

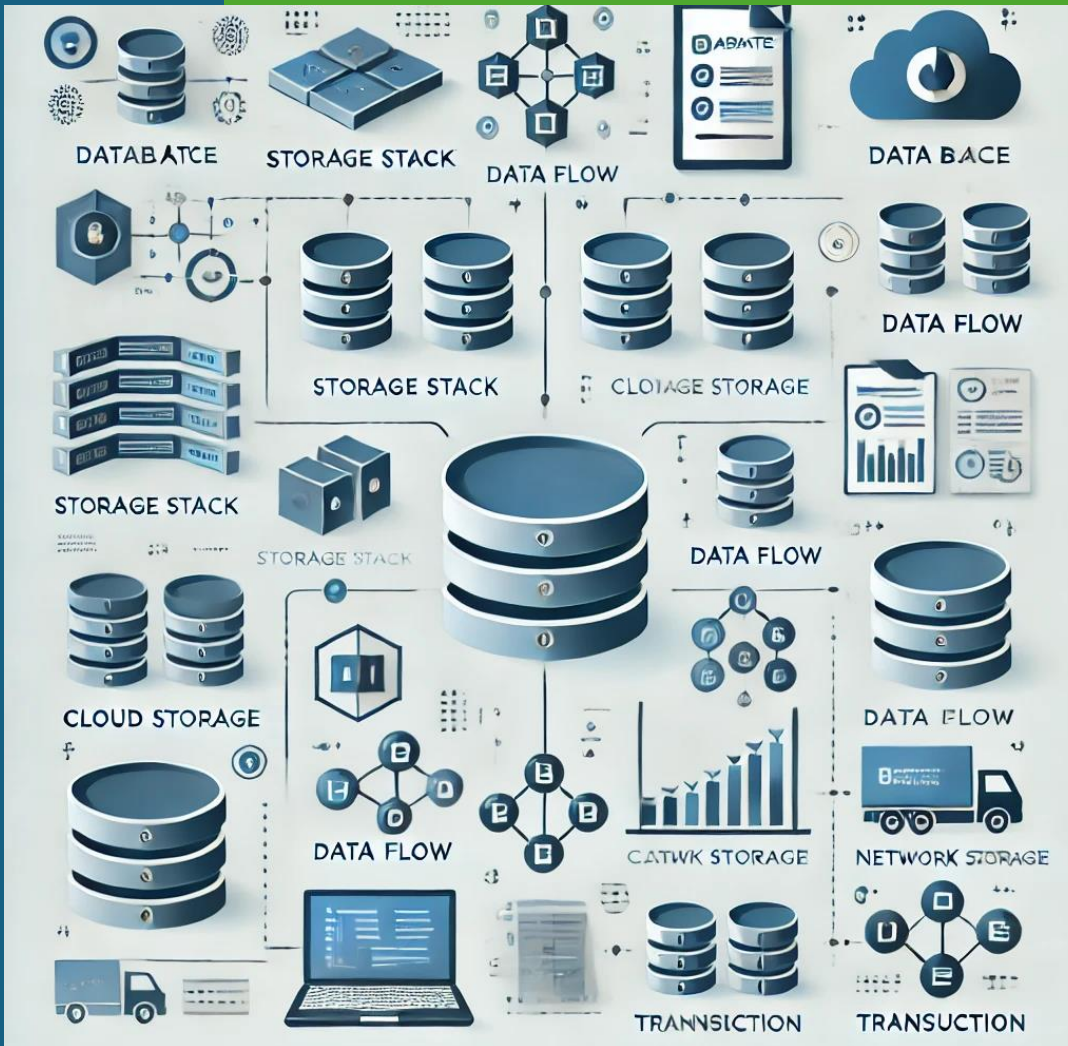
Temat 2 .

dr inż. Michał Malinowski

<https://www.drmalinowski.edu.pl>

Projektowanie relacyjnych baz danych

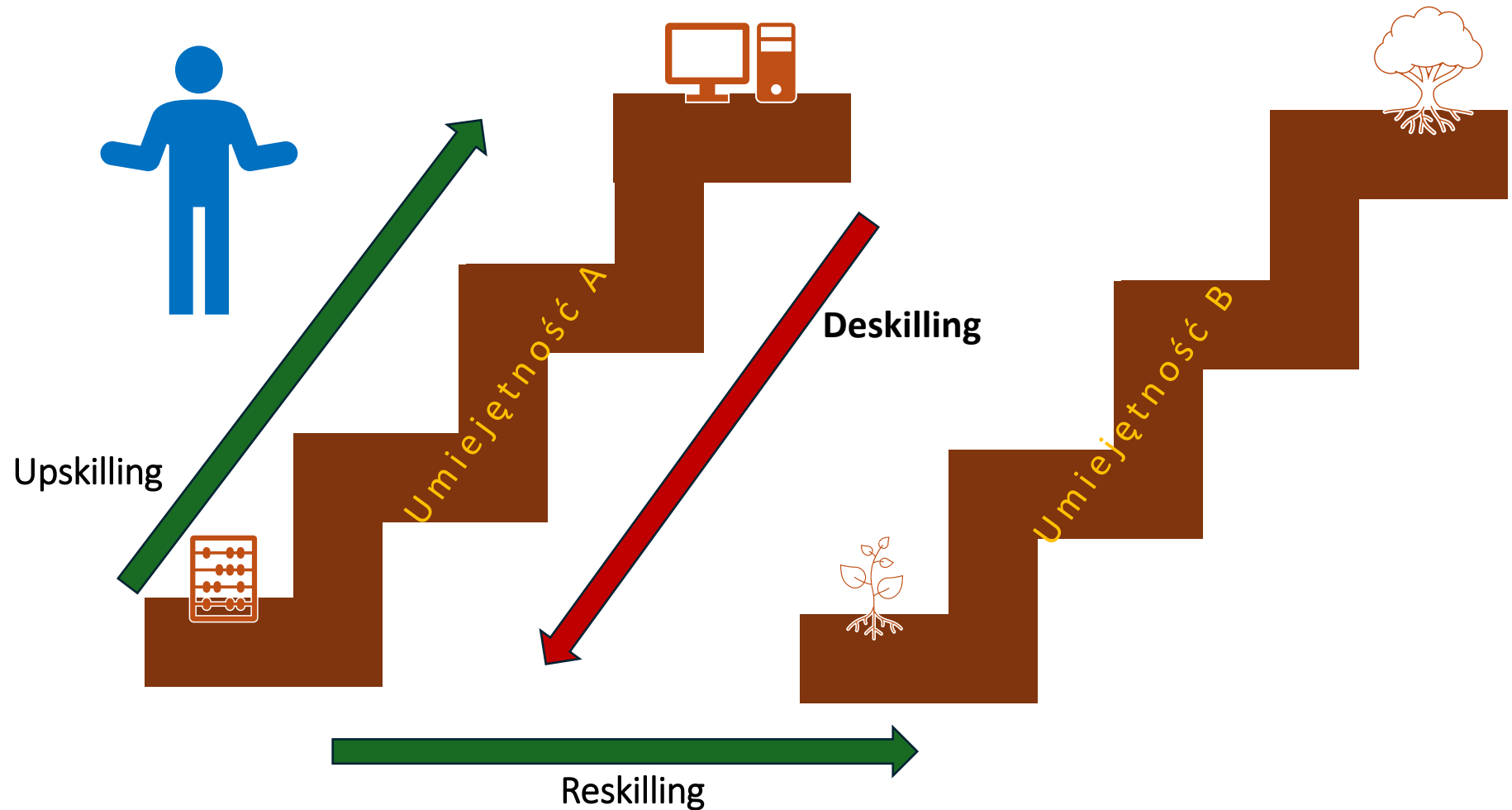
PODSTAWY BAZ DANYCH



Zagadnienia:

- Etapy procesu projektowania
- Diagram związków encji ERD
- Przekształcanie Diagramów ERD na Model Relacyjny
- Ograniczenia
- Ćwiczenie praktyczne !

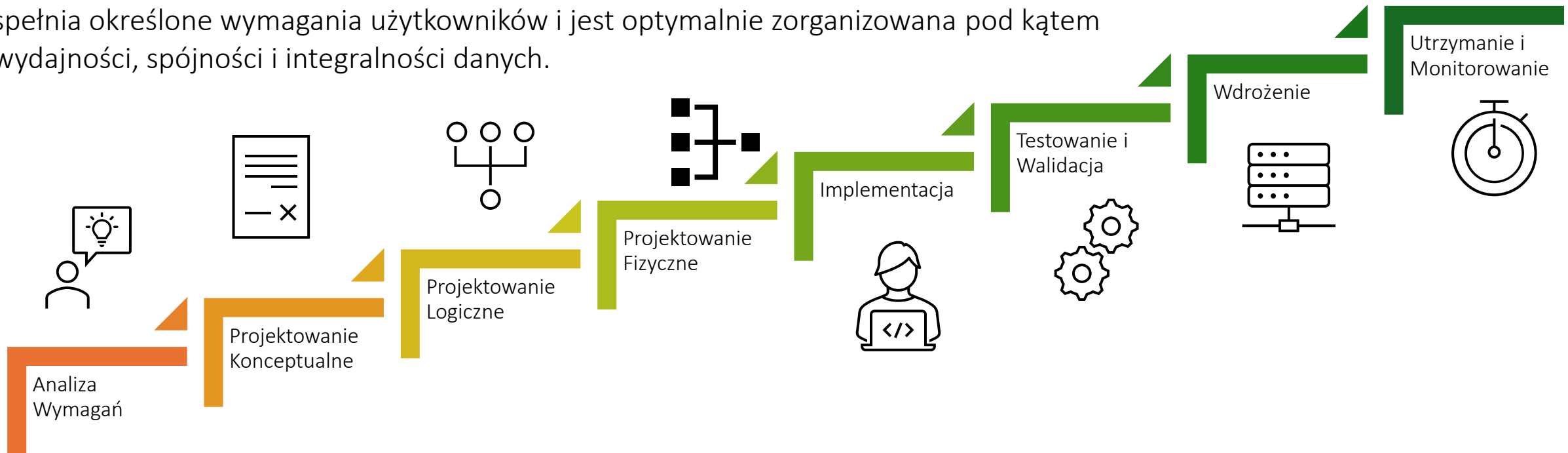
Zarządzanie kompetencjami w organizacji



Etapy procesu projektowania

Proces projektowania bazy danych

to zestaw uporządkowanych etapów mających na celu utworzenie bazy danych, która spełnia określone wymagania użytkowników i jest optymalnie zorganizowana pod kątem wydajności, spójności i integralności danych.



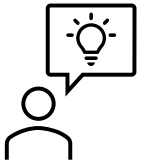
Etapy procesu projektowania

Analiza Wymagań

Zbieranie i zrozumienie wymagań użytkowników końcowych oraz specyfikacji systemu. Obejmuje identyfikację potrzebnych danych, operacji na danych oraz wymaganych raportów.

Cel: Ustalenie, co baza danych powinna zawierać i jakie funkcjonalności musi wspierać.

Produkt: Dokumentacja wymagań, specyfikacja wymagań funkcjonalnych i нефункциональных.

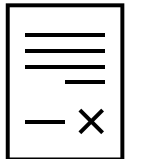


Projektowanie Konceptualne

Tworzenie abstrakcyjnego modelu danych, który odzwierciedla zidentyfikowane wymagania. Najczęściej wykorzystuje się diagramy ER (Entity-Relationship) do wizualizacji struktury bazy danych.

Cel: Reprezentacja danych oraz relacji między nimi w sposób niezależny od konkretnego systemu baz danych.

Produkt: Diagram ER (Entity-Relationship), model konceptualny danych.



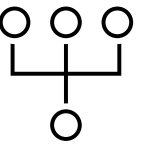
Etapy procesu projektowania

Projektowanie Logiczne

Przekształcenie modelu konceptualnego w model logiczny, który jest specyficzny dla wybranego systemu zarządzania bazą danych (DBMS). Definiowanie tabel, atrybutów, kluczy głównych i obcych oraz normalizacja schematów.

Cel: Zdefiniowanie struktury bazy danych w sposób, który umożliwia jej implementację na konkretnym DBMS.

Produkt: Schemat logiczny bazy danych, specyfikacja tabel, kluczy, normalizacja do odpowiedniej postaci normalnej (NF).

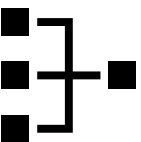


Projektowanie Fizyczne

Określenie fizycznej organizacji danych na nośnikach pamięci, wybór indeksów, ustalenie struktur plików, zarządzanie dostępem i partycjonowaniem.

Cel: Optymalizacja wydajności i zapewnienie efektywnego przechowywania oraz dostępu do danych.

Produkt: Schemat fizyczny bazy danych, plan indeksów, struktury przechowywania danych, strategia partycjonowania.



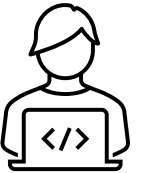
Etapy procesu projektowania

Implementacja

Tworzenie bazy danych zgodnie z projektem logicznym i fizycznym, definiowanie tabel, relacji, widoków, indeksów, procedur składowanych oraz reguł integralności.

Cel: Zbudowanie rzeczywistego systemu bazodanowego, który spełnia wszystkie zidentyfikowane wymagania.

Produkt: Działająca baza danych, skrypty SQL do tworzenia tabel, indeksów, widoków oraz procedur składowanych.

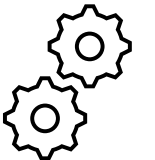


Testowanie i Walidacja

Testowanie bazy danych w celu weryfikacji poprawności implementacji, wydajności oraz zgodności z wymaganiami. Wykonywanie testów funkcjonalnych, wydajnościowych i obciążeniowych.

Cel: Zapewnienie, że baza danych działa zgodnie z oczekiwaniami i jest wolna od błędów.

Produkt: Raporty z testów, wyniki testów funkcjonalnych i wydajnościowych, lista poprawek.



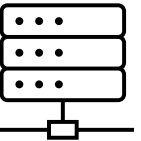
Etapy procesu projektowania

Wdrożenie

Przeniesienie bazy danych do środowiska produkcyjnego, skonfigurowanie środowiska, migracja danych z istniejących systemów oraz szkolenie użytkowników końcowych.

Cel: Udostępnienie działającej bazy danych użytkownikom w środowisku produkcyjnym.

Produkt: Działająca baza danych w środowisku produkcyjnym, dokumentacja migracji danych, przeszkoleni użytkownicy.



Utrzymanie i Monitorowanie

Monitorowanie wydajności, zarządzanie bezpieczeństwem, regularne kopie zapasowe, aktualizacje systemu oraz optymalizacje na podstawie obserwacji użytkowania.

Cel: Zapewnienie ciągłości działania, bezpieczeństwa i wydajności bazy danych w długim okresie.

Produkt: Raporty z monitorowania, bezpieczeństwa i wydajności. Aktualizacje bazy danych, kopie zapasowe.

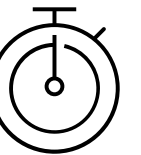
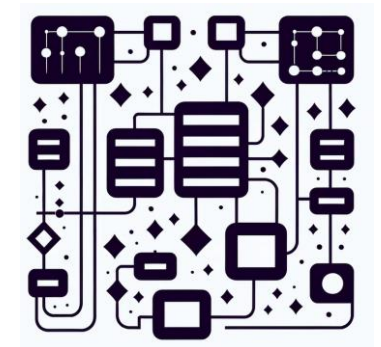


Diagram związków encji ERD

Diagram związków encji (Entity-Relationship Diagram, ERD)

to graficzna reprezentacja struktury bazy danych. Jest to narzędzie używane do modelowania i analizy systemów baz danych. ERD pomaga w zrozumieniu relacji pomiędzy różnymi encjami (takimi jak tabele w bazach danych) oraz atrybutami tych encji (takimi jak pola w tabelach).



Cechy

Reprezentacja modelu danych

Jest graficzną reprezentacją modelu danych, czyli sposobu, w jaki dane są przechowywane i powiązane ze sobą.

Modelowanie struktury bazy danych

Jest używany do modelowania struktury bazy danych, czyli sposobu, w jaki dane są przechowywane i powiązane ze sobą.

Komunikacja z interesariuszami

Jest wizualnym sposobem przedstawienia złożonych informacji o danych. Może być używany do komunikacji z interesariuszami, którzy nie są technicznymi ekspertami.

Elastyczność

Jest elastycznym narzędziem, które można dostosować do różnych potrzeb. Może być używany do modelowania różnych typów danych, w tym danych relacyjnych, obiektowych i rozproszonych.

Diagram związków encji ERD

Historia

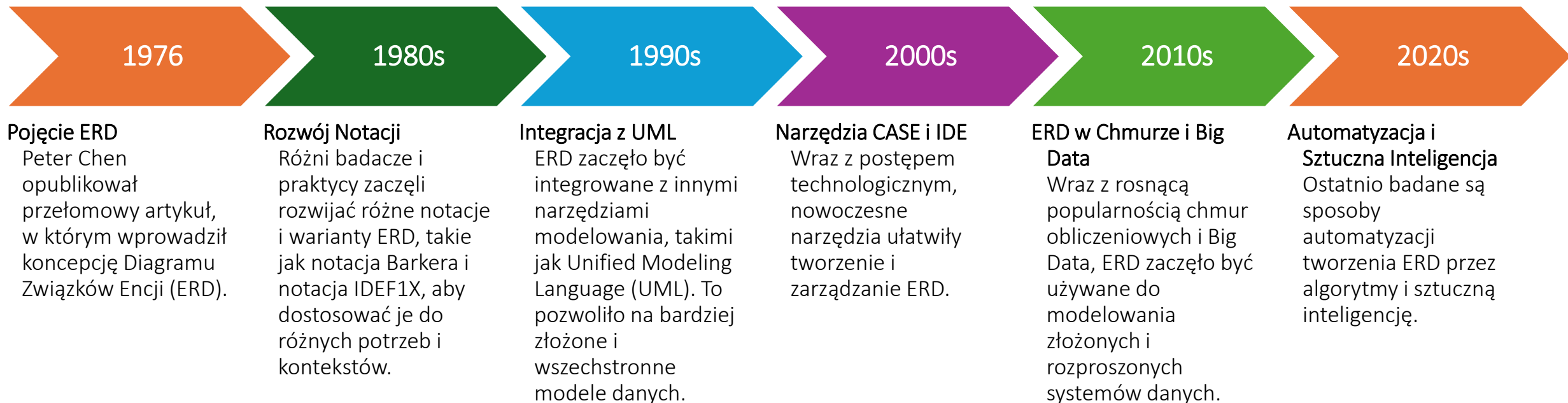


Diagram związków encji ERD

ERD obejmuje kilka rodzajów diagramów

Nazwa polska	Nazwa angielska	Opis
Koncepcyjny ERD	Conceptual ERD	Najbardziej ogólny rodzaj ERD, który służy do przedstawienia wysokopoziomowego obrazu struktury danych. Nie zawiera on szczegółowych informacji o atrybutach czy relacjach, ale skupia się na głównych encjach i ich związkach.
Logiczny ERD	Logical ERD	Bardziej szczegółowy niż model koncepcyjny i zawiera informacje o atrybutach encji oraz o relacjach między nimi. Jest to model niezależny od technologii, co oznacza, że nie uwzględnia on szczegółów implementacyjnych.
Fizyczny ERD	Physical ERD	Najbardziej szczegółowy rodzaj ERD, który uwzględnia konkretne elementy implementacyjne, takie jak tabele, klucze i indeksy w bazie danych. Jest on zwykle używany przez programistów podczas faktycznej implementacji bazy danych.
Notacja Chen	Chen Notation	Klasyczna notacja ERD, stworzona przez Petera Chena. Jest ona szeroko używana i skupia się na reprezentacji encji, atrybutów i relacji w sposób jasny i zrozumiały.
Notacja Crow's Foot	Crow's Foot Notation	Popularna notacja, która jest często używana w narzędziach do modelowania baz danych. Jest ona bardziej zorientowana na praktyczne aspekty i łatwość czytania.
IDEF1X	IDEF1X	Formalna notacja stworzona przez U.S. Air Force, która jest bardziej formalna i zawiera dodatkowe elementy, takie jak ograniczenia integralności.

Diagram związków encji ERD

Notacji Crow's Foot

to popularny sposób graficznego przedstawiania Diagramów Związków Encji (ERD), w którym encje są reprezentowane jako prostokąty, a ich związki za pomocą linii z symbolami na końcach, przypominającymi "łapki kruka". Te symbole określają kardynalność związku, czyli ilość rekordów jednej encji powiązanych z rekordami drugiej encji.

Zastosowanie

Projektowanie Baz Danych

Jest często używana przez architektów i projektantów baz danych do modelowania struktury danych przed jej implementacją.

Analiza Biznesowa

Analitycy biznesowi korzystają z tej notacji do zrozumienia i dokumentowania zasobów informacyjnych w organizacji.

Inżynieria Oprogramowania

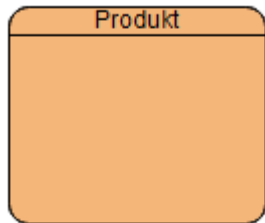
W fazie projektowania systemów informacyjnych, programiści i inżynierowie korzystają do przedstawienia, jak różne komponenty systemu będą ze sobą interagować na poziomie danych.

Edukacja i Szkolenia

Jest powszechnie używana w materiałach dydaktycznych i kursach z zakresu baz danych, dzięki swojej czytelności i łatwości zrozumienia.

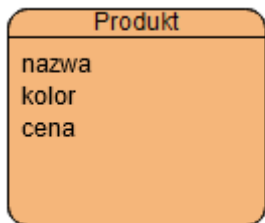
Diagram związków encji ERD

Element notacji



Encja (Entity)

Podstawowy obiekt w bazie danych. Na przykład, "Klient" czy "Produkt".

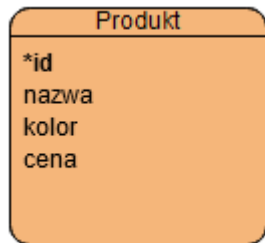


Atrybut (Attribute)

Właściwości encji. Na przykład, „Nazwa” czy „kolor”.

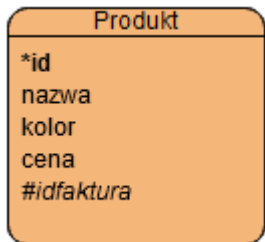
Diagram związków encji ERD

Element notacji



Klucz Główny (Primary Key)

Atrybut lub zestaw atrybutów, które jednoznacznie identyfikują rekord w encji (*).



Klucz Obcy (Foreign Key)

Atrybut służący do łączenia dwóch różnych encji (#)

Diagram związków encji ERD

Element notacji

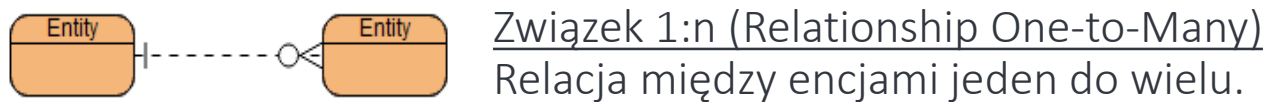
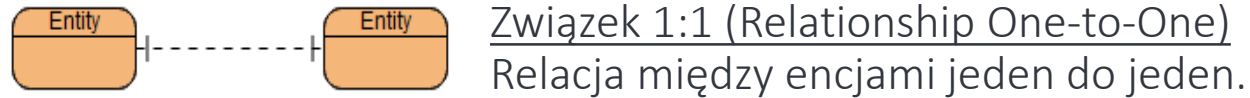
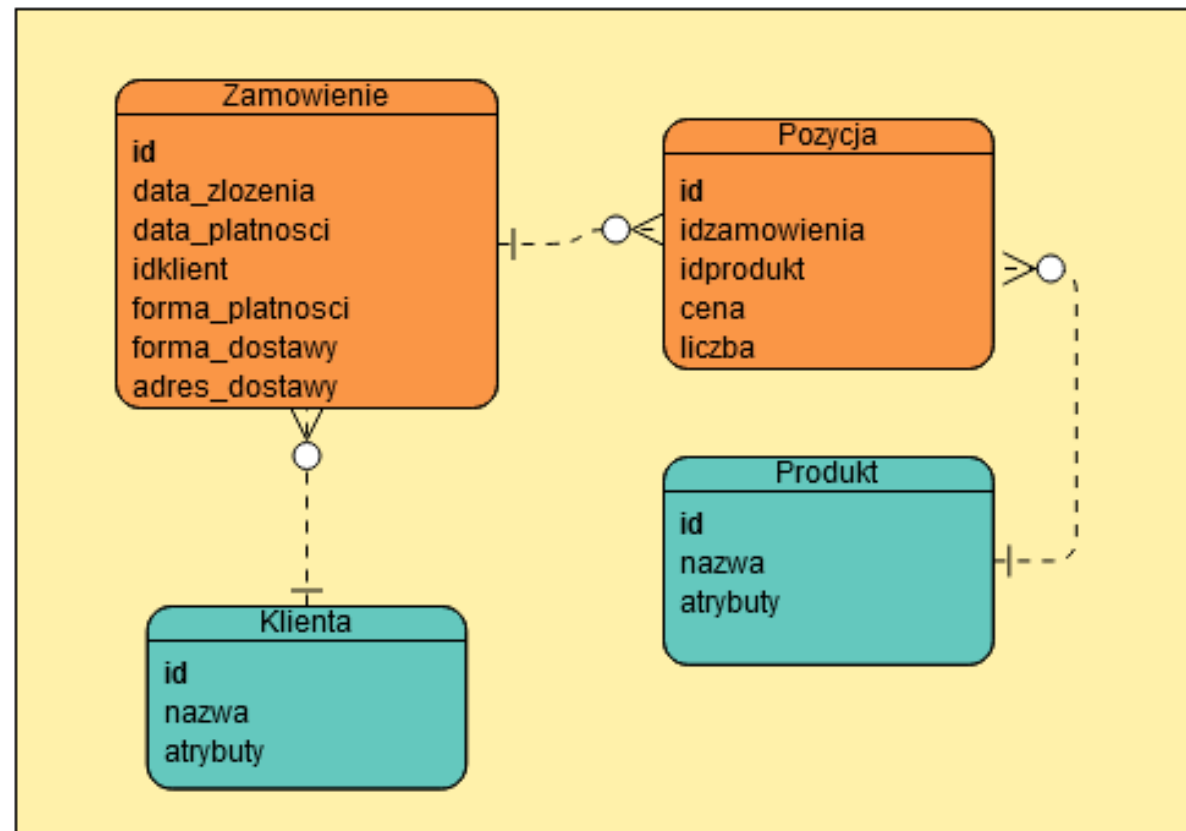


Diagram związków encji ERD

Przykład – sklep online



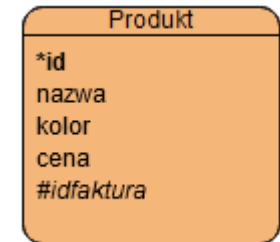
Przekształcanie Diagramów ERD na Model Relacyjny

Przekształcanie Encji na Tabele

Zasada: Każda encja (entity) w diagramie ERD staje się tabelą (relacją) w modelu relacyjnym.

Atrybuty encji: Atrybuty encji stają się kolumnami w tabeli.

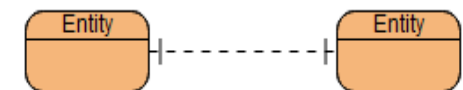
Klucz główny: Klucz główny encji staje się kluczem głównym tabeli.



Przekształcanie Związków Jeden-do-Jednego (1:1)

Zasada: Związek 1:1 może być odwzorowany przez dodanie klucza obcego do jednej z tabel lub przez połączenie obu encji w jedną tabelę.

Klucz obcy: Klucz główny jednej encji staje się kluczem obcym w tabeli drugiej encji.

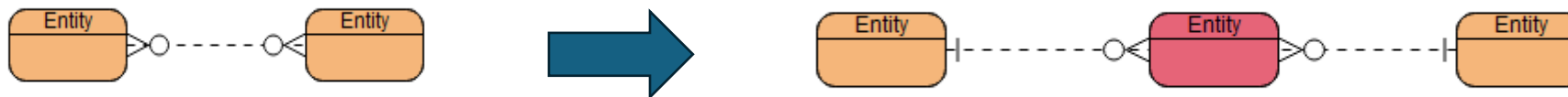


Przekształcanie Diagramów ERD na Model Relacyjny

Przekształcanie Związków Wiele-do-Wielu (n:n)

Zasada: Związek n:n przekształca się poprzez wprowadzenie tabeli pośredniczącej (łączącej, intersekcji), która przechowuje klucze główne obu encji.

Tabela pośrednicząca: Tworzenie nowej tabeli, która zawiera klucze obce do obu tabel powiązanych encji. Każdy rekord w tabeli pośredniczącej reprezentuje powiązanie jednego rekordu z jednej tabeli z jednym rekordem z drugiej tabeli.

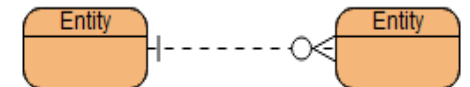


Przekształcanie Diagramów ERD na Model Relacyjny

Przekształcanie Związków Jeden-do-Wielu (1:n)

Zasada: W relacji 1:n, klucz główny encji po stronie "jeden" staje się kluczem obcym w tabeli encji po stronie "wiele".

Implementacja: Tworzenie relacji przez dodanie klucza obcego do tabeli po stronie "wiele".



Utrzymanie Integralności i Ograniczeń

Klucze główne i obce: Definiowanie kluczy głównych i obcych w modelu relacyjnym, aby zachować integralność referencyjną.

Ograniczenia: Wprowadzenie ograniczeń (np. NOT NULL, UNIQUE, itd...) na poziomie bazy danych, aby odzwierciedlić reguły biznesowe.

Ograniczenia

Ograniczenia (constraints)

to reguły lub zasady narzucane na dane w tabelach bazy danych w celu zapewnienia ich spójności, integralności i zgodności z założeniami biznesowymi oraz logiką aplikacji.

Ograniczenia pomagają kontrolować jakie dane mogą być wprowadzane do bazy danych, jak są one przechowywane i jak dane w różnych tabelach mogą się ze sobą powiązywać.

Zapewnienie poprawności danych:

- Ograniczenia uniemożliwiają wprowadzenie danych, które są sprzeczne z logiką aplikacji lub zasadami biznesowymi (np. niepoprawne wartości, duplikaty).

Zachowanie integralności referencyjnej:

- Ograniczenia pomagają utrzymać poprawne relacje między tabelami, np. poprzez klucze obce.

Zwiększenie bezpieczeństwa:

- Poprzez wymuszenie określonych reguł, ograniczenia mogą również pomóc w zabezpieczeniu danych przed nieautoryzowanymi zmianami.

Ograniczenia

Klucz główny (PRIMARY KEY)

Ograniczenie, które jednoznacznie identyfikuje każdy rekord w tabeli. Kolumna (lub zestaw kolumn) oznaczona jako PRIMARY KEY musi mieć unikalne wartości i nie może zawierać wartości NULL.

Przykład: W tabeli „Pracownicy” kolumna „ID” może być PRIMARY KEY, co oznacza, że każda wartość w tej kolumnie jest unikalna i nie może być pusta.

Klucz obcy (FOREIGN KEY)

Ograniczenie, które tworzy relację między dwiema tabelami. Klucz obcy (FOREIGN KEY) w jednej tabeli odnosi się do klucza głównego (PRIMARY KEY) w innej tabeli, zapewniając integralność referencyjną.

Przykład: W tabeli „Zamówienia” kolumna „KlientID” może być FOREIGN KEY odnoszącym się do kolumny „ID” w tabeli „Klienci”, zapewniając, że każde zamówienie jest powiązane z istniejącym klientem.

Ograniczenia

CHECK

Ograniczenie, które zapewnia, że wszystkie wartości w kolumnie spełniają określony warunek. Używane do sprawdzania, czy wartości w kolumnie spełniają zdefiniowane kryteria.

Przykład: W tabeli „Pracownicy” kolumna „Wiek” może mieć ograniczenie CHECK, które wymusza, że wartości muszą być większe lub równe 18.

DEFAULT

Ograniczenie, które ustawia domyślną wartość dla kolumny, jeśli podczas wstawiania rekordu nie zostanie podana wartość dla tej kolumny.

Przykład: W tabeli „Zamówienia” kolumna „DataZamówienia” może mieć ograniczenie DEFAULT ustawiające domyślną wartość na bieżącą datę.

Ograniczenia

AUTO_INCREMENT (lub SERIAL w niektórych DBMS)

Ograniczenie, które automatycznie generuje unikalne liczby całkowite dla nowych rekordów w kolumnie, często stosowane dla klucza głównego.

Przykład: W tabeli „Klienci” kolumna „ID” może być AUTO_INCREMENT, co oznacza, że przy każdym nowym rekordzie numer ID zostanie automatycznie zwiększony o 1.

Index (INDEX)

Tworzy wskaźnik (indeks) w kolumnie lub kolumnach, co przyspiesza operacje wyszukiwania i sortowania na tych kolumnach.

Przykład: W tabeli „Produkty” można utworzyć indeks na kolumnie „NazwaProduktu”, aby przyspieszyć wyszukiwanie produktów według nazwy.

Ograniczenia

Indeks (INDEX) w bazie danych

jest strukturą danych, która wspiera szybkie operacje wyszukiwania i dostępu do danych w tabeli.

Są to oddzielne struktury w bazach danych i pod względem struktury danych i złożoności czasowej, indeksy mogą być realizowane na różne sposoby, a najczęściej stosowane struktury to drzewa B-Tree i tablice mieszające (hash tables).

Kategoria	Opis B-Tree
Struktura danych	Zbalansowane drzewo wyszukiwania, gdzie wszystkie liście są na tym samym poziomie.
Wyszukiwanie	Operacja wyszukiwania ma złożoność czasową $O(\log n)$, co oznacza szybki dostęp do danych.
Wstawianie i Usuwanie	Wstawianie i usuwanie elementów również ma złożoność $O(\log n)$, co zapewnia efektywne przekształcanie drzewa.
Zastosowanie	Najczęściej wykorzystywane w bazach danych do indeksowania tabel, co przyspiesza operacje wyszukiwania.
Optymalizacja I/O	Minimalizuje liczbę operacji wejścia/wyjścia na dysku, co jest kluczowe dla wydajności w dużych bazach danych.
Zalety	Zbalansowana struktura, szybki dostęp, efektywność przy dużych zbiorach danych.
Wady	Złożoność implementacji i konieczność utrzymania równowagi drzewa przy operacjach wstawiania/usuwania.

Wykorzystanie przestrzeni	
Typ	Używanie
Dane	19,235.1 KB
Indeks	80,612.0 KB
Rekordy	531,808

Index ~ 3x dane

Ćwiczenie praktyczne

Treść

Zmodeluj struktury danych za pomocą diagramu ERD dla „Sklepu internetowego - Cytrynka”:

- Sklep internetowy sprzedaje różnorodne produkty, które są dostępne w różnych kategoriach.
- Każde zamówienie zawiera informacje o zamawiającym kliencie, ilości każdego produktu w zamówieniu oraz sposobie płatności i dostawy.
- Sklep oferuje różne metody płatności oraz opcje dostawy.

Edytor diagramów ERD <https://online.visual-paradigm.com/>

Należy wynik przestać w Moodle.

