BAZY DANYCH

WYKŁAD I

INFORMACJE, DANE, BAZA DANYCH

BAZA DANYCH to zbiór danych opisujący pewien wybrany fragment rzeczywistości. Zbiór ten zapisany jest w ściśle określony sposób, w strukturach odpowiadających założonemu modelowi danych.

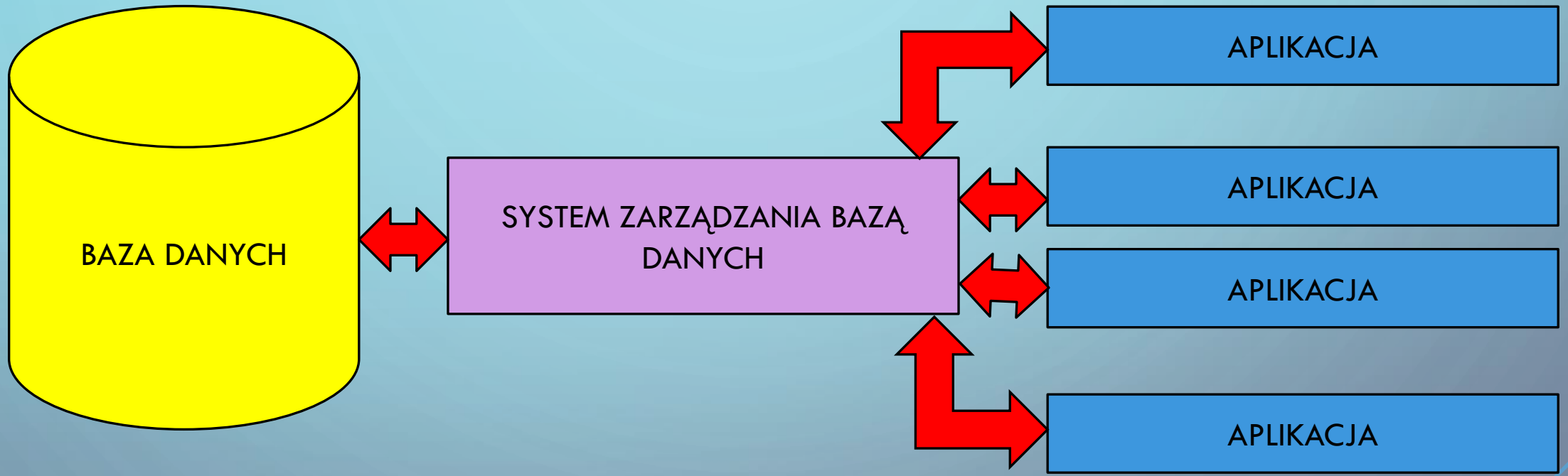
BAZA DANYCH to dane wraz ze strukturą (sposobem) ich przechowywania. Struktura danych i powiązania między nimi są opisane przez tzw. schemat bazy danych.

Informacja, dane – to pewien zasób, którego celem jest opis fragmentu rzeczywistości. Dane są jednym z zasobów i tak jak pozostałe zasoby wymagają zarządzania. Informacja to dane razem z ich semantyką.

SZBD

SZBD (System zarządzania bazą danych) - Jest to system informatyczny służący do zarządzania bazą danych. Z systemem bazy danych współpracują programy użytkowników, zwane aplikacjami. Zadaniem tych programów jest przetwarzanie danych (wstawianie nowych danych, modyfikowanie danych już istniejących, usuwanie danych nieaktualnych, wyszukiwanie danych).

SCHEMAT


BAZA DANYCH
SYSTEM ZARZĄDZANIA BAZĄ DANYCH
APLIKACJA
APLIKACJA
APLIKACJA
APLIKACJA

CHARAKTERYSTYKA BAZ DANYCH

Trwałość danych

- Czas życia (wiele lat)
- Niezależność od warstwy aplikacji

Rozmiar wolumenu danych

- Ze względu na ilość danych wymagana jest pamięć zewnętrzna (dane nie mieszczą się w pamięci operacyjnej)
- Trudność w liniowym dostępie do danych (danych jest zbyt dużo)

CHARAKTERYSTYKA BAZ DANYCH

Złożoność danych

- Złożoność strukturalna i złożoność zależności pomiędzy danymi
- Złożoność semantyczna
- Ograniczenia integralnościowe

WYMAGANIA

Od baz danych wymaga się gwarancji bezpiecznego przechowywania prawdziwych danych oraz umożliwienia dotarcia do tych danych osobom uprawnionym w możliwie najkrótszym czasie. W praktyce definiowane są następujące szczegółowe wymagania:

- Spójność bazy danych
- Efektywne przetwarzanie danych
- Poprawne modelowanie świata rzeczywistego
- Autoryzacja dostępu do danych
- Współbieżność dostępu do danych
- Metadane

SPÓJNOŚĆ BAZY DANYCH

Warunki poprawności danych:

- wierne odzwierciedlenie danych rzeczywistych
- spełnienie ograniczeń nałożonych przez użytkowników
- odporność na anomalie będące wynikiem współbieżności dostępu do baz danych, odporność na błędy, awarie i inne anormalne sytuacje wynikające z zawodności środowiska sprzętowo-programowego, odporność na błędy użytkowników.

EFEKTYWNE PRZETWARZANIE DANYCH

Warunki efektywnego przetwarzania danych:

- efektywne metody dostępu do danych
- optymalizacja metod dostępu do danych
- niezależność aplikacji od fizycznych metod dostępu

POPRAWNE MODELOWANIE ŚWIATA RZECZYWISTEGO

Warunki poprawnego modelowania świata rzeczywistego:

- wspomaganie procesu projektowania i utrzymania bazy danych
- różne poziomy modelowania danych
- transformacje między modelami danych

AUTORYZACJA DOSTĘPU DO DANYCH

Autoryzacja dostępu do danych (wymagania):

- użytkownicy z hasłami dostępu
- użytkownicy i ich uprawnienia

WSPÓŁBIEŻNOŚĆ DOSTĘPU DO DANYCH

Współbieżność dostępu do danych (warunki):

- równoczesny dostęp do tych samych danych przez wielu użytkowników
- konflikt odczyt-zapis, zapis-zapis

METADANE

- dane o danych, strukturach dostępu, użytkownikach i ich prawach

TECHNOLOGIE BAZ DANYCH

Fizyczne struktury danych i metody dostępu

- pliki uporządkowane, pliki haszowe, zgrupowane, indeksy drzewiaste i bitmapowe
- metoda połowienia binarnego, haszowanie statyczne i dynamiczne, metody połączenia, sortowanie, grupowanie
- składniowe i kosztowe metody optymalizacji dostępu
- fizyczna niezależność danych

TECHNOLOGIE BAZ DANYCH

Przetwarzanie transakcyjne (spójność baz danych)

- dostęp do bazy danych za pomocą transakcji o własnościach ACID
- metody synchronizacji transakcji (2PL, znaczniki czasowe, wielowersyjność danych)
- metody odtwarzania spójności bazy danych (plik logu, odtwarzanie i wycofywanie operacji, Write Ahead Log, punkty kontrolne)
- archiwizacja bazy danych i odtwarzanie po awarii

TECHNOLOGIE BAZ DANYCH

Modele danych

- modele pojęciowe (model związków-encji, UML)
- modele logiczne (relacyjny, obiektowy, obiektowo-relacyjny, semistrukturalny, hierarchiczny, sieciowy)

Narzędzia programistyczne

- języki budowy aplikacji
- narzędzia modelowania i projektowania
- metodyki projektowania

SYSTEM ZARZĄDZANIA BAZĄ DANYCH (SZBD)

Oprogramowanie zarządzające całą bazą danych

Funkcjonalność

- język bazy danych - tworzenie, definiowanie, wyszukiwanie i pielęgnacja danych w bazie danych
- struktury danych - efektywne składowanie i przetwarzanie dużych wolumenów danych
- optymalizacja dostępu do danych
- współbieżny dostęp do danych
- zapewnienie bezpieczeństwa danych zagrożonego awaryjnością środowiska sprzętowo-programowego
- autoryzacja dostępu do danych
- wielość interfejsów dostępu do bazy danych

MODEL DANYCH

Modele danych:

- kartotekowy
- hierarchiczny
- sieciowy
- relacyjny
- obiektowy
- obiektowo-relacyjny
- semistrukturalny.

MODEL DANYCH

Modele danych:

- kartotekowy - dane przechowywane są w tablicach (jednej lub wielu) w postaci rekordów o jednakowej strukturze. Rekordy mogą być sortowane, filtrowane. Każda tablica jest samodzielnym dokumentem, bez możliwości współpracy z innymi. Przykładem może być książka telefoniczna w telefonie komórkowym, a także arkusze Excela lub tabele wykonane w MS Word.
- hierarchiczny – dane są uporządkowanie w strukturze drzewa, na zasadzie rekordów nadrzędnych i podrzędnych. Każdy rekord (z wyjątkiem głównego – korzenia) przechowuje wskaźnik do jednego rekordu nadrzędnego. Wyszukiwanie danych w bazie hierarchicznej polega na poszukiwaniu rekordów podrzędnych względem znanego. Przykładem logicznym jest drzewo genealogiczne, a na co dzień z bazą danych zbudowaną w oparciu o ten model, mamy do czynienia w strukturze plików dyskowych, w systemach operacyjnych komputerów.

MODEL DANYCH

Modele danych:

- sieciowy – ten model danych stanowi pewną modyfikację modelu hierarchicznego, w której zamiast wskaźnika do rekordu nadrzędnego, każdy rekord posiada oznaczenie przynależności do kolekcji pewnego typu. Kolekcja to złożony typ, zawierający odniesienia do innych rekordów określonego typu. Zdefiniowanie typu kolekcji polega na podaniu typu rekordu „właściciela” i typu rekordów - elementów kolekcji. Struktura logiczna danych w tym modelu może zostać zobrazowana grafem.
- relacyjny - Model relacyjny zbudowany jest w oparciu o matematyczne pojęcie relacji, a jego struktura logiczna, to tabele o określonej strukturze i rekordy zapisane w tabelach

MODEL DANYCH

Modele danych:

- obiektowy - W modelu obiektowym obiekty w bazie danych reprezentują obiekty w świecie rzeczywistym, oraz posiadają tożsamość. Typ obiektowy (klasa), definiuje złożony typ danych, mogący zawierać w sobie inne typy obiektowe lub ich kolekcje. Struktury danych cechuje enkapsulacja (hermetyzacja), czyli ukrycie samej struktury obiektu przed użytkownikiem, przy czym znajomość tej struktury nie jest konieczna do korzystania z niej. Obiekty dziedziczą zarówno strukturę, jak też właściwości i metody swojej klasy.

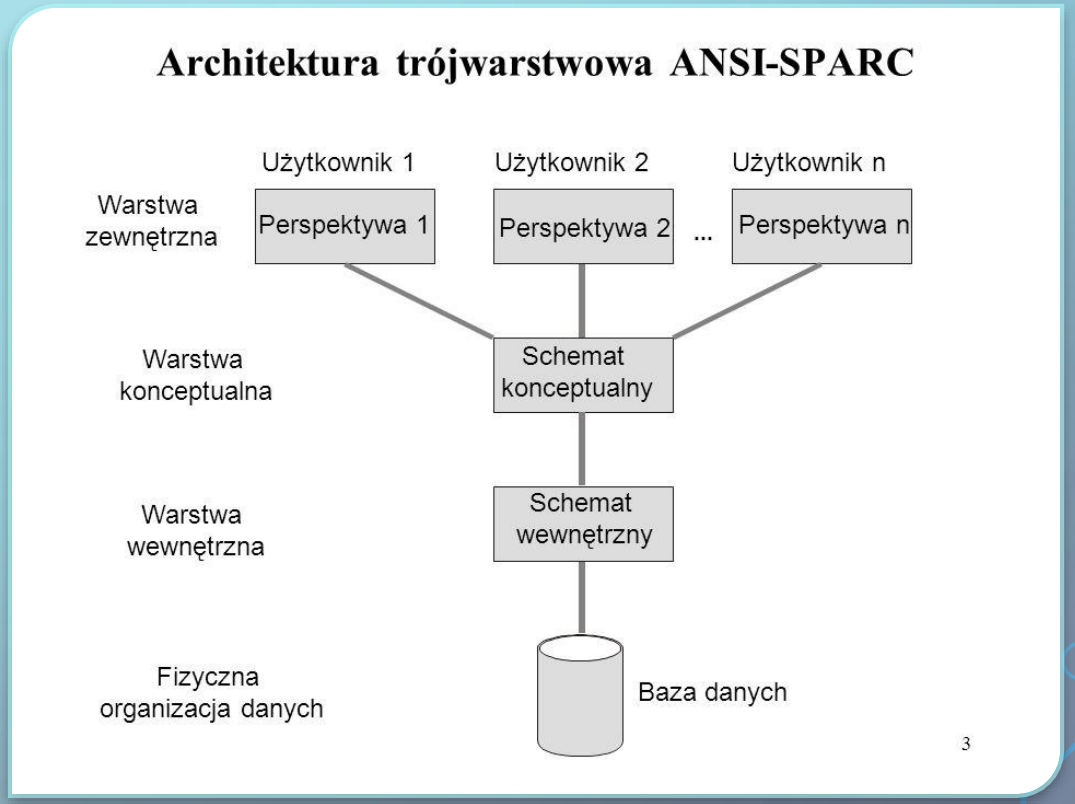
MODEL DANYCH

SIECI SEMANTYCZNE (SEMANTIC WEB)

Projekt, w którego przypadku trudno już mówić o modelu danych. Zakłada się powstanie technologii i standardów pozwalających maszynom i programom na wyszukiwanie i przetwarzanie danych w oparciu o ich znaczenie (semantykę). Jako „bazę danych” przyjmuje się Internet i wszystkie dane w nim dostępne.

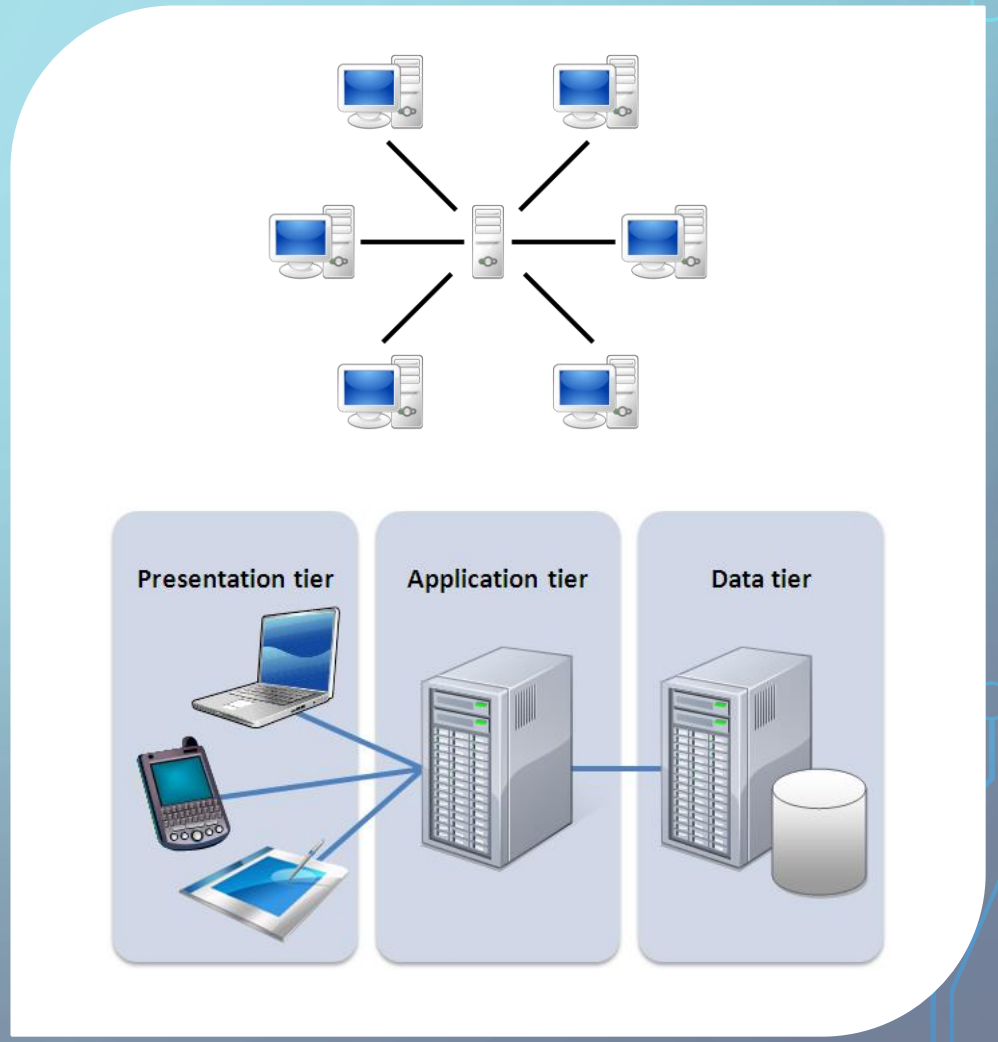
ARCHITEKTURA SBD

ANSI-SPARC - abstrakcyjny standard projektowania systemów zarządzania bazami danych w której wyróżnia się 3 następujące schematy: wewnętrzny, konceptualny, zewnętrzny.



ARCHITEKTURA KOMUNIKACYJNA

- Architektura klient-serwer
- Architektura trójwarstwowa



UŻYTKOWNICY

Użytkownicy SBD:

- Użytkownicy końcowi
- Programiści aplikacji
- Projektanci baz danych
- Analitycy systemowi
- Administratorzy systemów baz danych

INTERAKCJA Z BAZĄ DANYCH

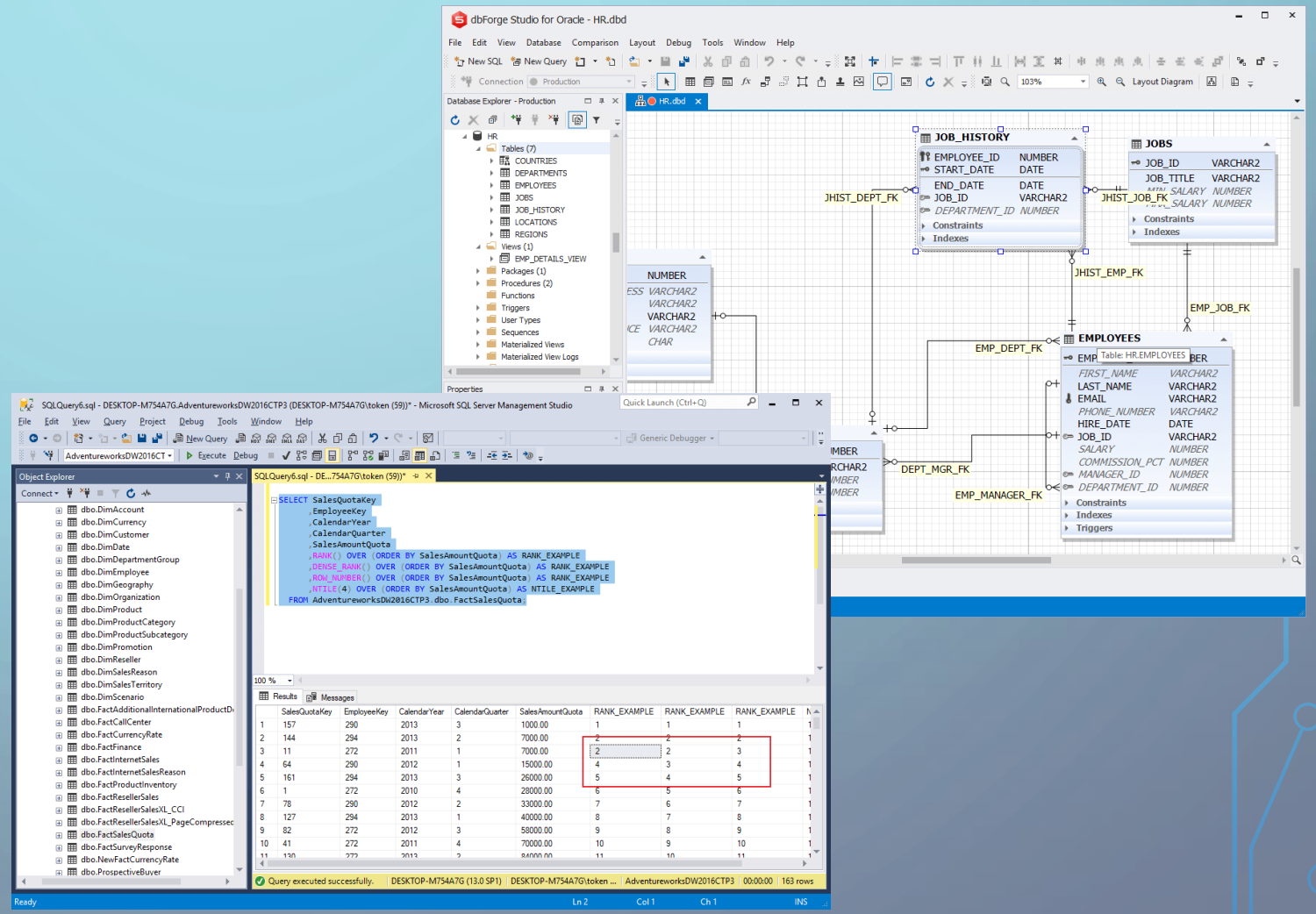
Język SQL - strukturalny język zapytań używany do tworzenia, modyfikowania baz danych oraz do umieszczania i pobierania danych z baz danych (różnice w strukturze języka MSSQL, MySQL, ORACLE ?)

Aplikacje – pozwalają na dostęp do danych z wykorzystaniem formularzy i raportów.

Technologie implementacyjne aplikacji : C, C++, Visual Basic, Visual C++, SAS 4GL, Oracle Forms, Java, PHP, Perl.

SYSTEMY ZARZĄDZANIA BAZĄ DANYCH

- DB2
- Informix Dynamic Server
- Firebird
- MariaDB
- **Microsoft SQL Server**
- MySQL
- **Oracle**
- PostgreSQL



ETAPY TWORZENIA BAZY DANYCH

- Ustalenie wymagań odbiorcy
- Modelowanie konceptualne
- Modelowanie logiczne
- Modelowanie fizyczne
- Realizacja bazy danych

METODY PROJEKTOWANIA

a) Diagram związków encji (diagram ERD)

- Metoda diagramu związków encji nie wymaga zgromadzenia danych przed przystąpieniem do projektowania bazy danych
- Dane są wprowadzane do bazy po jej zaprojektowaniu
- Projektowanie polega na analizie rzeczywistości

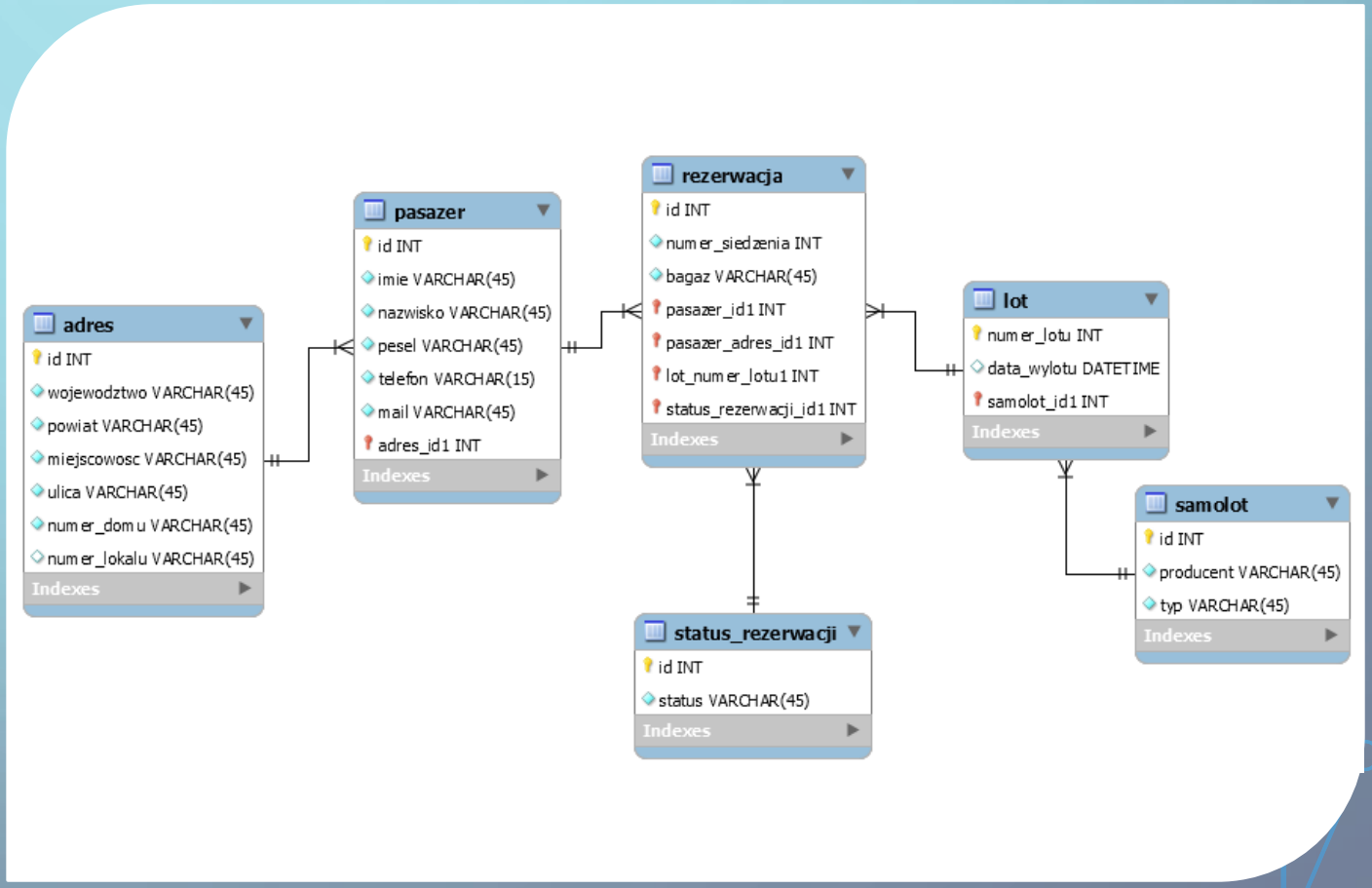
b) Alternatywną metodą jest normalizacja, która wymaga zidentyfikowania całego zbioru danych przed przystąpieniem do projektowania struktury bazy danych. Celem projektowania jest podzielenie zbioru danych na logiczne grupy.

- Metoda stosowana, gdy przed przystąpieniem do projektowania mamy już zgromadzone dane np. w arkuszach kalkulacyjnych MS Excell lub w formie ksiąg papierowych
- Projektowanie polega na analizie zgromadzonych danych

Obie metody powinny prowadzić do tego samego wyniku, a przynajmniej do podobnego.

METODA DIAGRAMÓW ZWIĄZKÓW ENCJI

- Rozpoznanie encji występujących w obszarze analizy
- Określenie związków zachodzących pomiędzy encjami
- Stworzenie diagramu związków encji
- Przekształcenie diagramu w schemat relacyjnej bazy danych



ENCJA

- Encja – pewien aspekt „świata rzeczywistego”, który istnieje niezależnie i może być jednoznacznie zidentyfikowany
- Encją może być rzecz, która w organizacji jest rozpoznawana jako rzecz istniejąca niezależnie i unikatowo zidentyfikowana
- Encją może być nie tylko rzecz istniejąca w przestrzeni ale i zdarzenie istniejące w czasie (np. sprzedaż) lub pojęcie (np. zamówienie, zaliczenie, dzierżawa)
- Jako aspekt świata rzeczywistego encja jest scharakteryzowana pewną liczbą właściwości – atrybutów
- Encja jest abstrakcją złożoności pewnej dziedziny
- Jeśli trzeba przechowywać dane na temat wielu właściwości jakiejś rzeczy, to taka rzecz jest prawdopodobnie encją

PRZYKŁADY ENCJI

- Encja klient posiadająca atrybuty: nazwisko, imię, nr telefonu, adres, rabat itp.
- Encja towar posiadająca atrybuty: nazwa, producent, jakość, cena itp.
- Encja sprzedaż posiadająca atrybuty: klient, towar, ilość, data itp.

Tworzenie diagramu:

- Określenie identyfikatorów encji
- Związki encji (liczebność związku)

PRZEKSZTAŁCENIE DIAGRAMU W SCHEMAT RELACYJNEJ BAZY DANYCH

- Dla każdej encji tworzona jest tabela
- Identyfikatory encji stają się kluczami głównymi tabel
- Atrybuty stają się atrybutami relacji (nagłówkami kolumn tabeli)
- Związki jeden do wiele realizowane są za pomocą klucza obcego w tabeli po stronie wiele
- Związek jeden do jeden można zrealizować za pomocą klucza obcego plus ograniczenie UNIQUE po stronie wiele
- Opcjonalność po stronie wiele można kontrolować dodając lub nie ograniczenie NOT NULL
- Związek rekurencyjny jeden do wiele przekształca się do tabeli z kluczem obcym odwołującym się do klucza głównego tej samej tabeli

NORMALIZACJA

Normalizowanie to proces porządkowania danych w bazie danych. Obejmuje to tworzenie tabel i ustanawianie relacji między tymi tabelami zgodnie z regułami zaprojektowanymi zarówno w celu ochrony danych, jak i w celu zapewnienia większej elastyczności bazy danych przez wyeliminowanie nadmiarowości i niespójnej zależności.

POSTACIE NORMALNE

1NF - wszystkie elementy są atomowe i żadne wiersze się nie powtarzają,

2NF - 1NF + atrybuty niekluczowe nie mogą być zależne od części klucza złożonego,

3NF - 2NF + wszystkie atrybuty niekluczowe zależą od klucza wyłącznie bezpośrednio (nie przechodnio),

BCNF - 3NF + każdy atrybut, od którego w pełni zależy inny atrybut, jest kluczem kandydującym,

4NF - BCNF + nie występują zależności wielowartościowe.

PIERWSZA POSTAĆ NORMALNA (1NF)

Reguły które musi spełniać tabela, aby była w pierwszej postaci normalnej:

- Każdy jej element powinien reprezentować pojedynczą wartość (np. tylko jeden numer PESEL),
- Nie mogą w niej występować dwa jednakowe wiersze,
- Wartości w danej kolumnie powinny być z tej samej domeny (tzn. powinny być tego samego typu/rodzaju),
- Nie ma żadnego warunku narzucającego kolejność jej wierszy i kolumn.

PIERWSZA POSTAĆ NORMALNA (1NF)

WorkerID	Name	ProgLang1	ProgLang2
1	Anon One	C++	Java
2	Anon Two	Javascript	Python

Przechowywanie informacji o językach programowania w dwóch kolumnach:

- Kolumny ProgLang1 i ProgLang2 mają tę samą dziedzinę
- W tabeli istnieje ograniczenie do dwóch języków na programistę
- Nastąpią komplikacje przy wyszukiwaniu programistów konkretnego języka
- Nastąpi potencjalne narzucenie kolejności kolumn (nazwy języków posortowane alfabetycznie)

PIERWSZA POSTAĆ NORMALNA (1NF)

WorkerID	Name	ProgLang
1	Anon One	C++, Java
2	Anon Two	JavaScript, Python

Przechowywanie informacji o wszystkich językach programowania w pojedynczej kolumnie:

- Naruszenie reguły atomowości danych,
- Trudności w wydobyciu właściwych informacji (dwa ciągi znaków w komórce ProgLang)
- Przy założeniu że dziedziną dla ProgLang jest zestaw języków programowania, tabela teoretycznie jest w 1NF.

PIERWSZA POSTAĆ NORMALNA (1NF)

Przechowywanie informacji o językach programowania

w większej liczbie wierszy

- Tabela jest w 1NF
- Redundancja danych (powtarzanie wartości w kolumnie Name)

WorkerID	Name	ProgLang
1	Anon One	C++
1	Anon One	Java
2	Anon Two	JavaScript
2	Anon Two	Python

PIERWSZA POSTAĆ NORMALNA (1NF)

PROGRAMMERS	
WorkerID	**Name**
1	Anon One
2	Anon Two

PROG_LANGUAGES	
WorkerID	**ProgLang**
1	C++
1	Java
2	JavaScript
2	Python

Podział danych na dwie tabele

- Obie tabele są w 1NF,
- Powtarzane są jedynie wartości w kolumnie WorkerID tabeli PROG_LANGUAGES (powtarzanie liczb zamiast ciągów znaków).

DRUGA POSTAĆ NORMALNA (2NF)

Aby tabela była w drugiej postaci normalnej (2NF), musi ona spełniać dwie reguły:

- musi ona być w pierwszej postaci normalnej (1NF)
- wszystkie jej atrybuty niekluczowe muszą być w pełni zależne od każdego z kluczy kandydujących (jeśli w tabeli występują klucze złożone, to atrybuty niekluczowe muszą być zależne od całych kluczy, a nie tylko ich części)

DRUGA POSTAĆ NORMALNA (2NF)

PARTICIPANTS					
IDENT	**NAME**	**CITY**	**INHAB**	**COURSE**	**GRADE**
P1	Collins	London	8000000	English	A
P1	Collins	London	8000000	Geography	C
P1	Collins	London	8000000	Logics	A
P2	Jones	Glasgow	400000	Geography	B
P2	Jones	Glasgow	400000	Databases	C
P3	Rodin	Aberdeen	400000	Physics	B
P4	Thatcher	London	8000000	Logics	A
P4	Thatcher	London	8000000	Chemistry	C
P5	Biggs	Bristol	800000	Databases	A
P5	Biggs	Bristol	800000	English	A
P5	Biggs	Bristol	800000	Biology	A

DRUGA POSTAĆ NORMALNA (2NF)

PARTICIPANTS

IDENT	NAME	CITY	INHAB	COURSE	GRADE
P1	Collins	London	8000000	English	A
P1	Collins	London	8000000	Geography	C
P1	Collins	London	8000000	Logics	A
P2	Jones	Glasgow	400000	Geography	B
P2	Jones	Glasgow	400000	Databases	C
P3	Rodin	Aberdeen	400000	Physics	B
P4	Thatcher	London	8000000	Logics	A
P4	Thatcher	London	8000000	Chemistry	C
P5	Biggs	Bristol	800000	Databases	A
P5	Biggs	Bristol	800000	English	A
P5	Biggs	Bristol	800000	Biology	A

Nadmiarowość danych w tabeli PARTICIPANTS może prowadzić do różnych problemów:

- Aktualizacja liczby mieszkańców danego miasta musi być dokonywana w wielu różnych wierszach jednocześnie, aby baza nie utraciła integralności (anomalia aktualizacji)
- W niektórych przypadkach, usunięcie informacji o danym kursancie może jednocześnie spowodować usunięcie informacji o liczbie mieszkańców miasta z którego on pochodzi (anomalia usuwania)
- Wstawienie informacji o nowym kursie ukończonym przez danego kursanta wymaga powtórzenia informacji dotyczących miasta z którego on pochodzi (anomalia wstawiania)

DRUGA POSTAĆ NORMALNA (2NF)

PART_DATA			
IDENT	**NAME**	**CITY**	**INHAB**
P1	Collins	London	8000000
P2	Jones	Glasgow	400000
P3	Rodin	Aberdeen	400000
P4	Thatcher	London	8000000
P5	Biggs	Bristol	800000

PART_COURSE		
IDENT	**COURSE**	**GRADE**
P1	English	A
P1	Geography	C
P1	Logics	A
P2	Geography	B
P2	Databases	C
P3	Physics	B
P4	Logics	A
P4	Chemistry	C
P5	Databases	A
P5	English	A
P5	Biology	A

Rozwiązanie to podział danych na dwie tabele

- Wyeliminowanie niepełnej zależności funkcyjnej

TRZECIA POSTAĆ NORMALNA (3NF)

Aby tabela była w trzeciej postaci normalnej (3NF), musi ona spełniać następujące reguły:

- musi ona być w drugiej postaci normalnej (2NF), zatem musi ona spełniać także reguły 1NF
- żaden atrybut niekluczowy nie może zależeć przechodnio od żadnego z kluczy kandydujących (tj. wszystkie atrybuty wtórne zależą bezpośrednio od wszystkich kluczy kandydujących)

TRZECIA POSTAĆ NORMALNA (3NF)

PART_ID

IDENT	NAME	CITY
P1	Collins	London
P2	Jones	Glasgow
P3	Rodin	Aberdeen
P4	Thatcher	London
P5	Biggs	Bristol

CITIES

CITY	INHAB
London	8000000
Glasgow	400000
Aberdeen	400000
Bristol	800000

PART_COURSE

IDENT	COURSE	GRADE
P1	English	A
P1	Geography	C
P1	Logics	A
P2	Geography	B
P2	Databases	C
P3	Physics	B
P4	Logics	A
P4	Chemistry	C
P5	Databases	A
P5	English	A
P5	Biology	A

POSTAĆ NORMALNA BOYCE'A CODDA

Aby tabela była w postaci normalnej Boyce'a Codda (BCNF), musi ona spełniać następujące reguły: musi ona być w trzeciej postaci normalnej (3NF); każdy atrybut w tabeli, od którego w pełni funkcyjnie zależy inny atrybut, musi być kluczem kandydującym

CZWARTA POSTAĆ NORMALNA (4NF)

Aby tabela była w czwartej postaci normalnej (4NF), musi ona spełniać następujące reguły: musi ona być w trzeciej postaci normalnej (3NF) lub w postaci normalnej Boyce'a Codda (BCNF); nie może ona zawierać zależności wielowartościowych

DZIĘKUJĘ ZA UWAGĘ