

# Wojskowa Akademia Techniczna

Wydział Elektroniki, Instytut Systemów Łączności

Zakład Systemów Telekomunikacyjnych

Bazy danych

(Bd)

## Instrukcja Laboratoryjna

Opracował:

dr inż. Łukasz Rybak

dr inż. Krzysztof Maślanka

dr hab. inż. Janusz Dudczyk, prof WAT

Warszawa 2023

## Spis treści

1	Organizacja ćwiczenia.....	3
1.1	Czas realizacji ćwiczenia .....	3
1.2	Miejsce realizacji ćwiczenia i organizacja grupy realizującej ćwiczenie .....	3
2	Wprowadzenie .....	3
3	Zadania do wykonania.....	4
3.1	Ocena dostateczna (3.0).....	4
3.2	Ocena dobra (4.0).....	4
3.3	Ocena bardzo dobra (5.0).....	4
3.4	Sposób oceny realizowanych zajęć .....	4
4	Instrukcja realizacji projektu .....	5

## 1 Organizacja ćwiczenia

### 1.1 Czas realizacji ćwiczenia

Na realizację ćwiczenia przewidziano 16 godzin lekcyjnych.

### 1.2 Miejsce realizacji ćwiczenia i organizacja grupy realizującej ćwiczenie

Ćwiczenie będzie realizowane w sali 119/47 i 22/47.

## 2 Wprowadzenie

Obecnie bazy danych pracują „pod” graficznym interfejsem użytkownika GUI (ang. *Graphic User Interface*) niemalże każdego systemu informatycznego. Poprawnie zaprojektowane bazy danych z zastosowaniem operacji realizowanych na danych, udostępnianych przez systemy zarządzania bazami danych, umożliwiają modelowanie zjawisk zachodzących w świecie rzeczywistym. Powyższe prowadzi do możliwości wytwarzania użytecznych aplikacji/systemów bazodanowych, funkcjonujących w różnych sektorach gospodarki.

Proces wytwarzania użytecznego oprogramowania, zgodnie z ogólnymi praktykami stosowanymi w biznesie, składa się z pięciu kluczowych etapów: analizy, projektowania, implementacji, wdrożenia i utrzymania. Zakres merytoryczny niniejszych zajęć laboratoryjnych obejmuje pierwsze trzy z wymienionych etapów: analizę, projektowanie, implementację oraz przygotowanie do kolejnego kroku – wdrożenia.

1. Na etapie analizy fragmentu rzeczywistości, będącego przedmiotem zainteresowania przez zespół projektowy, abstrahuje się od szczegółów implementacyjnych. Wówczas opracowywana jest ogólna koncepcja systemu, na którą składają się:
  - funkcjonalny opis analizowanego fragmentu świata rzeczywistego,
  - wymagania funkcjonalne i niefunkcjonalne aplikacji/systemu,
  - diagram związków encji ERD (ang. *Entity Relationship Diagram*).
2. Na etapie projektowania, korzystając z materiałów opracowanych na poprzednim etapie, uwzględniając już szczegóły implementacyjne aplikacji/systemu powstają:
  - lista formularzy z przypisanymi im funkcjonalnościami aplikacji/systemu oraz schematem interakcji pomiędzy nimi,
  - projekt (papierowy lub cyfrowy) GUI poszczególnych formularzy,
  - znormalizowany model relacyjny (schemat bazy danych, powstały w wyniku transformacji ERD).
3. Na etapie implementacji powstają:
  - baza danych spójna z zaprojektowanym schematem,
  - aplikacja udostępniająca zdefiniowane funkcjonalności i spełniająca nałożone na nią wymagania.

W kontekście merytoryki niniejszych zajęć laboratoryjnych zwieńczeniem trzech opisanych etapów procesu wytwarzania oprogramowania, pozwalającym na późniejszą realizację etapu wdrożenia aplikacji/systemu jest opracowanie jej instrukcji uruchomienia i obsługi.

**Celem zajęć**, przy zastosowaniu wiedzy wykładowej i poznanych dobrych praktyk stosowanych w biznesie, jest zespołowe zaprojektowanie i implementacja aplikacji/systemu bazodanowego dla wypożyczalni samochodów dysponującej ich własnym serwisem.

### 3 Zadania do wykonania

Zadania do wykonania (oczekiwania Zamawiającego aplikację/system)

#### 3.1 Ocena dostateczna (3.0)

1. Aplikacja ma otwierać się formularzem startowym, na którym mają znajdować się informacje o autorach aplikacji BD.
2. W bazie należy utworzyć formularz z tekstem, w którym ma być umieszczony własny opis projektu semestralnego. W opisie należy umieścić przyjęte założenia działania wypożyczalni oraz uzasadnienie dla przyjętych rozwiązań, reguły poprawności itd.
3. Po wyszukaniu samochodu dla klienta wśród dostępnych aktualnie w wypożyczalni według kryterium: jego nazwy; ceny wypożyczenia na godz./dzień - i po wyszukaniu zapisanego w bazie klienta (według początku nazwiska) lub zapisaniu informacji o nowym kliencie system powinien umożliwić zapisanie informacji o nowym wypożyczeniu.
4. Po wyszukaniu informacji o kliencie oddającym samochód (według np. początku nazwiska) i po wyszukaniu informacji o pożyczonym przez niego samochodzie system powinien umożliwić zapisanie informacji o zwrocie danego auta przez klienta.
5. Przy wypożyczaniu i zwrocie auta powinny być poprawnie naliczane (i zapisywane w bazie oraz wyświetlane na formularzu) opłaty za wypożyczanie oraz dodatkowe opłaty np. za nieoddanie w terminie, uszkodzenie auta itp. zgodnie z założeniami działania wypożyczalni przyjętymi przez studenta.

#### 3.2 Ocena dobra (4.0)

6. System ma również umożliwić zapisanie informacji o oddaniu do naprawy auta uszkodzonego i jego odbiorze z naprawy. Wprowadzić dane testowe i sprawdzić poprawność działania systemu.
7. Wyświetlać, co najmniej następujące informacje:
  - suma opłat, które klienci zapłacili za auta wypożyczone w przedziale czasu zadawanym w formie parametru,
  - najczęściej psujące się auto (lub auta, jeśli kilka było tyle samo razy w naprawie),
  - auta, które nigdy nie były w naprawie,
  - auta, które były wypożyczane najczęściej (wziąć pod uwagę liczbę wypożyczeń),
  - auto, które było wypożyczone najdłużej (wziąć pod uwagę łączną sumę godzin/dni jego wszystkich wypożyczeń).

#### 3.3 Ocena bardzo dobra (5.0)

8. Utworzyć raporty (używając różnych typów wykresów) wyświetlające:
  - zestawienie aut wypożyczonych w dowolnie zadanym przedziale czasu
  - podawanym w formie parametru i podsumowanie kosztów wypożyczenia każdego z tych aut (wykres słupkowy);
  - zestawienie wszystkich psujących się aut w dowolnie zadanym przedziale czasu podawanym w formie parametru i podsumowanie ilości napraw dla każdego z tych aut (wykres słupkowy);
  - zestawienie wszystkich klientów w SBD, którzy wypożyczali samochody i podanie dla każdego z tych klientów łącznej kwoty, na którą każdy z nich wypożyczył samochód/samochody.

#### 3.4 Sposób oceny realizowanych zajęć

Produkty podlegające rozliczeniu i ocenie we wskazanych terminach:

- 5-6 godzina zajęć:
  - ERD (notacja Barkera lub Martina)
  - model relacyjny (w III postaci normalnej, notacja Barkera lub Martina)
- 7-8 godzina zajęć:
  - kod SQL pustej bazy danych lub wyeksportowana pusta baza danych
- 15-16 godzina zajęć:
  - aplikacja/system bazodanowy
  - instrukcja uruchomienia i użytkowania

#### 4 Instrukcja realizacji projektu

1. W celu stworzenia diagramu związków encji oraz modelu relacyjnego można zastosować bezpłatne oprogramowanie „Diagrams.net”.
2. Tworząc ERD warto rozpocząć pracę od ekstrakcji encji. Należy pamiętać, że ich nazwy są rzeczownikami w liczbie pojedynczej (nie dotyczy rzeczowników *plurale tantum*).
3. Następnie należy określić typ każdego z atrybutów encji - identyfikatory oznaczane są symbolem „#”, deskryptory wymagane znakiem „\*”, a deskryptory opcjonalne „o”.
4. W kolejnym kroku należy zdefiniować związki zachodzące pomiędzy encjami, określając ich: stopień, typ asocjacji (kardynalność) oraz klasę przynależności.
5. Posiadając diagram związków encji należy dokonać jego transformacji do modelu relacyjnego. W tym celu najpierw należy przeprowadzić transformację encji i ich atrybutów, a następnie związków, które pomiędzy nimi zachodzą. Należy pamiętać, że nazwy relacji są rzeczownikami w liczbie mnogiej (nie dotyczy rzeczowników *singulare tantum*). W tym kroku warto zwrócić szczególną uwagę na związki o kardynalności wiele do wiele, które należy zastąpić relacjami powstałymi na podstawie encji słabych.
6. Kolejno należy przeprowadzić normalizację modelu relacyjnego do III postaci normalnej.
7. Na podstawie znormalizowanego modelu relacyjnego należy dokonać implementacji/utworzenia bazy danych w wybranym systemie zarządzania relacyjną bazą danych RDBMS (ang. *Relational Database Management System*): „Microsoft Access” (Windows) lub „LibreOffice Base” (Windows, Linux)<sup>1</sup>.
8. Następnie w wybranym RDBMS należy utworzyć zaprojektowane formularze z niezbędnymi elementami GUI (etykietami, polami tekstowymi, przyciskami). Implementując formularze należy pamiętać o konieczności walidacji danych, które będą za ich pośrednictwem przetwarzane. W tym celu można zastosować maski wprowadzania lub zaimplementować autorski mechanizm walidacji danych<sup>1</sup>.
9. Następnie należy zaimplementować mechanizmy przetwarzania danych, realizujące funkcjonalności wymagane przez Zamawiającego aplikację/system bazodanowy<sup>1</sup>.
10. W kolejnym kroku należy utworzyć raporty zawierające, określone przez Zamawiającego, zestawienia zgromadzonych danych<sup>1</sup>.

---

<sup>1</sup>Instrukcje użytkowania możliwych do zastosowania RDBMS:

Microsoft Access: <https://support.microsoft.com/en-us/access>

LibreOffice Base: <https://documentation.libreoffice.org/en/english-documentation/base>