

Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji

Bazy danych - Projekt

“Vault of Mistakes Museum”

Autorzy:

Bartosz Kloc 259175

Dawid Wolkiewicz 259090

Prowadzący:

mgr inż. Adam Włodarczyk

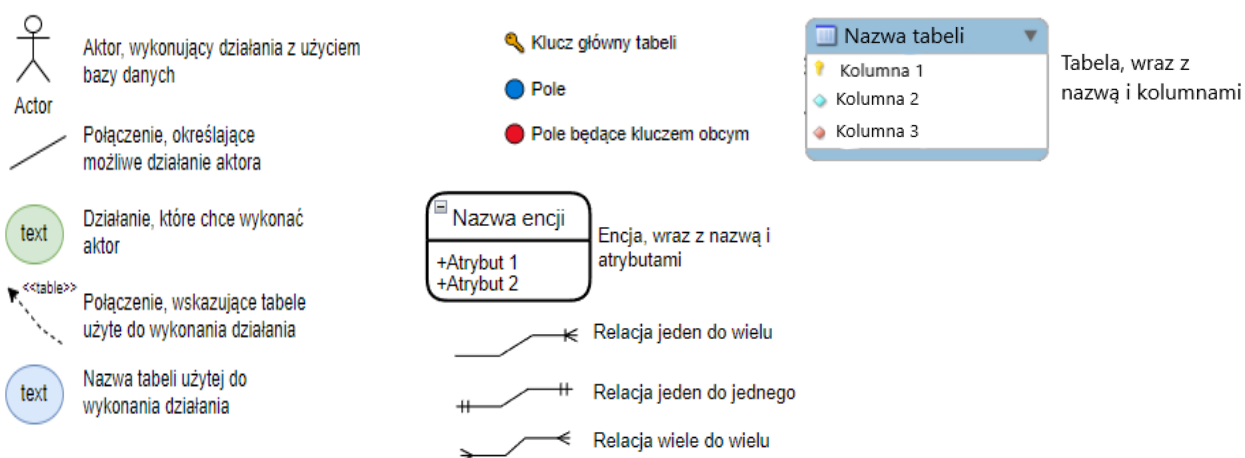
I. INFORMACJE WSTĘPNE

Sekcja zawiera podstawowe informacje potrzebne do poruszania się po dokumencie.

1. Spis treści

INFORMACJE WSTĘPNE	1
Spis treści	1
Legenda	1
OPIS BAZY DANYCH	2
Opis słowny systemu	2
Opis wymagań funkcjonalnych	2
DIAGRAMY I MODELE	3
Diagram przypadków użycia	3
Diagram związków encji	3
Model logiczny bazy	4
Model fizyczny bazy	4
ANALIZA	5
Analiza liczby instancji dla każdej encji po roku użytkowania	5
Analiza użycia identyfikująca poszczególne transakcje	6
PROGRAM	7
Środowisko programistyczne	7
Skrypty tworzące bazę danych:	7
Przykładowe fragmenty implementacji	7
ETAPY PRACY	9
Zapoznanie się z obszarem problemowym projektu	9
Projekt bazy danych	9
Implementacja bazy danych	9
Prezentacja projektu	9
Dostarczenie dokumentacji	9
WNIOSKI	9

2. Legenda



II. OPIS BAZY DANYCH

Sekcja zawiera opisy przedstawiające potrzeby projektowe bazy danych.

1. Opis słowny systemu

Przygotowana przez nas baza danych została przygotowana do zarządzania muzeum. Znajdują się w niej informacje pozwalające zarówno na administrowanie obiektem, inwentaryzację, jak i na obsługę klientów. Muzeum podzielono na piętra, pokoje, działy i wystawy, które zostały odwzorowane w bazie danych. Każdy dział tematyczny w muzeum ma przypisanego kierownika. Oprócz tego w każdym dziale znajdują się różne wystawy stałe oraz wystawy czasowe, które mogą się zmieniać. Każda wystawa zawiera eksponaty, które podlegają pracom konserwatorskim. W muzeum mamy też stanowiska pracownicze, są nimi: sprzedawca, przewodnik, woźny, ochroniarz, konserwator. Dla klientów przewidziano zróżnicowane bilety.

2. Opis wymagań funkcjonalnych

KUSTOSZ:

- Wygodniejszy wgląd do podstawowych informacji o strukturze administracyjnej muzeum
- Modyfikacja kadry pracowniczej
- Sprawdzenie stanu eksponatów muzeum i potrzeby ich konserwacji
- Wyłączenie danego eksponatu z użytku na czas trwania prac konserwatorskich
- Generowanie przychodów i wydatków obiektu
- Zarządzanie kalendarzem wystaw czasowych
- Możliwość aktualizacji mapy obiektu

KLIENT:

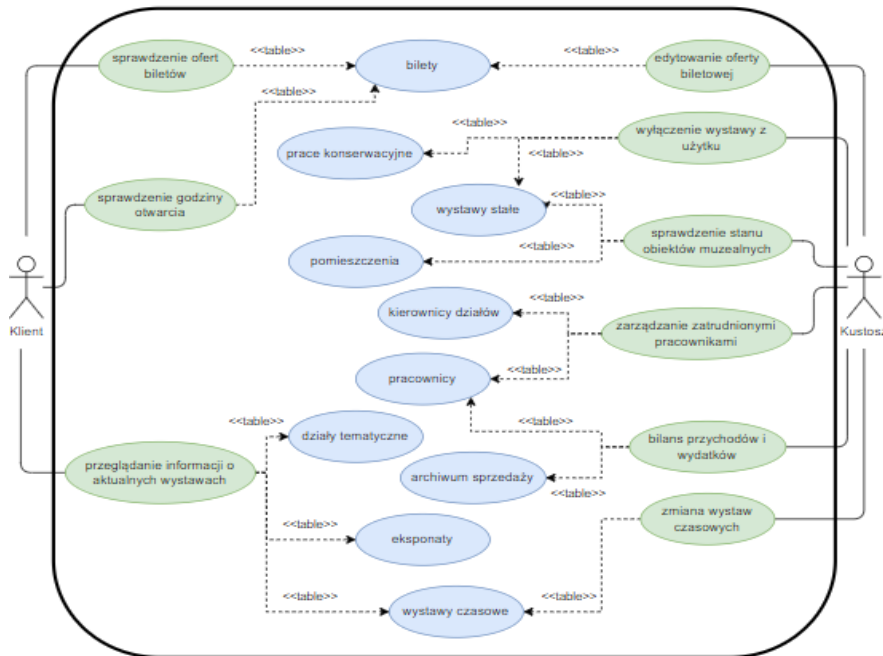
- Sprawdzenie oferty biletowej
- Przeglądanie informacji o aktualnych wystawach czasowych i stałych

III. DIAGRAMY I MODELE

Sekcja zawiera diagramy i modele przedstawiające strukturę bazy danych oraz możliwości jej wykorzystania.

1. Diagram przypadków użycia

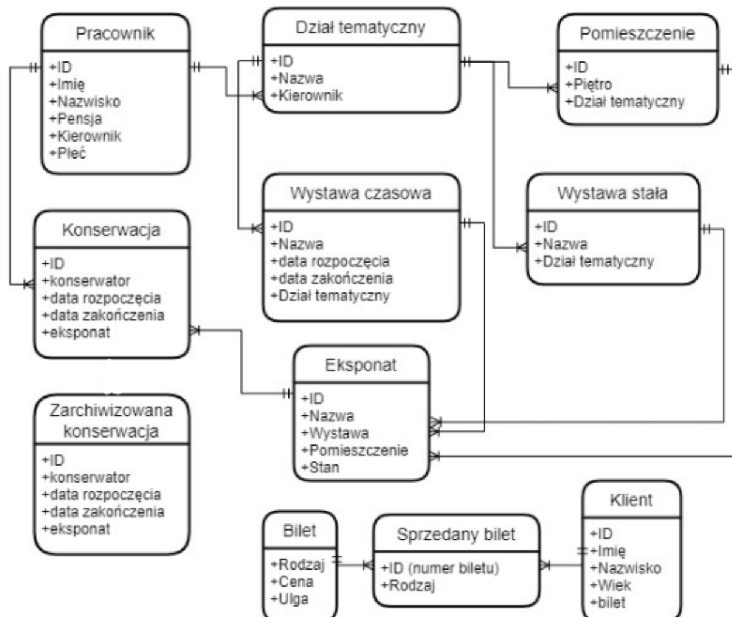
Rysunek 1. przedstawia wybrane przypadki użycia tabel przez dwóch użytkowników bazy danych - kustosa muzeum oraz klienta. Przypadek użycia połączony jest z tabelą, która zostanie wykorzystana do jego zrealizowania.



Rysunek 1. Wybrane przypadki użycia tabel w bazie danych

2. Diagram związków encji

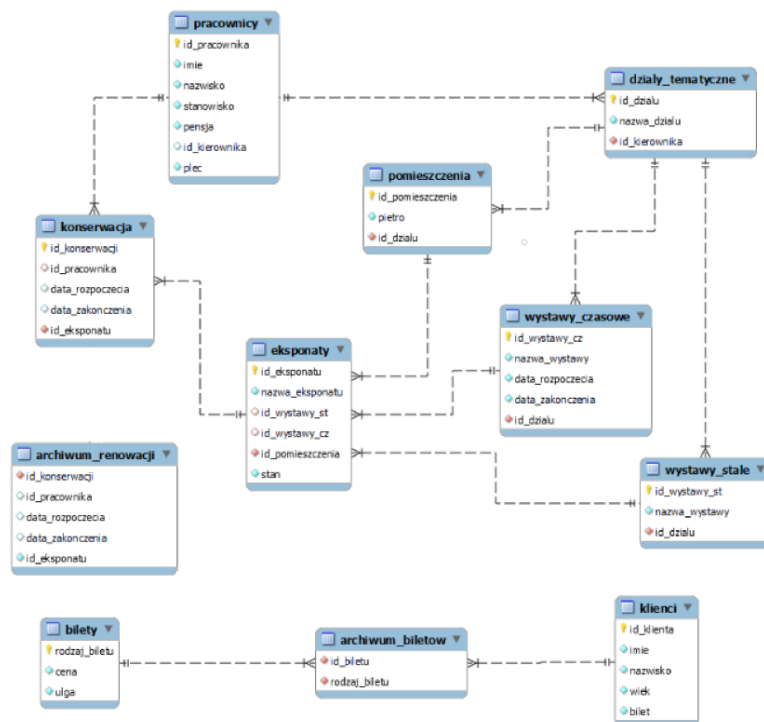
Rysunek 2. przedstawia encje, reprezentowane przez tabele w bazie danych, wraz z wyszczególnionymi atrybutami oraz połączenia (związki), między encjami.



Rysunek 2. Związki encji między poszczególnymi tabelami

3. Model logiczny bazy

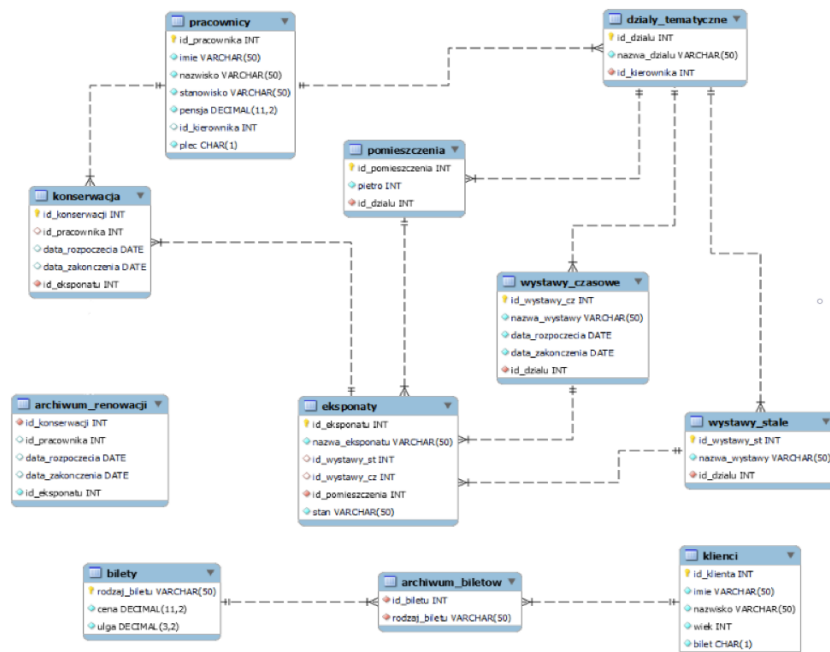
Rysunek 3. przedstawia model logiczny bazy danych. Określa on, konkretny system powinno się implementować w bazie danych bez wyszczególniania typów danych. Jego celem jest stworzenie technicznej mapy połączeń i struktur danych.



Rysunek 3. Model logiczny bazy danych

4. Model fizyczny bazy

Rysunek 4. przedstawia model fizyczny bazy danych. Określa on, jak konkretny system powinno się implementować w konkretnej bazie danych. Celem tego jest to faktyczne (fizyczne) wdrożenie bazy danych.



Rysunek 4. Model fizyczny bazy danych

IV. ANALIZA

Sekcja zawiera analizę użytkowania bazy danych.

1. Analiza liczby instancji dla każdej encji po roku użytkowania

Tabela 1. zawiera analizę liczby instancji dla poszczególnych encji, która została dobrana na podstawie danych statystycznych z już istniejących muzeów. W skład analizy wchodzi: nazwa encji, przewidywana liczba instancji po roku użytkowania bazy danych oraz opis encji.

Nazwa encji	Liczba instancji	Opis
Bilet	5	Encja zawiera rodzaje biletów możliwe do kupienia wraz z cenami
Konserwacja	7	Encja zawiera zaplanowane prace konserwatorskie danego eksponatu
Klient	30000	Lista klientów muzeum
Pracownik	50	Encja zawiera listę pracowników wraz z ich stanowiskiem i pensją
Pomieszczenie	20	Pomieszczenia w budynku
Dział tematyczny	8	Encja zawiera działy tematyczne muzeum
Wystawa CZ	4	Wystawy czasowe wraz z terminarzem oraz eksponatami jakie będą w nich wystawiane
Wystawa ST	10	Encja zawiera wszystkie wystawy stałe
Eksponat	1000	Encja zawiera wszystkie eksponaty muzealne wraz z ich rozlokowaniem oraz ich aktualnym stanem (na wystawie lub poddawany konserwacji)
Archiwum biletów	50000	Encja zawiera wszystkie sprzedane bilety, łączy tabelę Bilety z Klientami, liczba instancji to ilość rocznych transakcji
Archiwum konserwacji	50	Encja zawiera wszystkie zakończone prace konserwatorskie, liczba transakcji to ilość wszystkich prac jakie miały miejsce w muzeum

Tabela 1. Tabela przedstawia przewidywaną liczbę instancji dla poszczególnych encji po roku użytkowania bazy danych oraz opisy encji.

2. Analiza użycia identyfikująca poszczególne transakcje

- Dodanie nowego klienta
 - dodanie nowych danych osobowych
 - dodanie nowej krotki w tabeli „Klienci” i przypisanie danych osobowych do unikalnego numeru ID
 - przypisanie zakupionego biletu do klienta i umieszczenie ID biletu w tabeli “Archiwum sprzedaży”
- Zaplanowanie konserwacji eksponatu
 - dodanie nowej krotki w tabeli “Prace konserwacyjne”
 - przypisanie do pracy wybranego pracownika ze stanowiskiem “konserwator”
 - przypisanie eksponatu oraz zmiana jego statusu w tabeli “Eksponaty” na niedostępny
- Zmiana wystawy czasowej
 - usunięcie wystawy znajdującej się w wybranym pomieszczeniu z tabeli “Wystawy czasowe”
 - dodanie nowej wystawy
 - przypisanie daty rozpoczęcia i zakończenia do wystawy
- Modyfikacja cennika biletów
 - wyszukanie interesującego nas biletu w tabeli “Bilety”
 - modyfikacja pola cena
- Przeniesienie eksponatu
 - wyszukanie eksponatu w tabeli “Eksponaty”
 - modyfikacja pola ID_wystawyST/ID_wystawyCZ
- Archiwizacja prac konserwatorskich
 - usunięcie krotki w aktualnych pracach konserwatorskich (tabela “konserwacje”) w momencie zakończenia prac renowacyjnych
 - dodanie krotki z pracą konserwacyjną do archiwum

V. PROGRAM

Sekcja zawiera szczegóły implementacyjne bazy danych.

1. Środowisko programistyczne

Baza danych została wykonana w systemie zarządzania relacyjnymi bazami danych MySQL, przy pomocy wizualnego narzędzia do projektowania baz danych MySQL Workbench.

2. Skrypty tworzące bazę danych:

Baza danych tworzona jest przy pomocy 6 skryptów:

- schema_init.sql
- tabele_init.sql
- triggery_init.sql
- funkcje_i_procedury_init.sql
- widoki_init.sql
- userzy.sql

Aby poprawnie utworzyć bazę należy uruchomić schema_init.sql, a następnie przejść do bazy “muzeum” i wywołać w podanej powyżej kolejności kolejne skrypty.

3. Przykładowe fragmenty implementacji

Rysunek 5. zawiera fragment skryptu tworzącego bazę danych, odpowiedzialnego za stworzenie tabeli “konserwacja”, która zawiera informacje o aktualnie trwających pracach konserwacyjnych.

```
-- Tabela konserwacja --
CREATE TABLE konserwacja (
  id_konserwacji int NOT NULL AUTO_INCREMENT,
  id_pracownika int,
  data_roz poczenia date,
  data_zakonczenia date,
  id_eksponatu int NOT NULL,
  PRIMARY KEY (id_konserwacji),
  FOREIGN KEY (id_eksponatu) REFERENCES eksponaty(id_eksponatu),
  FOREIGN KEY (id_pracownika) REFERENCES pracownicy(id_pracownika),
  CONSTRAINT CHECK (data_roz poczenia < data_zakonczenia)
);
```

Rysunek 5. Skrypt tworzący tabelę “konserwacja”

Rysunek 6. zawiera fragment skryptu tworzącego bazę danych, odpowiedzialnego za stworzenie triggera “po_renowacji”, który odpowiada za zmianę stanu eksponatu na ‘stabilny’ po tym jak zostanie zakończona jego renowacja.

```
-- Usunięcie z konserwacji --
CREATE TRIGGER po_renowacji
AFTER DELETE
ON konserwacja
FOR EACH ROW
UPDATE eksponaty
SET eksponaty.stan = 'stabilny'
WHERE eksponaty.id_eksponatu not in (SELECT id_eksponatu FROM konserwacja);
```

Rysunek 6. Skrypt tworzący trigger odpowiedzialny za zmianę stanu eksponatu po zakończeniu konserwacji

Rysunek 7. zawiera fragment skryptu tworzącego bazę danych, odpowiedzialnego za stworzenie widoku “bilans”, pokazującego bilans wydatków na pensje pracowników oraz przychodów ze sprzedaży biletów w muzeum.

```
-- Bilans wydatków i przychodów --
DROP VIEW IF EXISTS bilans;

CREATE VIEW bilans AS
WITH przychody_h (przychody) AS (SELECT SUM(b.cena-(b.cena*b.ulga))
FROM archiwum_biletow a JOIN bilety b USING(rodzaj_biletu)),
wydatki_h (wydatki) AS ( SELECT SUM(p.pensja) FROM pracownicy p)
SELECT p.przychody-w.wydatki BILANS FROM przychody_h p, wydatki_h w;
```

Rysunek 7. Skrypt tworzący widok całkowitego bilansu wydatków i przychodów

Rysunek 8. zawiera fragment skryptu tworzącego bazę danych, odpowiedzialnego za stworzenie widoku “niestabilne”, który zawiera informacje o eksponatach potencjalnie nadających się do konserwacji, czyli takich, które nigdy nie przeszły renowacji lub i nie są aktualnie w jej trakcie.

```
-- Eksponaty do konserwacji --
DROP VIEW IF EXISTS niestabilne;

CREATE VIEW niestabilne AS
(SELECT e.nazwa_eksponatu EKSPONAT, (CASE WHEN e.id_wystawy_cz !=-1 THEN cz.nazwa_wystawy
ELSE st.nazwa_wystawy END) WYSTAWA, CONCAT(p.pietro,'/',p.id_pomieszczenia) 'PIETRO/POMIESZCZENIE'
FROM eksponaty e
JOIN wystawy_czasowe cz USING (id_wystawy_cz)
JOIN wystawy_stale st USING (id_wystawy_st)
JOIN pomieszczenia p USING (id_pomieszczenia)
WHERE e.stan = 'niestabilny');
```

Rysunek 8. Skrypt tworzący widok eksponatów, które potrzebują konserwacji

Rysunek 9. zawiera skrypt tworzący użytkowników bazy danych i przyznający im uprawnienia zgodnie diagramem przypadków użycia. Do konta kustosza przypisano również hasło.

```
#tworzenie konta Kustosza zabezpieczonego hasłem wraz z uprawnieniami do modyfikacji bazy
CREATE USER 'kustosz'@'localhost' IDENTIFIED BY 'kUst00$2Z';
GRANT ALL PRIVILEGES ON * . * TO 'kustosz'@'localhost';

#tworzenie użytkownika "klient" z możliwością przeglądania wybranych tabel
CREATE USER 'klient'@'localhost';
GRANT SELECT ON bilety TO 'klient'@'localhost';
GRANT SELECT ON dzialy_tematyczne TO 'klient'@'localhost';
GRANT SELECT ON wystawy_stale TO 'klient'@'localhost';
GRANT SELECT ON wystawy_czasowe TO 'klient'@'localhost';
GRANT SELECT ON eksponaty TO 'klient'@'localhost';
```

Rysunek 9. Skrypt tworzący użytkowników wraz z uprawnieniami

VI. ETAPY PRACY

Sekcja zawiera opis kolejnych etapów powstawania projektu.

1. Zapoznanie się z obszarem problemowym projektu

Pierwszym etapem pracy było zapoznanie się z zagadnieniami dotyczącymi projektowania bazy danych. Aby utworzyć opis wymagań funkcjonalnych należało przemyśleć jakie działania w muzeum mogą zostać przedstawione w bazie danych oraz kto może potrzebować je wykonać.

2. Projekt bazy danych

Kolejnym krokiem było utworzenie modelu teoretycznego bazy danych, dzięki któremu możliwe było wstępne zadeklarowanie tabel wchodzących w skład bazy danych. Pola w tabelach zostały dobrane tak, aby odzwierciedlać transakcje przeprowadzane w prawdziwym muzeum przez osoby nim zarządzające.

3. Implementacja bazy danych

Przed rozpoczęciem implementacji należało utworzyć serwer w narzędziu MySQL Workbench, na którym będą przeprowadzane dalsze testy operacji na bazie danych. Pierwszy został utworzony skrypt generujący tabele, został on odwzorowany na podstawie wcześniej stworzonego modelu logicznego i fizycznego. Następnie powstał skrypt dodający wyzwalacze, których zadaniem było modyfikowanie logicznie połączonych tabeli, aby dane w nich odpowiednio do siebie pasowały. Później został napisany skrypt tworzący widoki, mają one za zadanie przedstawiać przydatne dla użytkowników zestawienia danych z wielu tabel. Już istniejące tabele były później wypełniane danymi testowymi w celu sprawdzenia poprawności implementacji.

4. Prezentacja projektu

Podczas prezentacji projektu bazy danych został wykorzystany między innymi skrypt tests.sql, który zawiera zapytania pokazujące przykładowe możliwości użycia bazy oraz zapytania testujące wybrane funkcjonalności bazy, takie jak np. triggerzy. Zapytania w tym pliku zostały przez nas przedstawione prowadzącemu oraz omówione wraz z odniesieniami do implementacji w kodzie.

5. Dostarczenie dokumentacji

Dokumentacja projektu powstawała przez cały okres tworzenia bazy danych, była na bieżąco rozwijana i modyfikowana w razie potrzeby, czyli jeśli zaszły zmiany, które należało nanieść na aktualnie przedstawiony na schematach system. Opisy poszczególnych sekcji, podsekcji oraz rysunków zostały wykonane zgodnie z zaleceniami prowadzącego, naniesiono również wszystkie zaproponowane poprawki.

VII. WNIOSKI

Tworzenie bazy danych przebiegło bez większych problemów. Pewną drobną niedogodnością było używanie na zajęciach laboratoryjnych środowiska Oracle zamiast MySQL w którym język SQL minimalnie się różni. Lekko uciążliwe było wypełnienie bazy dużą ilością danych, baza lepiej by się prezentowała gdyby była faktycznie użytkowana przez jakiś czas. Mimo wszystko zadaliśmy o wystarczającą ilość danych by zaprezentować działanie triggerów, pokazać zależności między tabelami i stworzyć ciekawe widoki.