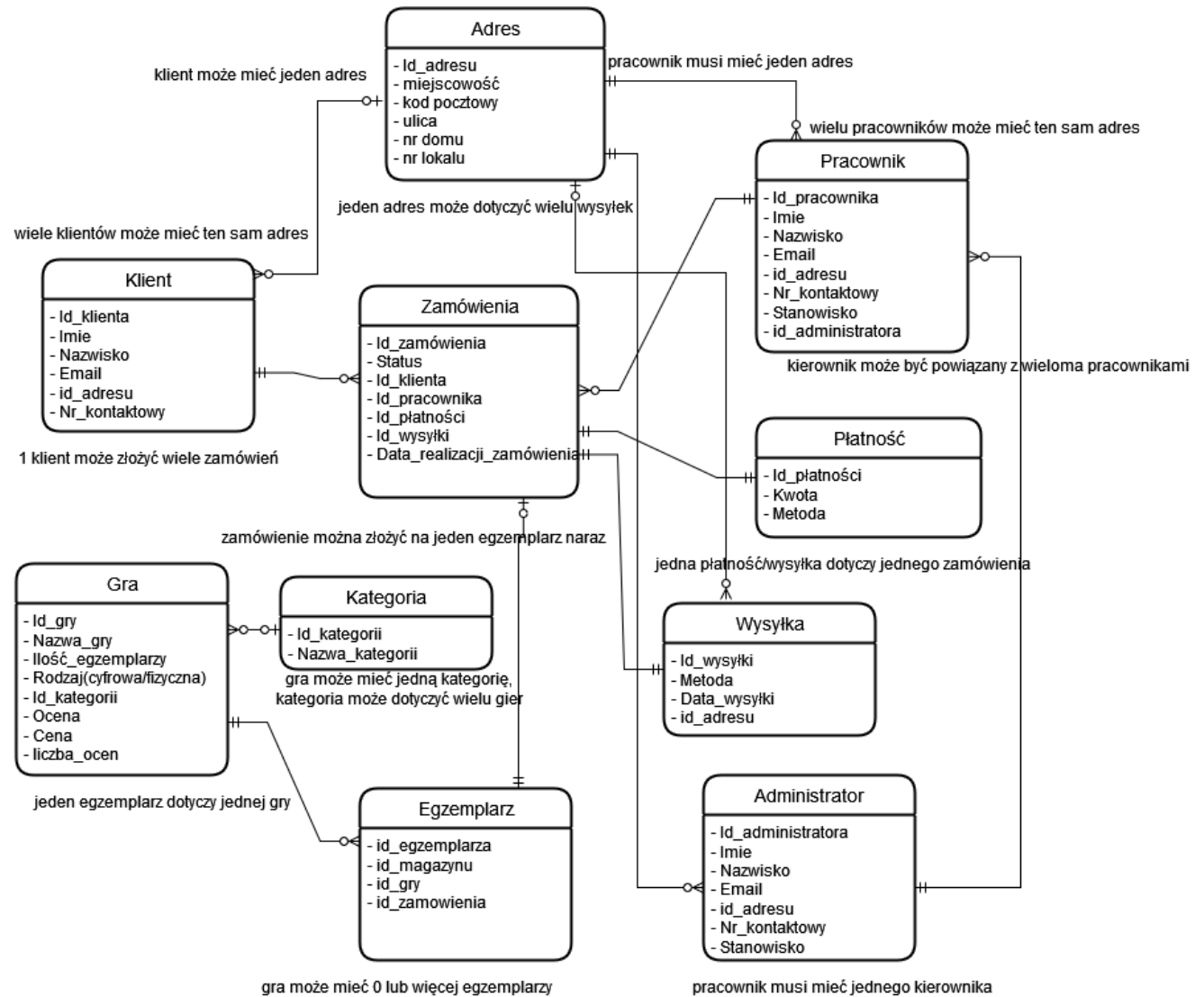
Bazę danych sklepu internetowego powinna cechować:

- Prostota i funkcjonalność - struktura bazy danych powinna być możliwie prosta i przejrzysta. Jednocześnie nie powinno to wpływać na funkcjonalność i wygodę użytkownika. Rozwiązanie oparte na tabelach jest łatwe do zrozumienia i spełnia wszystkie z uprzednio wymienionych wymagań.
- Wysoka wydajność - konieczne jest zapewnienie szybkiego dostępu do określonych informacji. Łatwa i szybko interakcja pomiędzy użytkownikiem a bazą ma zapewnić bezproblemowe zakupy i wpływa bezpośrednio na odczucia oraz zadowolenie klienta.
- Przejrzystość i niezależność danych - tabele w bazie relacyjnej są od siebie niezależne, dzięki czemu są łatwo dostępne i modyfikowalne.
- Skalowalność i wysoka dostępność - tworząc bazę danych dla sklepu internetowego zakładamy, że liczba użytkowników oraz produktów będzie się zwiększać w przyszłości.

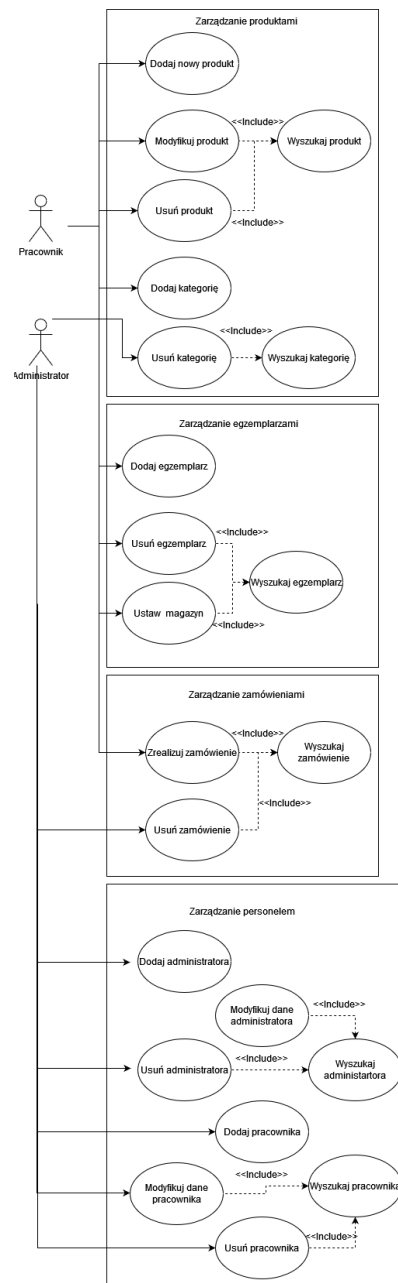
Do napisania aplikacji dostępowej wykorzystano biblioteki i funkcjonalności języka C#, posiłkując się środowiskiem deweloperskim Visual Studio firmy Microsoft.

5 Modele projektowe bazy danych

5.1 Model konceptualny



5.3 Diagram przypadków użycia



6 Elementy związane z implementacją bazy danych

6.1 Tabele - wersja opisowa

Nazwa atrybutu	Dziedzina	Indeks	Maska	OBL(+)/OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródła danych
id_klienta	int	11		+			+	PK	Zamowienia, Gra	SZBD
imie	varchar(30)	12		+						USER
nazwisko	varchar(30)	12		+						USER
email	varchar(30)	13		+			+			USER
nr_kontaktowy	varchar(15)			-						USER
id_adresu	int			-				FK	Adres	SZBD

Tabela 1: klient

Nazwa atrybutu	Dziedzina	Indeks	Maska	OBL(+)/OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródła danych
id_zamowienia	int	11		+			+	PK		SZBD
id_klienta	int	12		-				FK	Klient	SZBD
id_pracownika	int			-				FK	Pracownik	SZBD
id_platnosci	int			-			+	FK	Platnosc	SZBD
status_zamowienia	enum			+	"nieoplacone"	"nieoplacone"/"oplacone"/"zrealizowane"				USER
id_wysylki	int			-			+	FK	Wysylka	SZBD
data_realizacji	datetime		__--"---"----	-						USER

Tabela 2: zamowienie

Nazwa atrybutu	Dziedzina	Indeks	Maska	OBL(+)/OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródła danych
id_pracownika	int	11		+			+	PK	Zamowienia	SZBD
imie	varchar(30)	12		+						USER
nazwisko	varchar(30)	12		+						USER
nr_kontaktowy	varchar(15)			+						USER
stanowisko	varchar(50)	13		+						USER
id_administradora	int			+				FK	Administrator	SZBD
email	varchar(30)			+			+			USER
id_adresu	int			+				FK	Adres	SZBD

Tabela 3: pracownik

Nazwa atrybutu	Dziedzina	Indeks	Maska	OBL(+)/OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródła danych
id_administradora	int	11		+			+	PK	Pracownik	SZBD
imie	varchar(30)	12		+						USER
nazwisko	varchar(30)	12		+						USER
email	varchar(30)			+			+			USER
nr_kontaktowy	varchar(15)			+						USER
stanowisko	varchar(50)	13		+						USER
id_adresu	int			+				FK	Adres	SZBD

Tabela 4: administrator

Nazwa atrybutu	Dziedzina	Indeks	Maska	OBL(+)/OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródła danych
id_gry	int	11		+			+	PK	Zamowienie	SZBD
Nazwa_gry	varchar(100)	12		+						USER
ilosc_egzemplarzy	int			+						USER
rodzaj	enum			+						USER
id_kategorii	int			-				FK	Kategoria	USER
cena	double			+		>0				USER
ocena	double			-		>=0			Klient	USER

Tabela 5: gra

Nazwa atrybutu	Dziedzina	Indeks	Maska	OBL(+)/OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródła danych
id_platnosci	int	11		+			+	PK	Zamowienie	SZBD
kwota	double			+		>0				USER
metoda	enum			+		"karta"/"przelew"/"blik"				USER

Tabela 6: platnosc

Nazwa atrybutu	Dziedzina	Indeks	Maska	OBL(+)/OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródła danych
id_wysylki	int	11		+			+	PK	Zamowienie	SZBD
metoda	enum			+						USER
data_wysylki	date		-- ------	-						USER
id_adresu	int			-				FK	Adres	SZBD

Tabela 7: wysylka

Nazwa atrybutu	Dziedzina	Indeks	Maska	OBL(+)/OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródła danych
id_kategorii	int	11		+			+	PK	Gra	SZBD
Nazwa_kategorii	varchar(30)	12		+			+			USER

Tabela 8: kategoria

Nazwa atrybutu	Dziedzina	Indeks	Maska	OBL(+)/OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródła danych
id_egzemplarza	int	11		+			+	PK	Gra	SZBD
id_magazynu	int	12		+						USER
id_gry	int	13		+				FK	Gra	USER
id_zamowienia	int	14		-				FK	Zamowienie	USER

Tabela 9: egzemplarz

Nazwa atrybutu	Dziedzina	Indeks	Maska	OBL(+)/OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródła danych
id_adresu	int	11		+			+	PK	Klient, Wysylka, Pracownik, Administrator	SZBD
miasto	varchar(30)			+						USER
kod_pocztowy	char(6)		------	+						USER
ulica	varchar(30)			-						USER
numer_domu	int			+		> 0				USER
numer_lokalu	int			-		> 0				USER

Tabela 10: Adres

Nazwa atrybutu	Znaczenie
id_klienta	Unikalny identyfikator klienta, automatycznie nadawany przez system, klucz główny
imie	Pierwsze imię klienta, ciąg znaków o określonej maksymalnej długości
nazwisko	Nazwisko klienta, ciąg znaków o określonej maksymalnej długości
email	Adres poczty elektronicznej klienta, ciąg znaków o ograniczonej maksymalnej długości
nr_kontaktowy	Kontaktowy numer telefoniczny do klienta, ciąg znaków o ograniczonej maksymalnej długości
id_adresu	id encji przechowującej opcjonalne dane adresowe podane przez klienta

Tabela 11: klient - opis

Nazwa atrybutu	Znaczenie
id_pracownika	Unikalny identyfikator pracownika, automatycznie nadawany przez system, klucz główny
imie	Pierwsze imię pracownika, ciąg znaków o ograniczonej maksymalnej długości
nazwisko	Nazwisko pracownika, ciąg znaków o ograniczonej maksymalnej długości
email	Adres poczty elektronicznej pracownika, ciąg znaków o ograniczonej maksymalnej długości
nr_kontaktowy	Kontaktowy numer telefoniczny do pracownika, ciąg znaków o ograniczonej maksymalnej długości
stanowisko	Stanowisko pracy pracownika, ciąg znaków o ograniczonej maksymalnej długości
id_administradora	ID przełożonego pracownika, liczba naturalna, klucz obcy
id_adresu	id encji przechowującej obowiązkowe dane adresowe podane przez pracownika

Tabela 12: pracownik - opis

Nazwa atrybutu	Znaczenie
id_administradora	Unikalny identyfikator administratora, automatycznie nadawany przez system, klucz główny
imie	Pierwsze imię administratora, ciąg znaków o ograniczonej maksymalnej długości
nazwisko	Nazwisko administratora, ciąg znaków o ograniczonej maksymalnej długości
email	Adres poczty elektronicznej administratora, ciąg znaków o ograniczonej maksymalnej długości
id_adresu	id encji przechowującej obowiązkowe dane adresowe podane przez administratora
nr_kontaktowy	Kontaktowy numer telefoniczny do administratora, ciąg znaków o ograniczonej maksymalnej długości
stanowisko	Stanowisko pracy administratora, ciąg znaków o ograniczonej maksymalnej długości

Tabela 13: administrator - opis

Nazwa atrybutu	Znaczenie
id_zamowienia	Unikalny identyfikator zamówienia, automatycznie nadawany przez system, klucz główny
id_klienta	Identyfikator klienta, który wykonał zamówienie, liczba naturalna, klucz obcy
id_pracownika	Identyfikator pracownika odpowiedzialnego za zamówienie, liczba naturalna, klucz obcy
id_platnosci	Identyfikator płatności za zamówienie, liczba naturalna, klucz obcy
status	Status zamówienia, "nieopłacone", "opłacone" lub "zrealizowane"
id_wysylki	Identyfikator wysyłki wybranej do zamówienia, liczba naturalna, klucz obcy
data_realizacji	Data realizacji zamówienia, razem z godziną

Tabela 14: Zamówienie - opis

Nazwa atrybutu	Znaczenie
id_gry	Unikalny identyfikator gry, automatycznie nadawany przez system, klucz główny
id_egzemplarzy	Identyfikator egzemplarza konkretnego produktu, liczba naturalna, klucz obcy
id_kategorii	Identyfikator kategorii, do której należy produkt, liczba naturalna, klucz obcy
Nazwa_gry	Nazwa gry, ciąg znaków o ograniczonej maksymalnej długości
rodzaj	Rodzaj gry, zakładamy, że gra jest "cyfrowa" lub "pudełkowa"
cena	Cena konkretnej gry, liczba double większa od 0
ocena	opcjonalna ocena gry, liczba typu double
ilosc_egzemplarzy	liczba naturalna określająca ilość dostępnych kopii produktu (tj. istniejących oraz niezamówionych)

Tabela 15: gra - opis

Nazwa atrybutu	Znaczenie
id_platnosci	Unikalny identyfikator płatności, nadawany poprzez zamówienie, klucz główny
kwota	Kwota należności, dodatnia liczba rzeczywista
metoda	Metoda płatności, "przelew bankowy" lub "blik" lub "karta"

Tabela 16: platnosc - opis

Nazwa atrybutu	Znaczenie
id_wysylki	Unikalny identyfikator wysyłki, nadawany poprzez zamówienie, klucz główny
metoda	Metoda wysyłki, "na adres email" lub "na adres fizyczny" lub "punkt odbioru"
data_wysylki	Data wysyłki, do momentu wysłania null
id_adresu	id encji przechowującej dane adresowe dla przesyłek fizycznych

Tabela 17: Wysyłka - opis

Nazwa atrybutu	Znaczenie
id_kategorii	Unikalny identyfikator kategorii, automatycznie nadawany przez system, klucz główny
Nazwa_kategorii	Nazwa kategorii, ciąg znaków o zadanej maksymalnej długości

Tabela 18: kategoria - opis

Nazwa atrybutu	Znaczenie
id_egzemplarza	Unikalny identyfikator egzemplarza, automatycznie nadawany przez system, klucz główny
id_magazynu	Identyfikator magazynu, w którym znajduje się egzemplarz, liczba naturalna lub null
id_gry	Identyfikator gry, której odpowiada egzemplarz, liczba naturalna
id_zamowienia	Identyfikator zamówienia, któremu odpowiada egzemplarz, liczba naturalna

Tabela 19: egzemplarz - opis

Nazwa atrybutu	Znaczenie
id_adresu	Unikalny identyfikator adresu, automatycznie nadawany przez system, klucz główny
miasto	nazwa miejscowości, ciąg znaków o ograniczonej maksymalnej długości
kod_pocztowy	Kod pocztowy placówki pocztowej, format XX-XXX, gdzie w miejsce X należy wstawić liczby
ulica	opcjonalna, ciąg znaków o ograniczonej maksymalnej długości
numer_domu	liczba naturalna
numer_lokalu	liczba naturalna, opcjonalna

Tabela 20: Adres - opis

6.2 Tabele - wersja fizyczna

Poniżej przedstawiono przykładowy kod źródłowy pozwalający na utworzenie elementów tabeli

1 • `SELECT * FROM baza.administrator;`

id_administratora	imie	nazwisko	email	nr_kontakowy	stanowisko	id_adresu
29	Grzegorz	Kowalski	kowal@wp.pl	+48123456789	Kierownik działu 1	44
30	Wojciech	Nowak	nowak@wp.pl	+48143456789	Kierownik działu 2	45
31	Mateusz	Florczak	mat@onet.pl	+48123456789	CEO	49

6.2.1 Pola - wygenerowane przez dbdiagram.io

```
CREATE TABLE `Administrator` (
  `id_administratora` int PRIMARY KEY AUTO_INCREMENT COMMENT 'Primary key',
  `imie` varchar(30) NOT NULL COMMENT 'not null, Pierwsze imie administratora',
  `nazwisko` varchar(30) NOT NULL COMMENT 'not null, Jedno nazwisko administratora',
  `email` varchar(30) UNIQUE NOT NULL COMMENT 'not null, email kontaktowy administratora',
  `nr_kontakowy` varchar(15) NOT NULL COMMENT 'Numer telefonu administratora',
  `stanowisko` varchar(50) NOT NULL COMMENT 'Not null, dokładna nazwa stanowiska w firmie',
  `id_adresu` int NOT NULL COMMENT 'Not null, adres z którym powiazany jest administrator'
);
```

6.2.2 Klucze obce - wygenerowane przez dbdiagram.io

```
ALTER TABLE `Administrator` ADD FOREIGN KEY (`id_adresu`) REFERENCES `adres` (`id_adresu`);
```

6.2.3 Indeksy - dodane ręcznie w programie MySQL Workbench

```

109 • CREATE INDEX `Administrator_index_10` ON `Administrator` (`imie`, `nazwisko`);
110
111 • CREATE INDEX `Administrator_index_11` ON `Administrator` (`stanowisko`);
112

```

6.3 Widoki - wersja opisowa

Na poniższych grafikach zaprezentowano przykładowy przegląd dostępnych w bazie danych widoków (perspektyw).

Nazwa widoku: przeglądanie_produkow – widok klienta

GRA				KATEGORIA
Nazwa_gry	Rodzaj	Ocena	Cena	Nazwa_kategorii
"Gra1"	cyfrowa	1	120,00	Wyścigowa
"Gra2"	fizyczna	3	220,00	Przygodowa
"Gra3"	fizyczna	5	128,00	Strategiczna

Nazwa widoku: zamowienie_produkow – widok klienta

GRA			ZAMOWIENIE			PLATNOSC		WYSYLKA	
Nazwa_gry	Rodzaj	Cena	Status	Data_realizacji_zamowienia	Id_zamowienia	Kwota	Metoda	Metoda	Data_wysylki
"Gra1"	cyfrowa	120,00	Zrealizowane	01.04.2022	101	120,00	Przelew	email	01.04.2022
"Gra2"	fizyczna	220,00	Zrealizowane	05.04.2022	102	220,00	Blik	Punkt odbioru	06.04.2022
"Gra3"	fizyczna	128,00	Oplacone	21.04.2022	103	128,00	Przelew	Punkt odbioru	-

ADRES				
Miasto	Kod_pocztowy	ulica	nr_domu	nr_lokalu
-	-	-	-	-
Wrocław	50-002	Skłódowskiej-Curie	15	13
Kraków	30-001	Kościuszki	8	-

Nazwa widoku: przegląd_zamówień – widok pracownika i administratora

ZAMÓWIENIE				KLIENT				WYSYŁKA		EGZEMPLARZ	
Id_zamówienia	Id_klienta	Status	Data_realizacji_zamówienia	Imie	Nazwisko	Email	Nr_kontak.	Data_wysyłki	Metoda_wysyłki	Id_egzemplarza	Id_magazynu
101	22	Zrealizowane	01.04.2022	Michał	Kowalski	kowal@gmail.com	128569096	02.04.2022	email	123	1
102	23	Zrealizowane	05.04.2022	Robert	Nowak	robnow@gmail.com	727548323	06.04.2022	punkt	431	2
103	24	Opłacone	21.04.2022	Walery	Poniatowski	poni@gmail.com	235428700	-	kurier	634	5

GRA			ADRES				
Nazwa_gry	Ilosc-egzemplarzy	Rodzaj	Miasto	Kod_pocztowy	ulica	nr_domu	nr_lokalu
"Gra1"	1	cyfrowa	-	-	-	-	-
"Gra2"	10	fizyczna	Wrocław	50-002	Skłódowskiej-Curie	15	13
"Gra3"	28	fizyczna	Kraków	30-001	Kościuszki	8	-

Nazwa widoku: Przeglądanie_produkty – widok pracownika i administratora

GRA				KATEGORIA		EGZEMPLARZ	
id gry	Nazwa_gry	Ilosc-egzemplarzy	Rodzaj	Nazwa_kategorii	id_kategorii	Id_egzemplarza	id magazynu
101	"Gra1"	1	cyfrowa	Wyścigowa	5	134	1
102	"Gra2"	10	fizyczna	Przygodowa	6	635	2
103	"Gra3"	28	fizyczna	Strategiczna	7	234	5

Nazwa widoku: Przegląd_danych_klienta – widok klienta

KLIENT				ADRES				
Imie	Nazwisko	Email	Nr_kontak.	Miasto	Kod_pocztowy	ulica	nr_domu	nr_lokalu
Michał	Kowalski	kowal@gmail.com	128569096	-	-	-	-	-
Robert	Nowak	robnow@gmail.com	727548323	Wrocław	50-002	Skłódowskiej-Curie	15	13
Walery	Poniatowski	poni@gmail.com	235428700	Kraków	30-001	Kościuszki	8	-

Nazwa widoku: Przegląd_danych_pracownika – widok pracownika i administratora

PRACOWNIK						ADRES				
Imie	Nazwisko	Email	Nr_kontak.	Stanowisko	id_administratora	Miasto	Kod_poczty	ulica	nr_domu	nr_lokalu
Michał	Kowalski	kowal@gmail.com	128569096	Pracownik działu dystrybucji cyfrowej nr 1	1	Wrocław	50-002	Grabiszyska	32	-
Robert	Nowak	robnow@gmail.com	727548323	Pracownik działu dystrybucji fizycznej nr 2	1	Wrocław	50-008	Skłodowskiej-Curie	15	13
Walery	Poniatowski	poni@gmail.com	235428700	Pracownik działu dystrybucji fizycznej nr 2	4	Kraków	30-001	Kościuszki	8	-

Nazwa widoku: Przegląd_danych_administratora – widok administratora

ADMINISTRATOR					ADRES				
Imie	Nazwisko	Email	Nr_kontak.	Stanowisko	Miasto	Kod_poczty	ulica	nr_domu	nr_lokalu
Michał	Kowalski	kowal@gmail.com	128569096	Kierownik działu dystrybucji cyfrowej nr 1	Wrocław	50-002	Grabiszyska	32	-
Robert	Nowak	robnow@gmail.com	727548323	Kierownik działu dystrybucji fizycznej nr 2	Wrocław	50-008	Skłodowskiej-Curie	15	13
Walery	Poniatowski	poni@gmail.com	235428700	Kierownik działu dystrybucji fizycznej nr 2	Kraków	30-001	Kościuszki	8	-

6.4 Widoki - wersja fizyczna

Poniżej przedstawiono przykładowe zastosowanie oraz implementację widoku.

1 • `SELECT * FROM baza.przeglądanie_produkow_klient;`

	Nazwa_gry	id_gry	rodzaj	cena	ocena	ilosc_egzemplarzy	nazwa_kategorii
▶	Zabili go i ucieł	111	cyfrowa	80	NULL	2	Akcja
	Szybcy i wścieki 52	112	Pudełkowa	90	NULL	0	Akcja
	Ghost Division 2	114	cyfrowa	75	NULL	1	Strategia

6.4.1 Implementacja widoku - kod napisany ręcznie

```
1 • CREATE OR REPLACE VIEW przegladanie_produkow_klient AS
2 SELECT
3     gra.Nazwa_gry, gra.id_gry, gra.rodzaj, gra.cena,
4     gra.ocena, gra.ilosc_egzemplarzy, kategoria.nazwa_kategorii
5 FROM
6     gra LEFT JOIN kategoria ON gra.id_kategorii = kategoria.id_kategorii
7
```

6.5 Prawa użytkowników w bazie danych - wersja opisowa

Opisywane funkcjonalności będą możliwe do realizacji pośrednio, poprzez wywołanie właściwej procedury. Oprócz wymienionych poniżej, każdy z użytkowników posiada możliwość odczytu wszystkich danych dotyczących swojej encji oraz innych wybranych informacji dostępnych poprzez dedykowane widoki opisywane w poprzednim punkcie.

6.5.1 Klient

- usuwanie i modyfikacja danych z tabeli "klient" oraz powiązanej tabeli "adres"
- inicjalizacja i usuwanie encji "zamówienie"
- inicjalizacja danych z tabeli "platnosc", "adres" i "wyslka"

6.5.2 Pracownik sklepu

- modyfikacja wybranych danych z tabeli "wyslka" (data_wyslki)
- modyfikacja wybranych danych z tabeli "zamówienie" (data_realizacji)
- inicjalizacja, usuwanie, i modyfikacja danych z tabel "egzemplarz" oraz "gra"

6.5.3 Administrator

- inicjalizacja, usuwanie i modyfikacja danych z encji "pracownik" (wraz z przypisanym adresem), "administrator", "kategoria"
- ponadto wszystkie uprawnienia zwykłego pracownika sklepu oraz możliwość bezpośredniego poruszania się po bazie danych, bez wykorzystywania procedur

6.6 Prawa użytkowników w bazie danych - wersja fizyczna

Fizyczną implementację opisywanych uprawnień otrzymano poprzez utworzenie na serwerze MySQL odpowiednich ról, które będą przypisywane użytkownikom podczas tworzenia ich kont.

6.6.1 Implementacja dodania ograniczonych uprawnień - pracownik

Name: The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `AddUserGroup_Worker`()
2 BEGIN
3     CREATE ROLE 'pracownik';
4     GRANT EXECUTE ON PROCEDURE AddGameCopy TO 'pracownik';
5     GRANT EXECUTE ON PROCEDURE AddProduct TO 'pracownik';
6     GRANT EXECUTE ON PROCEDURE ChangeMagazynGames TO 'pracownik';
7     GRANT EXECUTE ON PROCEDURE DeleteGameCopy TO 'pracownik';
8     GRANT EXECUTE ON PROCEDURE SearchProductWorker TO 'pracownik';
9     GRANT EXECUTE ON PROCEDURE SendDate TO 'pracownik';
10    GRANT EXECUTE ON PROCEDURE SetDateOfRealization TO 'pracownik';
11    GRANT EXECUTE ON PROCEDURE ViewALLOrders TO 'pracownik';
12    GRANT EXECUTE ON PROCEDURE ViewProductsWorker TO 'pracownik';
13    SET GLOBAL activate_all_roles_on_login = ON;
14 END
```

6.6.2 Implementacja dodania pełnych uprawnień - administrator

Name: The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `AddUserGroup_Admin`()
2 BEGIN
3     CREATE ROLE 'admin';
4     GRANT ALL PRIVILEGES ON baza.* TO 'admin';
5     SET GLOBAL activate_all_roles_on_login = ON;
6 END
```

6.7 Procedury składowane - wersja opisowa

W naszej bazie możemy wyróżnić zestawione poniżej procedury i wyzwalacze. Argumenty procedur podano w nawiasach okrągłych (ich błędne podanie może skutkować niewykonaniem się procedury). W każdym z opisów ponadto zawarto główne operacje języka MySQL realizowane przez procedurę. Oprócz wymienionych poniżej, użytkownik root musi zadbać o wywołanie skryptów dodających role użytkowników AddUserGroup_Admin, AddUserGroup_Client i AddUserGroup_Worker przed udostępnieniem bazy pracownikom.

6.7.1 Procedury klienta (nie dostępne w aplikacji)

- ViewProductClient() - procedura przeznaczona dla klienta i mająca umożliwić mu alfabetyczne przeglądanie wszystkich produktów sklepu. Klient ma dostęp tylko do informacji o produkcie opisanych w widoku
 - SELECT wszystkich danych z widoku 'przeglądanie_produkow_klient'
- SearchProductClient(varchar) - procedura pozwalająca klientowi wyszukać produkty w sklepie (arg. - nazwa produktu lub nazwa kategorii). Zwracana wartość jest zależna od pomyślności wyszukania.
 - SELECT wybranego wiersza z widoku 'przeglądanie_produkow_klient'
- SubmitOrderClient(int, int, enum, enum) procedura przeznaczona dla klienta, mająca mu umożliwić złożenie zamówienia na wskazany produkt (arg. - id klienta, id gry, metoda płatności, metoda wysyłki)
 - SELECT id dostępnego egzemplarza z tabeli "egzemplarze"
 - opcjonalny SELECT id_adresu z tabeli "klient"
 - INSERT odpowiednich danych do tabel "zamowienie", "wyslka" i "platnosc" (przyjmuje się takie same id dodanych wierszy)
 - UPDATE ilości egzemplarzy w odpowiednim wierszu tabeli "gra" oraz id zamówienia przypisanego do egzemplarza

Może zwrócić błąd poprzez odpowiedni komunikat, jeśli wskazany produkt nie ma już dostępnych egzemplarzy.

- DeleteOrderClient(INT) - procedura umożliwiająca anulowanie nieopłaconego zamówienia (arg. - id zamówienia). Może zwrócić błąd poprzez odpowiedni komunikat jeśli zamówienie ma inny status niż 'nieoplacone'

- DELETE odpowiednich wierszy z tabel "zamowienie", "wysylka" i "platnosc"
- UPDATE ilości egzemplarzy w odpowiednim wierszu tabeli "gra" oraz usunięcie id zamówienia przypisanego do egzemplarza
- SubmitPurchase(int) - zmiana statusu w aktualnym zamówieniu z "nieopłacone" na "opłacone" (arg. - id zamówienia)
 - UPDATE statusu zamówienia w odpowiednim wierszu tabeli "zamowienia"
- ViewOrdersClient(int) - procedura przeznaczona dla klienta umożliwiająca mu dostęp do historii swoich zamówień (arg. - swoje id klienta)
 - SELECT danych z widoku 'zamowienie_produkтов' odpowiadających podanemu id klienta
- ChangeDataClient(int, varchar, varchar) - procedura umożliwiająca edycję wybranej danej dotyczącej konta (arg. id klienta, dana, nowa wartość). Procedura może zwrócić błąd np. w przypadku niewłaściwego wprowadzenia danych.
 - UPDATE wskazanego przez drugi argument pola w encji klient poprzez przypisanie mu wartości argumentu trzeciego
- DeleteAccountClient(int) - procedura umożliwiająca usunięcie konta klienta (arg. id klienta). Może zwrócić błąd i odpowiedni komunikat jeżeli klient ma przypisane co najmniej jedno niezrealizowane zamówienie. Procedura jest powiązana z wyzwalaczem AFTER DELETE usuwającym adres klienta.
 - UPDATE encji zamówienie poprzez ustawienie usuwanego id_klienta na NULL
 - usunięcie konta klienta z bazy danych wykorzystując DROP USER
 - usunięcie danych z encji klient odpowiadającej wskazanemu id klienta (DELETE FROM)

6.7.2 Procedury pracownika (dostępne w aplikacji)

- SearchProductWorker(vchar) - procedura pozwalająca pracownikowi wyszukać produkt w sklepie (arg. - nazwa produktu, nazwa kategorii lub id)
 - warunkowy SELECT wierszy z widoku 'przeglądanie_produkow_pracownik' (pasujących do podanego argumentu)
- ViewALLOrders() - procedura przeznaczona dla pracownika umożliwiająca mu przegląd historii dodanych zamówień wszystkich użytkowników
 - SELECT danych z widoku 'przegląd_zamowien_full'
- AddProduct(vchar, enum, double, int) procedura przeznaczona dla pracownika, służąca do dodania nowego produktu (arg. - nazwa gry, rodzaj, cena, id kategorii [może być NULL]). Może zwrócić błąd, jeżeli podane id kategorii nie ma swojego odpowiednika w bazie danych.
 - INSERT wskazanych przez dane wejściowe wartości do encji gra, ponadto wyzerowanie liczby egzemplarzy w wierszu.
- ChangeDataGames(int, vchar, vchar) - procedura pozwalająca pracownikowi na edycję wybranej danej dotyczącej gry (arg. - id gry, dana, nowa wartość). Wynik działania procedury zależy od poprawności danych wejściowych.
 - UPDATE wskazanego przez drugi argument pola w encji gra poprzez przypisanie mu wartości argumentu trzeciego
- AddGameCopy(int, int) - procedura dodająca egzemplarz danego produktu (arg. - id gry, id magazynu). Może zwrócić błąd i odpowiedni komunikat jeśli id gry jest niepoprawne. Powiązane z wyzwalaczem AFTER INSERT, zwiększającym ilość egzemplarzy w encji gra.
 - INSERT wartości wskazanych przez dane wejściowe do tabeli egzemplarze
- ChangeMagazynGames(int, int) - procedura pozwalająca pracownikowi na zmianę magazynu wybranego egzemplarza (arg. - id egzemplarza, nowe id magazynu)
 - UPDATE pola id_magazynu we wskazanej encji egzemplarz

- DeleteGameCopy(int) - usuwa egzemplarz o zadanym id, jeśli odpowiada ono któremuś z wierszy w encji egzemplarze (arg. - id egzemplarza). Zwraca informację o ilości usuniętych egzemplarzy (0 albo 1). Powiązane z wyzwalaczem AFTER_DELETE, zmniejszającym liczbę egzemplarzy w encji gra.
 - DELETE odpowiedniego wiersza z encji egzemplarze
- SetDateOfRealization(int, int, datetime) - ustawia datę realizacji wybranego zamówienia oraz informację o id pracownika potwierdzającego realizację, ponadto zmienia status zamówienia na 'zrealizowane' (arg. - id pracownika, id zamówienia, data)
 - UPDATE odpowiednich pól w zadanym wierszu tabeli zamówienia
- SendDate(int, datetime) - ustawia datę wysyłki wybranego zamówienia, o ile istnieje (arg. - id zamówienia/wysyłki, data)
 - UPDATE pola data_wysyłki we wskazanym wierszu tabeli wysyłka

6.7.3 Procedury administratora (dostępne w aplikacji)

- AddAdmin(varchar, varchar, varchar, varchar, varchar, varchar, char, varchar, int, int, varchar) - procedura dodająca konto nowego administratora (arg. imię, nazwisko, numer telefonu, stanowisko w firmie, email, miasto, kod pocztowy, ulica [opcj.], nr domu, nr mieszkania [opcj.], hasło do logowania)
 - INSERT podanych danych adresowych do przypisanej administratorowi (poprzez id adresu) tabeli adres
 - INSERT pozostałych danych do tabeli administrator
 - stworzenie nowego użytkownika za pomocą CREATE USER (login - adres email) oraz nadanie mu roli 'admin' za pomocą komendy GRANT
- SearchAdmin(varchar, varchar, varchar) - umożliwia wyszukanie konta administratora, jeśli pasuje do danych wejściowych (arg. - nazwisko, stanowisko, id_administratora). Może być wykorzystywane do podglądu własnego profilu. Przyjmuje się, że procedura jest wywoływana z jednym wybranym argumentem, a pozostałe wynoszą wtedy NULL.
 - SELECT pasujących danych z widoku 'przegląd_danych_admina'

- `ChangeDataAdmin(int, varchar, varchar)` - umożliwia edycję wybranych danych zadanej encji administratora, o ile taka istnieje (arg. - id administratora, dana, nowa wartość)
 - UPDATE wskazanego pola w tabeli administrator
- `DeleteAccountAdmin(int, int)` - usuwa konto administratora o zadanym id, jeśli jest poprawne (arg. - swoje id, id usuwanego admina)
 - usunięcie użytkownika za pomocą `DROP USER`, kierując się emailem zapisanym we wskazanym wierszu tabeli administrator
 - UPDATE pól 'id_administratora' w tabeli pracownik - przypisanie swojego id zamiast usuwanego użytkownika
 - usunięcie danych z tabeli administrator za pomocą `DELETE FROM`
- `AddWorker(...)` - procedura dodająca konto nowego pracownika. Identyczna w użyciu co `AddAdmin`, ale rozszerzona o pole 'id_administratora' (arg. - wszystkie dane jak w 'AddAdmin' + id administratora)
 - INSERT podanych danych adresowych do przypisanej pracownikowi (poprzez id adresu) tabeli adres
 - INSERT pozostałych danych do tabeli pracownik
 - stworzenie nowego użytkownika za pomocą `CREATE USER` (login - adres email) oraz nadanie mu roli 'worker' za pomocą komendy `GRANT`
- `SearchWorker(...)` - procedura pozwalająca administratorowi wyszukać konto pracownika. Identyczna w użyciu co `SearchAdmin`, ale rozszerzona o pole 'id_administratora' (arg. - wszystkie dane jak w 'SearchAdmin' + id administratora)
 - SELECT pasujących danych z widoku 'przegląd_danych_pracownika'
- `ChangeDataWorker(int, varchar, varchar)` - umożliwia edycję wybranej danej istniejącego pracownika o zadanym id (arg. - id pracownika, dana, nowa wartość)
 - UPDATE wskazanego pola w tabeli pracownik

- DeleteAccountWorker(int) - usuwa konto pracownika o zadany id, jeśli jest poprawne (arg. - id pracownika)
 - usunięcie użytkownika za pomocą DROP USER, kierując się email'em zapisanym we wskazanym wierszu tabeli pracownik
 - UPDATE pól 'id_pracownik' w tabeli zamówienia - przypisanie NULL zamiast id usuwanego użytkownika
 - usunięcie danych z tabeli pracownik za pomocą DELETE FROM
- AddCategory(varchar) - procedura dodająca nową kategorię (arg. - nazwa kategorii)
 - INSERT nazwy kategorii do tabeli kategoria
- SearchCategory(varchar) - umożliwia wyszukanie kategorii, jeśli jakaś pasuje do danych wejściowych (arg. - nazwa kategorii)
 - SELECT pasujących danych z tabeli kategoria
- DeleteCategory(int) - usuwa kategorię o zadany id, jeśli jest poprawne (arg. - id kategorii)
 - DELETE odpowiedniego wiersza z tabeli kategoria

Ponadto dostęp do wszystkich procedur pracownika.

6.8 Wyzwalacze

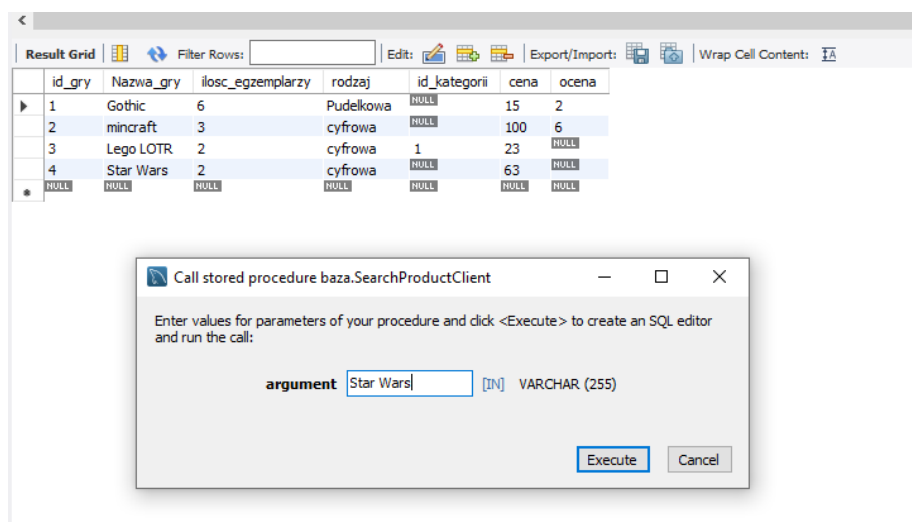
- egzemplarz_AFTER_INSERT - po dodaniu egzemplarza gry zwiększenie ich ilości w polu ilosc_egzemplarzy o +1 (operacja UPDATE)
- egzemplarz_AFTER_DELETE - po usunięciu egzemplarza gry zmniejszenie ich ilości w polu ilosc_egzemplarzy o -1 (operacja UPDATE)
- gra_BEFORE_UPDATE - ocena gry zostaje zaktualizowana, wliczając ocenę użytkownika do średniej pozostałych ocen (operacja UPDATE)
- klient_AFTER_DELETE - wyzwalacz automatycznie usuwający adres przypisany do klienta który usunął swoje konto (operacja DELETE FROM)

6.9 Procedury składowane i wyzwalacze - testy implementacji

Dokumentacja implementacji bazy danych nie zawiera dokładnych opisów działania widoków, wyzwalaczy, procedur czy innych elementów, ponieważ byłoby to powtórzeniem wcześniejszych opisów poszczególnych części składowych. W tej części większy nacisk został położony na zaprezentowanie funkcjonalności bazy oraz wyników przeprowadzonych testów poprawności. Poniżej zostały zaprezentowane najważniejsze z nich.

- SearchProductClient - jest zbliżona do pozostałych procedur służących do wyszukiwania, takich jak:
 - SearchCategory
 - SearchProductWorker
 - SearchAdmin
 - SearchWorker

Procedura wyszukuje produkty w bazie po nazwie gry lub kategorii. W przypadku sukcesu zwraca wyszukany element, w przypadku niepowodzenia nie zwraca nic. Wszystkie procedury z powyżej wypisanej grupy zostały poddane testom, które zakończyły się powodzeniem. Poniżej został przedstawiony efekt działania SearchProductClient.



Rysunek 1: Interfejs wyszukiwania

Result Grid					Filter Rows:	Export:	Wrap Cell Content:
Nazwa_gry	rodzaj	cena	ocena	nazwa_kategorii			
Star Wars	cyfrowa	63	100%	100%			

Result 1 x			
Output			
Action Output			
#	Time	Action	Message
1	19:38:48	call baza.SearchProductClient('Star Wars')	1 row(s) returned

Rysunek 2: Sukces wyszukiwania

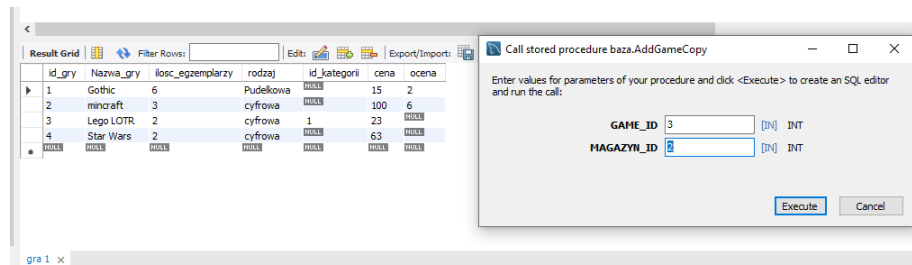
Result Grid					Filter Rows:	Export:	Wrap Cell Content:
Nazwa_gry	rodzaj	cena	ocena	nazwa_kategorii			

Result 1 x			
Output			
Action Output			
#	Time	Action	Message
1	19:38:48	call baza.SearchProductClient('Star Wars')	1 row(s) returned
2	19:39:23	call baza.SearchProductClient('L')	1 row(s) returned
3	19:39:39	call baza.SearchProductClient('min')	1 row(s) returned
4	19:55:28	call baza.SearchCategory('sandbox')	1 row(s) returned
5	20:30:39	call baza.SearchProductClient('Polo')	0 row(s) returned

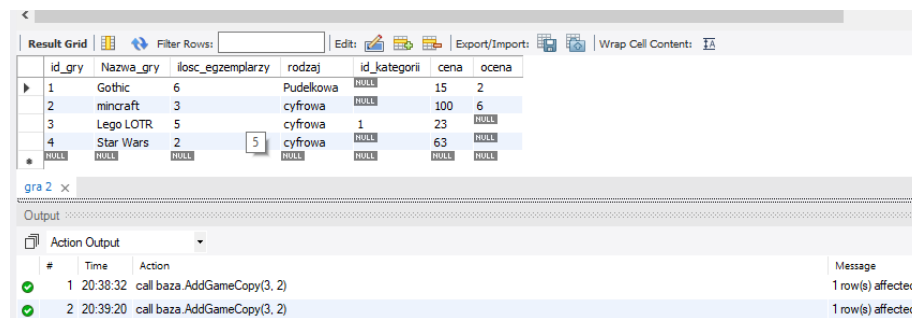
Rysunek 3: Porażka wyszukiwania - zwrócenie pustego wiersza

- AddProduct - zbliżona do pozostałych procedur w bazie służących do dodawania elementów encji:
 - AddCategory
 - AddClientAdress
 - AddProduct

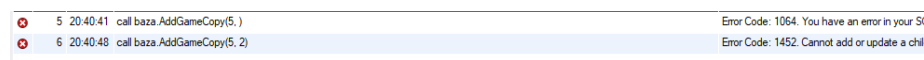
Procedura dodaje egzemplarze o danym id gry oraz id magazynu. Dodanie egzemplarza dodatkowo zmienia ilość egzemplarzy konkretnej gry. Próba dodania gry o id nieistniejącym w bazie kończy się niepowodzeniem i zostaje zwrócony komunikat o błędzie. Wszystkie procedury z powyżej wypisanej grupy zostały poddane testom, które zakończyły się powodzeniem.



Rysunek 4: Interfejs dodania wpisu



Rysunek 5: Sukces dodania wpisu



Rysunek 6: Nieudane dodanie wpisu - niepoprawne id

- AddClient - zbliżona do pozostałych procedur w bazie służących do dodawania elementów danej encji oraz tworzenia konta użytkownika:

- AddAdmin
- AddWorker

Procedura dodaje klienta o danym imieniu, nazwisku, numerze telefonu oraz adresie e-mail. Tworzy mu też konto, gdzie nazwą użytkownika jest adres e-mail. Wszystkie procedury z powyżej wypisanej grupy zostały poddane testom, które zakończyły się powodzeniem. Poniżej został przedstawiony efekt działania procedury.

User	From Host
aaa@aaa	localhost
aaa@wp.pl	localhost
aaqq@aaa	localhost
abc@wp.pl	localhost
admin	%
klient	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
pracownik	%
pracownik1	localhost
root	localhost

Rysunek 7: Dodanie użytkowników - stan przed

	id_klienta	imie	nazwisko	email	nr_kontakowy	id_adresu
▶	1	nowe_imie	nazwisko	abc@wp.pl	111111111	2
	2	a	b	aaa@aaa	111	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

Call stored procedure baza.AddClient

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

FIRST_NAME [IN] VARCHAR (30)
LAST_NAME [IN] VARCHAR (30)
PHONE_NUMBER [IN] VARCHAR (30)
EMAIL [IN] VARCHAR(30)

Execute Cancel

Rysunek 8: Interfejs dodania użytkownika

	id_klienta	imie	nazwisko	email	nr_kontakowy	id_adresu
▶	1	nowe_imie	nazwisko	abc@wp.pl	111111111	2
	2	a	b	aaa@aaa	111	NULL
	3	imie	nazwisko	email@example.com	123456789	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

Rysunek 9: Sukces dodania użytkownika - stan tabel

User Accounts	
User	From Host
aaa@aaa	localhost
aaa@wp.pl	localhost
aaqqq@aaa	localhost
abc@wp.pl	localhost
admin	%
email@example.com	localhost
klient	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
pracownik	%
pracownik1	localhost
root	localhost

Rysunek 10: Sukces dodania użytkownika - stan serwera

Ponieważ encja klienta początkowo nie jest połączona z pozostałymi elementami bazy danych, niepoprawne działanie procedury mogłoby wynikać jedynie w wypadku przekroczenia ilości znaków.

W celu zapewnienia większej przejrzystości, w finalnej wersji planuje się rozszerzyć zbiór argumentów procedury o hasło logowania.

- AddUserGroup_Client - zbliżona do pozostałych procedur w bazie służących do dodawania roli użytkownika:
 - AddUserGroup_Admin
 - AddUserGroup_Worker

Procedura dodaje rolę klienta, określając jego uprawnienia, nadawane mu automatycznie podczas tworzenia konta. Wszystkie procedury z powyżej wypisanej grupy zostały poddane testom, które zakończyły się powodzeniem.

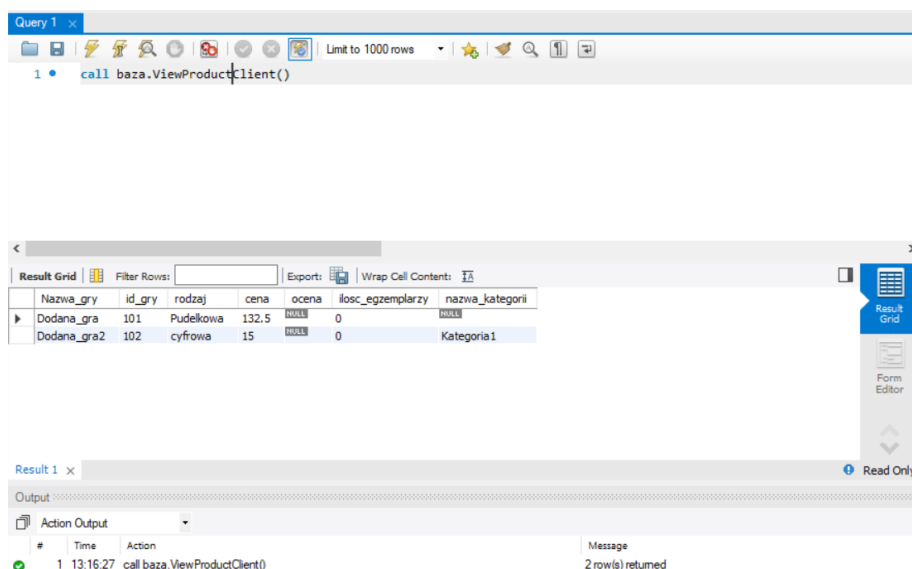
User	From Host
aaa@aaa	localhost
aaa@wp.pl	localhost
aaqq@aaa	localhost
abc@wp.pl	localhost
admin	%
email@example.com	localhost
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
pracownik	%
pracownik1	localhost
root	localhost

Rysunek 11: Role/użytkownicy przed dodaniem roli

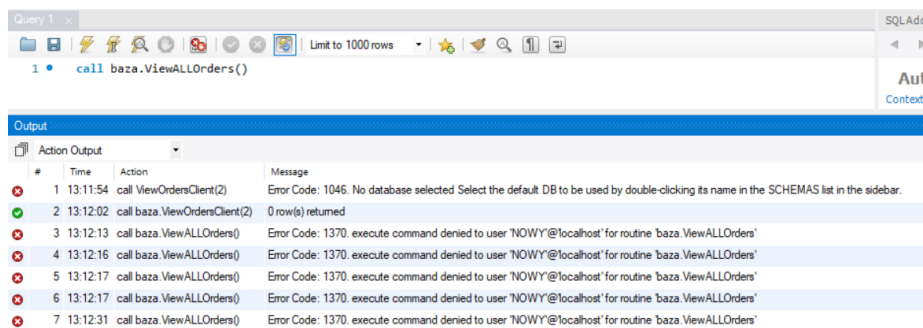
User	From Host
aaa@aaa	localhost
aaa@wp.pl	localhost
aaqq@aaa	localhost
abc@wp.pl	localhost
admin	%
email@example.com	localhost
klient	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
pracownik	%
pracownik1	localhost
root	localhost

Rysunek 12: Role/użytkownicy po dodaniu roli

Użytkownicy posiadający rolę klienta mają narzucone ograniczenia - nie mogą wykonywać pewnych procedur. Poniżej zaprezentowany został przykład procedury, którą klient może wykonać oraz takiej, której nie może.



Rysunek 13: Dozwolona procedura



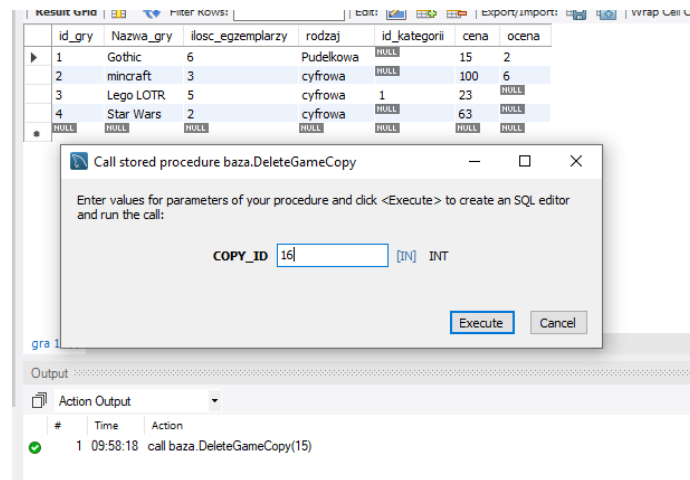
Rysunek 14: Niedozwolona procedura

- DeleteGameCopy - zbliżona do pozostałych procedur w bazie, służących do usuwania elementów danej encji:

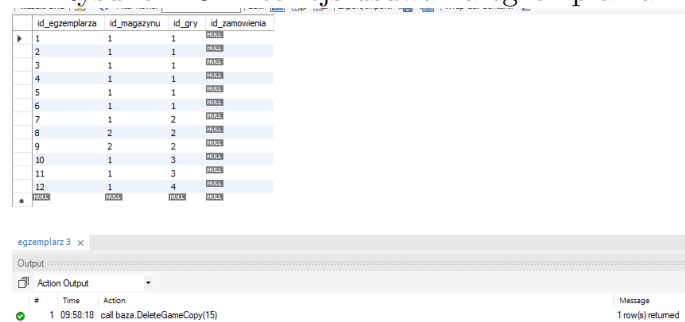
- DeleteCategory
- DeleteOrder
- DeleteOrderClient

Procedura usuwa egzemplarz o danym id_egzemplarza, aktualizując następnie ilość egzemplarzy gry (zastosowanie wyzwalacza) Próba usunięcia gry o id nieistniejącym w bazie kończy się zwróceniem komunikatu "rows 0".

Procedury z wyżej wypisanej grupy zostały poddane testom, które zakończyły się powodzeniem.



Rysunek 15: Interfejs usuwania egzemplarza



Rysunek 16: Stan tabeli egzemplarz po usunięciu

- DeleteAccountClient - zbliżona do pozostałych procedur w bazie służących do usuwania użytkowników o określonych uprawnieniach:

- DeleteAccountAdmin
- DeleteAccountWorker

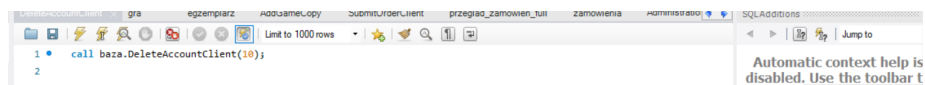
Procedura usuwa klienta o zadanym id_klienta. Próba usunięcia klienta o id nieistniejącym w bazie kończy się zwróceniem komunikatu "rows 0". Ponadto procedura zapobiega usunięciu klienta, który ma niezrealizowane zamówienia, wyświetlając wtedy stosowny komunikat. Wszystkie procedury z powyżej wypisanej grupy zostały poddane testom, które zakończyły się powodzeniem.

mysql.sys	localhost
nowy_klient	localhost
nowy_klient2	localhost
pracownik	%
root	localhost

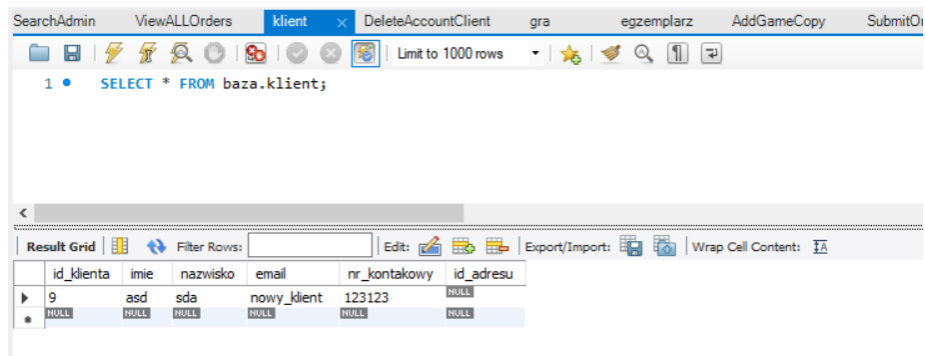
Rysunek 17: Lista użytkowników przed usunięciem użytkownika

id_klienta	imie	nazwisko	email	nr_kontakowy	id_adresu
9	asd	sda	nowy_klient	123123	NULL
10	asd	sda	nowy_klient2	123123	NULL
NULL	NULL	NULL	NULL	NULL	NULL

Rysunek 18: Tabela 'klient' przed usunięciem użytkownika



Rysunek 19: Wywołanie procedury usunięcia użytkownika

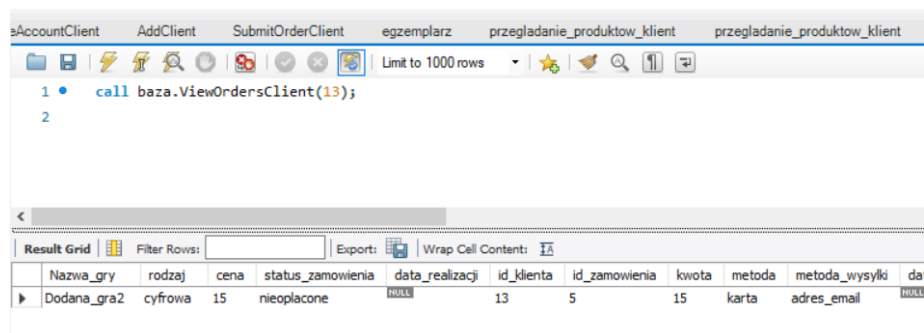


Rysunek 20: Tabela 'klient' po usunięciu użytkownika

```
mysql.session      localhost
mysql.sys          localhost
nowy_klient        localhost
pracownik          %
```

Rysunek 21: Stan serwera po usunięciu użytkownika

W przypadku gdy klient będzie miał niezrealizowane zamówienia:



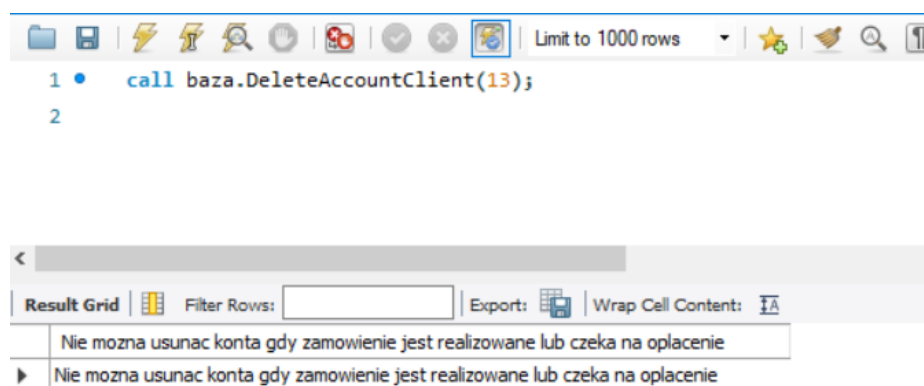
The screenshot shows a web application interface with a menu bar at the top containing: AccountClient, AddClient, SubmitOrderClient, egzemplarz, przeglądanie_produktow_klient, and przeglądanie_produktow_klient. Below the menu is a toolbar with various icons and a 'Limit to 1000 rows' dropdown. The main content area displays a code snippet:

```
1 • call baza.ViewOrdersClient(13);
2
```

Below the code is a 'Result Grid' table with the following columns: Nazwa_gry, rodzaj, cena, status_zamowienia, data_realizacji, id_klienta, id_zamowienia, kwota, metoda, metoda_wysylki, and data_realizacji. The table contains one row of data:

Nazwa_gry	rodzaj	cena	status_zamowienia	data_realizacji	id_klienta	id_zamowienia	kwota	metoda	metoda_wysylki	data_realizacji
Dodana_gra2	cyfrowa	15	nieoplacone	NULL	13	5	15	karta	adres_email	NULL

Rysunek 22: Blokada - klient złożył zamówienie



The screenshot shows a web application interface with a menu bar at the top containing: AccountClient, AddClient, SubmitOrderClient, egzemplarz, przeglądanie_produktow_klient, and przeglądanie_produktow_klient. Below the menu is a toolbar with various icons and a 'Limit to 1000 rows' dropdown. The main content area displays a code snippet:

```
1 • call baza.DeleteAccountClient(13);
2
```

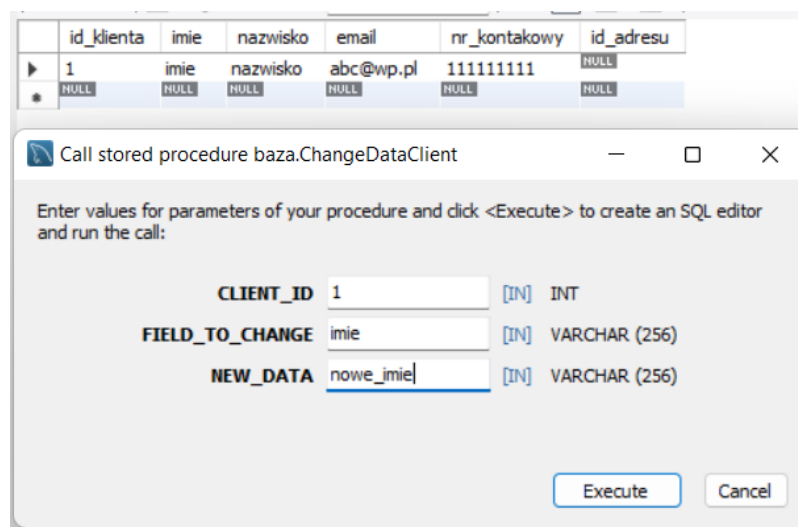
Below the code is a 'Result Grid' table with the following columns: Nazwa_gry, rodzaj, cena, status_zamowienia, data_realizacji, id_klienta, id_zamowienia, kwota, metoda, metoda_wysylki, and data_realizacji. The table contains one row of data:

Nazwa_gry	rodzaj	cena	status_zamowienia	data_realizacji	id_klienta	id_zamowienia	kwota	metoda	metoda_wysylki	data_realizacji
Nie mozna usunac konta gdy zamowienie jest realizowane lub czeka na oplacenie										

Rysunek 23: Komunikat o błędzie przy wywołaniu usuwania klienta

- ChangeDataClient zbliżona do pozostałych procedur w bazie, służących do zmiany elementów danej encji:
 - ChangeDataAdmin
 - ChangeDataGames
 - ChangeDataWorker
 - ChangeMagazynGames

Procedura aktualizuje wybrane pole wśród danych klienta o zadanym id_klienta. Próba zmiany danych nieistniejącego klienta lub próba zmiany nieistniejącej danej nie wywrą żadnego efektu. Wszystkie procedury z powyżej wypisanej grupy zostały poddane testom, które zakończyły się powodzeniem. Poniżej przedstawiony został efekt działania procedury.



Rysunek 24: Tabela przed zmianą danych oraz interfejs wywołania procedury zmiany

	id_klienta	imie	nazwisko	email	nr_kontakowy	id_adresu
▶	1	nowe_imie	nazwisko	abc@wp.pl	111111111	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

Rysunek 25: Tabela po zmianie danych

- SendDate - zbliżona do procedury SetDateOfRealization. Procedura ustawia podaną datę dla danej wysyłki. Nie jest możliwe wprowadzenie daty w błędnym formacie, jest natomiast możliwe wybranie id wysyłki, która nie istnieje - procedura wtedy nie będzie miała żadnego efektu. W aplikacji dostępowej procedura domyślnie pobiera jako argument moment jej wywołania. Poniżej przedstawiony został efekt działania procedury.

Call stored procedure baza.SendDate

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

SHIPMENT_ID 1 [IN] INT

DATE_SEND 2022-05-24 10:00:00 [IN] datetime

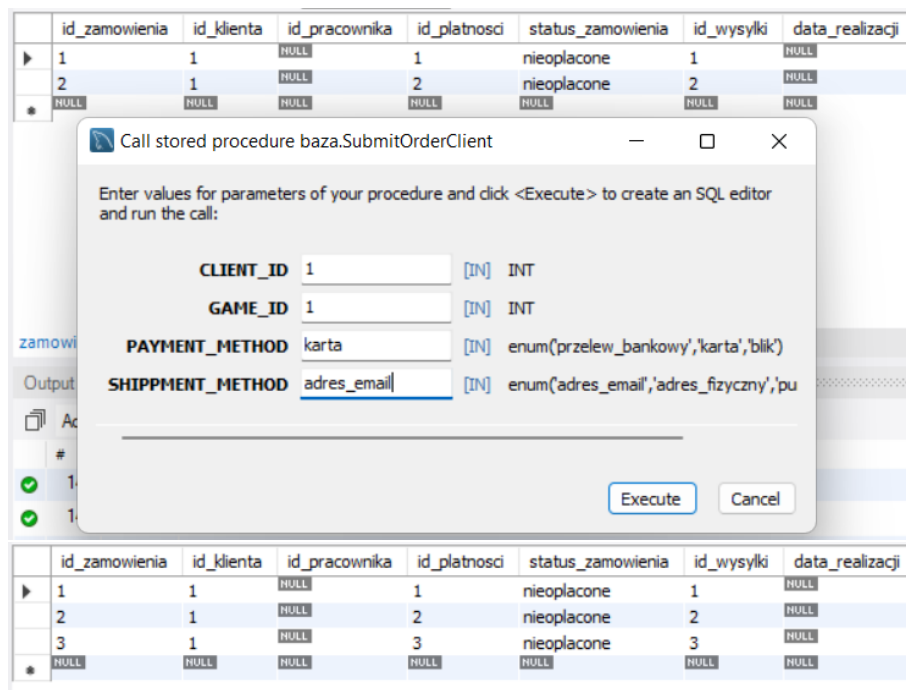
Execute Cancel

Rysunek 26: Tabela 'wysylka' przed ustawieniem daty

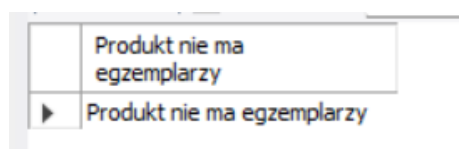
	id_wysylki	metoda_wysylki	data_wysylki	id_adresu
▶	1	adres_fizyczny	2022-05-24 10:00:00	2
★	NULL	NULL	NULL	NULL

Rysunek 27: Tabela 'wysylka' po ustawieniu daty

- SubmitOrderClient - procedura tworzy zamówienie na dany produkt. Ponadto tworzona jest również płatność z daną metodą płatności i wysyłka z odpowiednią metodą wysyłki. Jeżeli wybrany produkt nie ma egzemplarzy, procedura wyświetli stosowny komunikat. Poniżej przedstawiony został efekt działania procedury.

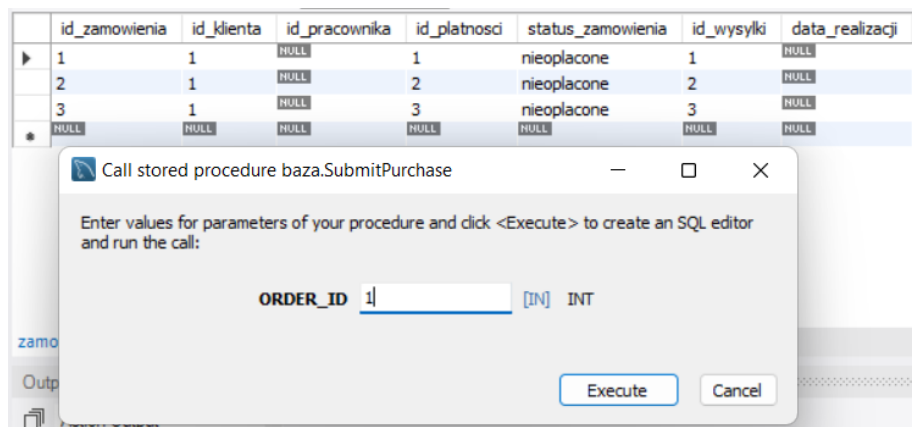


Rysunek 28: Interfejs oraz zmiany wywołane zamówieniem produktu



Rysunek 29: Komunikat o błędzie w przypadku braku egzemplarzy

- SubmitPurchase - procedura zmienia status danego zamówienia na "opłacone".
Poniżej przedstawiony został efekt działania procedury.



Rysunek 30: Stan zamówienia przed potwierdzeniem opłaty oraz interfejs wywołania

	id_zamowienia	id_klienta	id_pracownika	id_platnosc	status_zamowienia	id_wysylki	data_realizacji
▶	1	1	NULL	1	oplacone	1	NULL
	2	1	NULL	2	nieoplacone	2	NULL
	3	1	NULL	3	nieoplacone	3	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Rysunek 31: Stan tabeli 'zamowienia' po potwierdzeniu opłaty

- ViewProductClient - zbliżona do pozostałych procedur w bazie, służących do wywoływania widoków:
 - ViewAdmins
 - ViewALLOrders
 - ViewOrdersClient
 - ViewProductsWorker
 - ViewWorkers

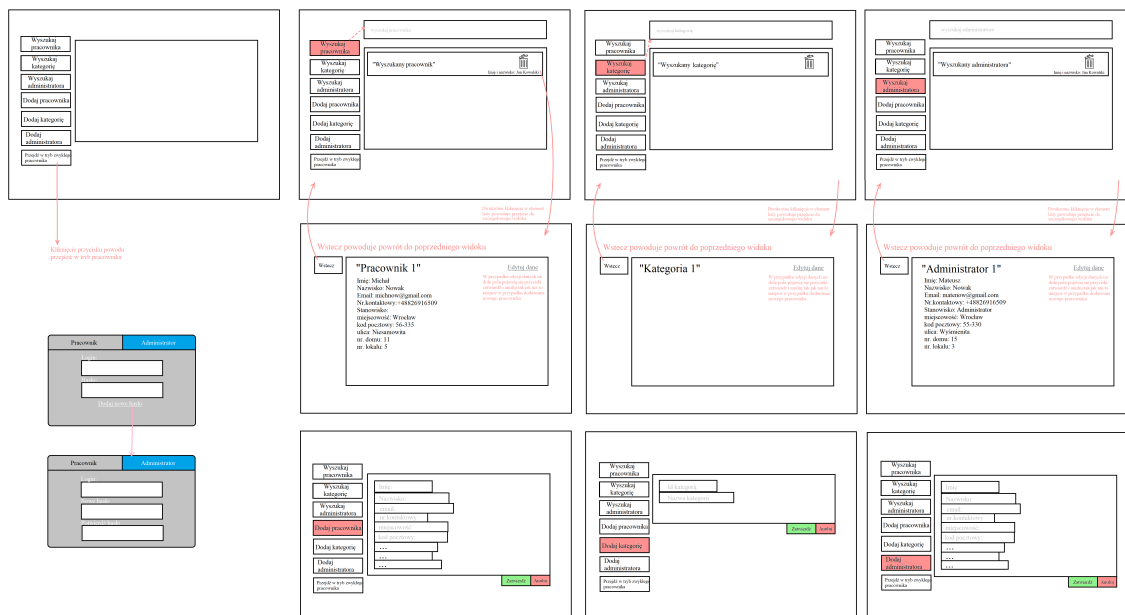
Procedura wywołuje widok dający dostęp do informacji na temat gier obecnych w bazie danych. Wszystkie procedury z powyżej wypisanej grupy zostały poddane testom, które zakończyły się powodzeniem. Poniżej przedstawiony został efekt działania procedury.

	Nazwa_gry	id_gry	rodzaj	cena	ocena	ilosc_egzemplarzy	nazwa_kategorii
▶	gra1	1	cyfrowa	39.99	HULL	4	kategoria1
	innna gra	2	cyfrowa	12.12	HULL	0	kategoria1

Rysunek 32: Przykładowe wywołanie widoku

Jeżeli pamiętano o wprowadzeniu do bazy wszystkich widoków, to nie przewiduje się żadnych warunków w których procedury z rodziny View mogłyby zwrócić błąd.

- Wyzwalacze - wyzwalacze zaimplementowane w bazie pozwalają na bezproblemowe wykonywanie wyżej wymienionych operacji, a testy poprawności ich działania wbudowane są w testy wywołania procedur. Przykładowo wyzwalacz egzemplarz_AFTER_INSERT pilnuje, by po dodaniu egzemplarza gry zwiększyła się ilość egzemplarzy w encji gra i jest uruchamiany automatycznie po poprawnym zakończeniu procedury dodającej grę.



Rysunek 34: Makieta aplikacji dostępowej w trybie administratora

W swojej pierwotnej wersji model zakładał aplikację z GUI jednakże na etapie implementacji pojawiły się problemy, które zmusiły nas finalnie do zmiany koncepcji i stworzenia aplikacji konsolowej. Wygląd aplikacji oraz niektóre z funkcjonalności zostaną przedstawione w dalszej części dokumentacji w punktach 7.3, 7.4.

7.2 Bezpieczeństwo aplikacji oraz bazy danych

W projekcie założono, że największe zagrożenia w kwestii bezpieczeństwa bazy danych oraz aplikacji to wewnętrzne ryzyka, które najczęściej mogłyby wynikać z niekompetencji bądź pomyłek pracowników.

Aplikacja dostępowa była projektowana w możliwie przejrzysty i intuicyjny sposób. Miało to zapewnić dobre rozumienie sposobu działania aplikacji nawet w przypadku nowych pracowników, którzy jeszcze nie są tak dobrze zaznajomieni z aplikacją.






Głównym elementem zabezpieczającym aplikację jest indywidualny login (dla uproszczenia przyjęty jako adres email) oraz hasło, które posiada każdy z pracowników. Login oraz ustawiane przez administratora hasło jest wymagane przez aplikację za każdym razem kiedy pracownik będzie chciał się do niej zalogować. Na etapie modelowania przyjmuje się, że loginy oraz hasła w postaci zaszyfrowanej jednostronnie będą przechowywane w plikach serwera MySQL.

Same dane w tabelach będą poddawane procesowi wstępnej walidacji, co ograniczy możliwość wprowadzenia niepożądanych i błędnych informacji.

Do realizacji zabezpieczenia informacji wykorzystano także podział na odpowiednie procedury, wyzwalacze oraz widoki. Użytkownicy aplikacji nie będą mogli wychodzić poza określone ramy wynikające z ich uprawnień i wykonywać niepożądanych operacji, mogących zaszkodzić bazie danych.

7.3 Instrukcja obsługi i instalacji aplikacji

W celu rozpoczęcia pracy z aplikacją należy uruchomić ją na urządzeniu z wgraną bazą danych sklepu. Jeżeli zachowano domyślne wartości proponowane przez program MySQL Workbench, to w celu połączenia wystarczy uruchomić plik 'B1.exe'.

	B1	30.05.2022 19:46	Folder plików	
	packages	30.05.2022 10:16	Folder plików	
	B1.sln	30.05.2022 09:56	Visual Studio Solu...	2 KB
	Backup	06.06.2022 21:12	Dokument tekstowy	20 KB
	B1	30.05.2022 19:44	Aplikacja	9 KB

Rysunek 35: Sposób uruchomienia aplikacji dostępowej

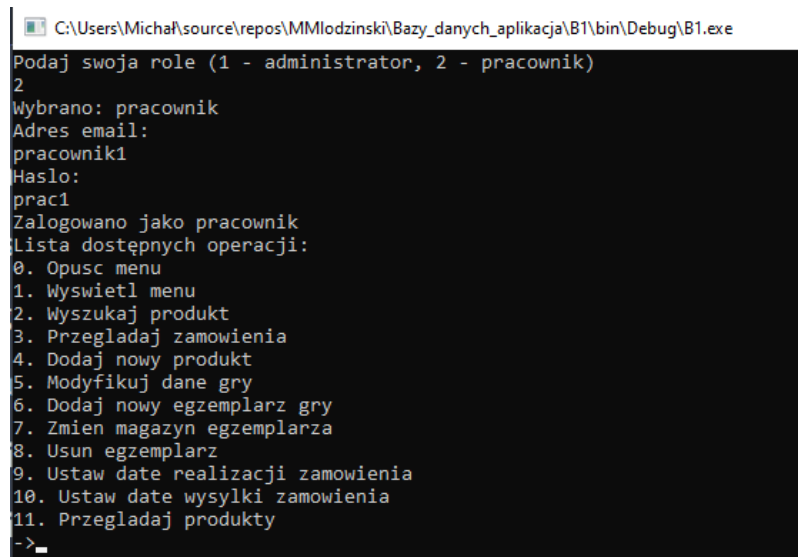
Aplikacja dostępowa umożliwia dwa tryby pracy z bazą danych: jako zwykły pracownik oraz jako administrator. Po uruchomieniu aplikacji na ekranie jako pierwsze pokazuje się okno terminala z prośbą wyboru trybu pracy - 1 lub 2.

```
C:\Users\Micha\Source\Repos\MMlodzinski\Bazy_danych_aplikacja\B1\bin\Debug\B1.exe
Podaj swoja role (1 - administrator, 2 - pracownik)
5
Blad! Niepoprawny wybor
Podaj swoja role (1 - administrator, 2 - pracownik)
2
Wybrano: pracownik
Adres email:
_
```

Rysunek 36: Wybór trybu pracy

Należy wybrać jedną z proponowanych opcji. Wpisanie przez użytkownika wartości spoza zakresu dostępnych spowoduje wyświetlenie komunikatu o wprowadzeniu nieobsługiwanej wartości.

Po poprawnym wyborze trybu pracy użytkownik zostanie przekierowany do widoku logowania. Użytkownik ma do wpisania kolejno login oraz hasło przypisane do swojego konta. Aplikacja umożliwi przejście dalej dopiero po poprawnym wprowadzeniu danych.



```
C:\Users\Michał\source\repos\MMłodzinski\Bazy_danych_aplikacja\B1\bin\Debug\B1.exe
Podaj swoją rolę (1 - administrator, 2 - pracownik)
2
Wybrano: pracownik
Adres email:
pracownik1
Hasło:
prac1
Zalogowano jako pracownik
Lista dostępnych operacji:
0. Opuść menu
1. Wyświetl menu
2. Wyszukaj produkt
3. Przeglądaj zamówienia
4. Dodaj nowy produkt
5. Modyfikuj dane gry
6. Dodaj nowy egzemplarz gry
7. Zmień magazyn egzemplarza
8. Usun egzemplarz
9. Ustaw datę realizacji zamówienia
10. Ustaw datę wysyłki zamówienia
11. Przeglądaj produkty
->
```

Rysunek 37: Wprowadzenie danych logowania oraz wyświetlenie menu pracownika

W programie została uwzględniona sytuacja wyjątkowa, w której zwykły pracownik próbuje zalogować się jako administrator. Spowoduje to wyświetlenie błędu o próbie korzystania z zasobów, do których nie ma się uprawnień i uniemożliwi dalszy dostęp. Podobne działania występują również w przypadku niepoprawnego wywołania procedury z innej przyczyny.

Po poprawnym zalogowaniu na ekranie powinno wyświetlić się menu dostępnych operacji. Wygląd menu pracownika został przedstawiony na rysunku 38. natomiast wygląd menu administratora został przedstawiony poniżej.

```

Zalogowano jako administrator
Lista dostępnych operacji:
0. Opusc menu
1. Wyświetl menu
2. Wyszukaj produkt
3. Przeglądaj zamówienia
4. Dodaj nowy produkt
5. Modyfikuj dane gry
6. Dodaj nowy egzemplarz gry
7. Zmien magazyn egzemplarza
8. Usun egzemplarz
9. Ustaw date realizacji zamówienia
10. Ustaw date wysyłki zamówienia
11. Przeglądaj produkty
12. Przeglądaj adminow
13. Przeglądaj pracowników
14. Wyszukaj admina
15. Zmien dane admina
16. Dodaj admina
17. Wyszukaj pracownika
18. Zmien dane pracownika
19. Usun konto pracownika
20. Usun konto admina
21. Dodaj pracownika
22. Wyszukaj kategorie
23. Dodaj kategorie
24. Usun kategorie
->

```

Rysunek 38: Wyświetlenie menu administratora

Aplikacja będzie oczekiwać wprowadzenia liczby odpowiadającej numerowi z listy operacji. Wybranie prawidłowej opcji z menu spowoduje rozpoczęcie wybranej operacji. Widok wykonania przykładowej procedury został przedstawiony poniżej.

C:\Users\Michał\source\repos\MM\lodziński\Bazy_danych_aplikacja\B1\bin\Debug\B1.exe

id_gry	Nazwa_gry	ilosc_egzemplarzy	cena	rodzaj	nazwa_kategorii	id_kategorii	id_egzemplarza	id_magazynu
1	Minecraft	0	120	cyfrowa	Strzelanka	1	null	null
2	Lego SW	0	100	Pudełkowa	Przygodowa	4	null	null
3	Gothic1	1	20	cyfrowa	Przygodowa	4	5	2
4	Gothic3	3	70	cyfrowa	2	2	2	1
4	Gothic3	3	70	cyfrowa	2	2	3	1
4	Gothic3	3	70	cyfrowa	2	2	4	1
5	Gothic2	1	50	cyfrowa	Przygodowa	4	6	2
6	Crysis1	1	230	cyfrowa	Strzelanka	1	1	1
7	Gothic4	0	105	cyfrowa	Akcji	6	null	null
8	Crysis2	0	120	cyfrowa	Przygodowa	4	null	null

Naciśnij enter, aby kontynuować.

Rysunek 39: Widok po wybraniu opcji przeglądu produktów z menu pracownika

Reszta operacji dostępnych w aplikacji działa analogicznie. Dodatkowo po każdym wykonaniu dowolnej operacji z listy, aplikacja wraca do menu głównego i możliwy jest dalszy wybór dostępnych opcji. Aby zakończyć pracę aplikacji użytkownik musi wprowadzić "0" z poziomu menu głównego. Spowoduje to natychmiastowe zamknięcie aplikacji.

7.4 Testy poprawności działania - od strony użytkownika

- Dodawanie elementów

Jedną z grup funkcji jest dodawanie nowych elementów do bazy danych. Przykładem jest funkcja dodawania produktu, która może być wywołana w aplikacji w następujący sposób:

```
Podaj wartosci poszczegolnych atrybow produktu, ktory chcesz dodac do bazy:  
Nazwa:Forza Motorsport 8  
Rodzaj('Pudelkowa', 'cyfrowa'):Pudelkowa  
Cena:150  
Id kategorii(puste/niepoprawne -> NULL):1
```

Rysunek 40: Aplikacja - dodawanie nowego produktu

Efektom działania tego wywołania jest nowy wpis w bazie danych:

	id_gry	Nazwa_gry	ilosc_egzemplarzy	rodzaj	id_kategorii	cena	ocena	liczba_ocen
▶	1	Forza Motorsport 5	0	Pudelkowa	1	80	NULL	0
	2	Forza Motorsport 7	0	cyfrowa	1	120	NULL	0
	3	Forza Motorsport 8	0	Pudelkowa	1	150	NULL	0
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Rysunek 41: Test - Nowy produkt w bazie danych

- Modyfikacja elementów

Inną grupą funkcji są funkcje modyfikujące dane. Przykładem jest funkcja odpowiedzialna za modyfikowanie istniejącego produktu, która może zostać wywołana w następujący sposób w aplikacji:

```
Podaj wartosc id produktu, nazwe pola ktore chcesz zmienic oraz nowa wartosc:  
Id_gry:3  
Nazwa pola do zmodyfikowania:rodzaj  
Nowa wartosc:cyfrowa  
  
Nacisnij enter, aby kontynuowac.
```

Rysunek 42: Aplikacja - Modyfikacja produktu

Efektom działania tego wywołania jest zmieniony wpis w bazie danych.

	id_gry	Nazwa_gry	ilosc_egzemplarzy	rodzaj	id_kategorii	cena	ocena	liczba_ocen
▶	1	Forza Motorsport 5	0	Pudełkowa	1	80	NULL	0
	2	Forza Motorsport 7	0	cyfrowa	1	120	NULL	0
	3	Forza Motorsport 8	0	cyfrowa	1	150	NULL	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Rysunek 43: Test - Zmodyfikowany produkt w bazie danych

- Usuwanie elementów

Kolejną grupą funkcji są funkcje usuwające elementy z bazy danych, dla przykładu funkcja usuwająca administratora.

	id_administratora	imie	nazwisko	email	nr_kontakowy	stanowisko	id_adresu
▶	1	Jan	Kowalski	jkowalski@wp.pl	+48666111999	Główny magazynier	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Rysunek 44: Test - stan przed usunięciem administratora

Wywołanie funkcji usuwającej administratora jest następujące:

```

Podaj swoje id:
1
Podaj id admina, którego chcesz usunąć:
1

Nacisnij enter, aby kontynuowac.

```

Rysunek 45: Aplikacja - Usuwanie administratora

Poniżej zaprezentowano efekt działania funkcji:

	id_administratora	imie	nazwisko	email	nr_kontakowy	stanowisko	id_adresu
▶*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Rysunek 46: Test - stan po usunięciu administratora

- Przeszukiwanie

Następną grupą funkcji są funkcje przeszukujące bazę danych, dla przykładu funkcja przeszukująca produkty obecne w bazie danych. Jej działanie wygląda następująco:

```
Wpisz nazwe produktu do wyszukania:
7
id_gry|Nazwa_gry      |ilosc_egzemplarzy|cena|rodzaj  |nazwa_kategorii|id_kategorii|id_egzemplarza|id_magazynu|
-----|-----|-----|-----|-----|-----|-----|-----|
2  |Forza Motorsport 7|0                |120 |cyfrowa  |wyścigowe      |1          |null          |null        |
4  |Gran Turismo 7   |0                |320 |Pudełkowa|wyścigowe      |1          |null          |null        |
-----|-----|-----|-----|-----|-----|-----|-----|
Nacisnij enter, aby kontynuować.
```

Rysunek 47: Aplikacja - Wyszukiwanie produktów w bazie danych

- Przeglądanie danych

Istnieje też grupa funkcji pozwalająca na przeglądanie wszystkich danych z konkretnej kategorii, na przykład produktów. Poniżej zaprezentowano działanie tej funkcji.

```
id_gry|Nazwa_gry      |ilosc_egzemplarzy|cena|rodzaj  |nazwa_kategorii|id_kategorii|id_egzemplarza|id_magazynu|
-----|-----|-----|-----|-----|-----|-----|-----|
1  |Forza Motorsport 5|0                |80  |Pudełkowa|wyścigowe      |1          |null          |null        |
2  |Forza Motorsport 7|0                |120 |cyfrowa  |wyścigowe      |1          |null          |null        |
3  |Forza Motorsport 8|0                |150 |cyfrowa  |wyścigowe      |1          |null          |null        |
4  |Gran Turismo 7   |0                |320 |Pudełkowa|wyścigowe      |1          |null          |null        |
-----|-----|-----|-----|-----|-----|-----|-----|
Nacisnij enter, aby kontynuować.
```

Rysunek 48: Aplikacja - Przeglądanie produktów w bazie danych

7.5 Testy poprawności działania - realizacja w kodzie źródłowym

W aplikacji zadbano o obsługę wyjątków zarówno wynikających ze złego formatu danych wprowadzanych przez użytkownika jak i tych związanych z niepoprawnie wykonującymi się operacjami. Dla pierwszej z tych kategorii zostały stworzone dedykowane funkcje walidacyjne sprawdzające np. czy liczba jest dodatnia, czy jest całkowita, czy format wprowadzonego numeru telefonu jest poprawny, itd. Kilka przykładowych sytuacji, w których wystąpi wyjątek, zostało przedstawionych poniżej.

```
Id:d
Indeks musi byc liczba dodatnia
->-5
Indeks musi byc liczba dodatnia
->■
```

Rysunek 49: Sprawdzenie czy wprowadzona wartość jest liczbą dodatnią

```
Nr_kontaktowy:
aaa
Nr. telefonu został źle wprowadzony. Akceptowany format '+48xxxxxxxxx'.
->123456789
Nr. telefonu został źle wprowadzony. Akceptowany format '+48xxxxxxxxx'.
->■
```

Rysunek 50: Zły format wprowadzonego numeru telefonu

```
Rodzaj('Pudelkowa', 'cyfrowa'):fizyczna
Prosze podac odpowiedni rodzaj gry (cyfrowa lub Pudelkowa).
->■
```

Rysunek 51: Wprowadzona wartość nie zawiera się w dozwolonych zakresie

W przypadku drugiej grupy wystąpienie wyjątku jest sprawdzane automatycznie przez serwer MySQL wewnątrz każdej funkcji wykonującej operację na bazie danych. Należy wyróżnić dwie sytuacje będące powodem powstawania wyjątków:

- próba uzyskania dostępu do zasobów, do których użytkownik nie ma uprawnień
- powstanie błędu w trakcie wykonywania danej procedury, np. próba usunięcia elementu o nieistniejącym id

```
C:\Users\Michał\source\repos\MMlodzinski\Bazy_danych_aplikacja\B1\bin\Debug\B1.exe
Podaj nazwę kategorii:
Przygodowa
execute command denied to user 'pracownik1'@'localhost' for routine 'baza.SearchCategory'
Błąd dostępu lub wywołania procedury
Zalogowano jako administrator
Lista dostępnych operacji:
0. Opusc menu
1. Wyświetl menu
2. Wyszukaj produkt
3. Przeglądaj zamówienia
4. Dodaj nowy produkt
5. Modyfikuj dane gry
```

Rysunek 52: Próba wykonania procedury, do której pracownik nie ma uprawnień

Dodatkowo aplikacja została wzbogacona o testy jednostkowe, które umożliwiają zautomatyzowane sprawdzanie poprawności procedur. Liczbę testów jednostkowych oraz ich poziom skomplikowania można zwiększyć w przypadku dalszego rozwijania aplikacji, co poskutkowałoby usprawnieniem bezpieczeństwa i niezawodności działania aplikacji. Przykładowy test jednostkowy został przedstawiony poniżej.

```
✓ AddProductTest 446 ms
Test Detail Summary
✓ AddProductTest
Source: UserContactTests.cs line 20
Duration: 446 ms
Standard Output:
id_gry|Nazwa_gry|ilosc_egzemplarzy|cena|rodzaj |nazwa_kategorii|id_kategorii
-----
1 |Minecraft|0 |120 |cyfrowa |Strzelanka |1
-----
2 |Lego SW |0 |100 |Pudelkowa|Przygodowa |4
-----
3 |Gothic1 |1 |20 |cyfrowa |Przygodowa |4
-----
4 |Gothic3 |3 |70 |cyfrowa |2 |2
-----
```

Rysunek 53: Test jednostkowy funkcji AddProduct()

8 Wykorzystane materiały

- dr. inż. Roman Ptak, prezentacje dydaktyczne dedykowane dla kursu "Bazy Danych", 2022
- Oracle, MySQL Workbench Reference Manual, <https://dev.mysql.com/doc/workbench/en/> (dostęp 21.06.2022)
- Oracle, MySQL Connector/NET Developer Guide, <https://dev.mysql.com/doc/connector-net/en/> (dostęp 21.06.2022)

9 Spis rysunków z etapu testowania procedur i aplikacji

Spis rysunków

1	Interfejs wyszukiwania	25
2	Sukces wyszukiwania	26
3	Porażka wyszukiwania - zwrócenie pustego wiersza	26
4	Interfejs dodania wpisu	27
5	Sukces dodania wpisu	27
6	Nieudane dodanie wpisu - niepoprawne id	27
7	Dodanie użytkowników - stan przed	28
8	Interfejs dodania użytkownika	28
9	Sukces dodania użytkownika - stan tabel	28
10	Sukces dodania użytkownika - stan serwera	29
11	Role/użytkownicy przed dodaniem roli	30
12	Role/użytkownicy po dodaniu roli	30
13	Dozwolona procedura	31
14	Niedozwolona procedura	31
15	Interfejs usuwania egzemplarza	32
16	Stan tabeli egzemplarz po usunięciu	32
17	Lista użytkowników przed usunięciem użytkownika	33
18	Tabela 'klient' przed usunięciem użytkownika	33
19	Wywołanie procedury usunięcia użytkownika	34
20	Tabela 'klient' po usunięciu użytkownika	34
21	Stan serwera po usunięciu użytkownika	34
22	Blokada - klient złożył zamówienie	35
23	Komunikat o błędzie przy wywołaniu usuwania klienta	35

24	Tabela przed zmianą danych oraz interfejs wywołania procedury zmiany	36
25	Tabela po zmianie danych	36
26	Tabela 'wysylka' przed ustawieniem daty	37
27	Tabela 'wysylka' po ustawieniu daty	37
28	Interfejs oraz zmiany wywołane zamówieniem produktu	38
29	Komunikat o błędzie w przypadku braku egzemplarzy	38
30	Stan zamówienia przed potwierdzeniem opłaty oraz interfejs wywołania	39
31	Stan tabeli 'zamowienia' po potwierdzeniu opłaty	39
32	Przykładowe wywołanie widoku	40
33	Makieta aplikacji dostępowej w trybie zwykłego pracownika	41
34	Makieta aplikacji dostępowej w trybie administratora	42
35	Sposób uruchomienia aplikacji dostępowej	44
36	Wybór trybu pracy	44
37	Wprowadzenie danych logowania oraz wyświetlenie menu pracownika	45
38	Wyświetlenie menu administratora	46
39	Widok po wybraniu opcji przeglądu produktów z menu pracownika .	46
40	Aplikacja - dodawanie nowego produktu	47
41	Test - Nowy produkt w bazie danych	47
42	Aplikacja - Modyfikacja produktu	47
43	Test - Zmodyfikowany produkt w bazie danych	48
44	Test - stan przed usunięciem administratora	48
45	Aplikacja - Usuwanie administratora	48
46	Test - stan po usunięciu administratora	48
47	Aplikacja - Wyszukiwanie produktów w bazie danych	49
48	Aplikacja - Przeglądanie produktów w bazie danych	49
49	Sprawdzenie czy wprowadzona wartość jest liczbą dodatnią	50
50	Zły format wprowadzonego numeru telefonu	50
51	Wprowadzona wartość nie zawiera się w dozwolonych zakresie	50
52	Próba wykonania procedury, do której pracownik nie ma uprawnień .	51
53	Test jednostkowy funkcji AddProduct()	51

10 Spis modeli i opisów tabel

Spis tabel

1	klient	9
2	zamowienie	9
3	pracownik	9
4	administrator	9
5	gra	10
6	platnosc	10
7	wysylka	10
8	kategoria	10
9	egzemplarz	10
10	Adres	10
11	klient - opis	11
12	pracownik - opis	11
13	administrator - opis	11
14	Zamówienie - opis	11
15	gra - opis	12
16	platnosc - opis	12
17	Wysyłka - opis	12
18	kategoria - opis	12
19	egzemplarz - opis	12
20	Adres - opis	13

11 Podsumowanie

Zestawiając wiedzę własną z poznanymi w trakcie semestru informacjami dotyczącymi modelowania i obsługi relacyjnych baz danych w języku SQL udało się zaimplementować bazę oraz zaprogramować aplikację, jednocześnie spełniając przy tym większość z zamierzonych funkcjonalności. Utworzone rozwiązania są zgodne z powszechnie stosowanymi standardami i mogą być rozwijane w przyszłości w celu zastosowania w rzeczywistych sytuacjach.