**Baza danych** jest zbiorem danych oraz narzędzi sytemu zarzadzania bazą danych DBMS (database managemet system) przeznaczonego do zarzadzania bazą danych oraz gromadzenia, przekształcania i wyszukiwania danych.

**Cechy charakterystyczne bazy danych:**

* **Trwałość danych** – oznacza możliwość przechowywania danych w pamięci masowej komputera
* **Niezależność danych** – pozwala osiągnąć większą elastyczność, ponieważ programy wymieniające informacje z baza danych są niezależne od przechowywania danych na dysku i szczegółów reprezentacji danych na dysku
* **Ochrona danych** – baza danych oferuje mechanizm kontroli dostępu do danych w sposób umożliwiający użytkowanie danych wyłącznie przez uprawnionych użytkowników
* **Integralność danych** – zgodność danych z rzeczywistością

**Funkcje DBMS:**

* **Tworzenie struktur bazy danych**
* **Wykonywanie operacji na bazie** (CRUD, Create Read Update Delete)
* **Obsługa zapytań** (kwerendy)
* **Generowanie raportów**
* **Administrowanie baza danych** (np. zarzadzanie użytkownikami, nadawanie uprawnień, tworzenie i odtwarzanie backup’ow).

**Model danych** – to zintegrowany zbiór zasad opisujących dane, relacje pomiędzy nimi, dozwolone operacje i ograniczenia nakładane na dane.

Model danych jest próbą reprezentacji świata realnego i występujących w nim obiektów, zdarzeń i związków zachodzących między nimi. Model danych można opisać jako konstrukcje składające się trzech komponentów:

* **Części strukturalnej** – składającej się z reguł określających budowę bazy danych
* **Części manipulacyjnej** – określającej które operacje można wykonywać na danych
* **Części zawierającej reguły integralności** – gwarantującej stabilność działania systemu

**Rodzaje modeli bazy danych:**

1. **Model jednorodny** – to model, w którym wszystkie dane są umieszczone w jednej tabeli. Cechuje go łatwość odczytywania i zapisu danych. Jego wadą jest duża liczba duplikatów, których przetwarzanie spowalnia pracę systemu i zwiększa zużycie zasobów komputera
2. **Model hierarchiczny** – przypomina strukturę odwróconego drzewa. Istnieje jeden korzeń (tabela nadrzędna) oraz synowie (tabela podrzędna). Jeden ojciec może mieć wiele dzieci, ale każde dziecko ma tylko jednego ojca. Powiązanie nadrzędny podrzędny to związek jeden do wielu. Model ten różni się od relacyjnego, ponieważ w modelu relacyjnym powiązania zachodzą przez klucze obce, a w hierarchicznym związek nadrzędny podrzędny.
3. **Model sieciowy** – przyjmuje podobnie jak hierarchiczny strukturę odwróconego drzewa z tą różnicą, że gałęzie jednego drzewa mogą być wspólne z gałęziami innych drzew.
4. **Model relacyjno-obiektowy** – model ten pozwala w relacyjnych tabelach tworzyć kolumny, w których przechowane są dane typu obiektowego. Pozwala na definiowanie zmiennych oraz metod, które będą wykonywane na danych wprowadzonych do obiektu.
5. **Model obiektowy** – opiera się na koncepcji obiektów. Odwołania do określonego obiektu są wykonywane za pomocą interfejsu, dzięki któremu są zachowane integralność i bezpieczeństwo danych
6. **Model relacyjny** – podstawą tego modelu jest pojęcie relacji, czyli powiązania pomiędzy dwoma tabelami. Dane przechowuje się w tabelach zwanych relacjami składających się z wierszy (czyli krotek) oraz kolumn (atrybutów lub polem)

**Integralność danych** określana jest również mianem spójności danych jest to funkcja systemu zarzadzania bazą danych, która gwarantuje, że dane nie zostaną usunięte lub zmienione w nieautoryzowany sposób. Ochrona integralności polega również na zapewnieniu, że dane nie ulegną zniekształceniu podczas wykonywania na nich operacji. Zachowanie poprawności bazy danych opiera się na utrzymaniu poprawności w obrębie semantycznym, encji i referencyjnym.

* **Integralność semantyczna** – polega na utrzymaniu ograniczeń nakładanych na dane m.in. dane muszą być zgodne z typem pola tabeli do którego są wpisywane
* **Integralność encji** – zgodnie z regułami integralności encji każda tabela powinna posiadać Klucz Główny (Klucz Podstawowy). Kolumnę, w której dane nie mogą się powtarzać i w której nie może wystąpić wartość NULL (typ pusty)
* **Integralność referencyjna** – polega na wprowadzeniu i utrzymaniu powiązań między tabelami. Związki te tworzy się poprzez umieszczenie kolumny pełniącej role Klucza Głównego w drugiej tabeli co nadaje kolumnie funkcje Klucza Obcego, relacja zachodzi pomiędzy Kluczem Głównym w jednej tabeli i Kluczem Obcym w drugiej.

**System zarządzania bazą danych (DBMS)** to oprogramowanie umożliwiające użytkownikom definiowanie, tworzenie i zarządzanie bazą danych oraz kontrolowanie dostępu do nich.

DBMS pełnią funkcję pośrednika pomiędzy danymi a użytkownikiem chcącym z tych danych skorzystać. Oprócz organizacji przechowywania danych zapewniają również odpowiednią ochronę zarówno przed nieprawidłowym działaniem programu jak i przed dostępem ze strony nieuprzywilejowanych aplikacji. Funkcja taka nazywa się ochroną danych. System DBMS zabezpiecza również, aby dane były logicznie spójne i odpowiadały rzeczywistości. Funkcja ta nazywa się ochroną spójności. DBMS daje również możliwość tworzenia kopii zapasowych zarządza nimi oraz odtwarzania bazy danych po awarii.

**Encja** – określenie obiektu fizycznego lub grupy obiektów (np. osoba, książka, samochód) o którym informacje przechowujemy w bazie danych

**Atrybut encji** – cechy charakterystyczne opisujące obiekt (np. imię, nazwisko, klasa itp.)

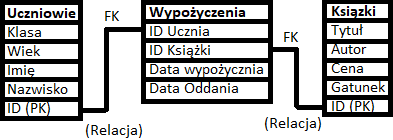
**Krotka** – jest wierszem tabeli (rekordem)

**Relacja** – jest to tabela, która spełnia wymogi relacji

**Atrybut** – jest definiowany jako kolumna relacji mająca identyfikator (nazwę)

**Klucz Główny** (Primary Key) – jest to atrybut tabeli (relacji) który został wybrany, aby unikatowo identyfikować krotki tabeli

**Klucz Obcy** – jest to atrybut (kolumna) w jednej tabeli, która pasuje do Klucza Podstawowego w drugiej tabeli. Wykorzystywany jest do tworzenia związków pomiędzy tabelami

****

**Projektowanie konceptualne** to proces konstrukcji modelu danych używany niezależnie od fizycznej implementacji. Rzeczywistość oddawana przez model konceptualny nie jest ograniczona do reguł implementacji, rodzaju DBMS oraz aplikacji i programów.

**Projektowanie logiczne** to proces konstrukcji modelu danych oparty na specyficznym modelu, ale niezależny od DMBS.

**Model fizyczny** to proces implementacji bazy danych, opisujący relacje, organizacje plików, indeksy, dostęp do danych, ograniczenia i stosowanie reguł bezpieczeństwa

**Proces projektowania logicznego można podzielić na etapy:**

1. Przygotowanie diagramu związków encji
2. Normalizacja
3. Wprowadzenie zasad wymuszających integralności danych

**Zależności funkcyjne**

Zależnością funkcyjną jest przedstawienie relacji R o dwóch atrybutach A i B w której atrybut B jest zależny od A gdy we wszystkich krotkach zawierających wartość A wystąpi tylko jedna odpowiadająca jej wartość B. Jednak dla danej wartości B może wystąpić kilka różnych wartości A.

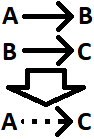
**Pełna zależność funkcyjna** –atrybuty B są w pełni funkcyjnie zależne od atrybutów A, ale nie jest funkcyjnie zależny od żadnego właściwego podzbioru A.

**Oznaczenie:**

Atrybut B w pełni zależny od A

**Przechodnia zależność funkcyjna** – z przechodnią zależnością funkcyjną mamy do czynienia, gdy A nie jest bezpośrednim wyznacznikiem C. Atrybut A determinuje funkcyjnie od niego zależny atrybut B, natomiast atrybut B determinuje funkcyjnie zależny atrybut C.

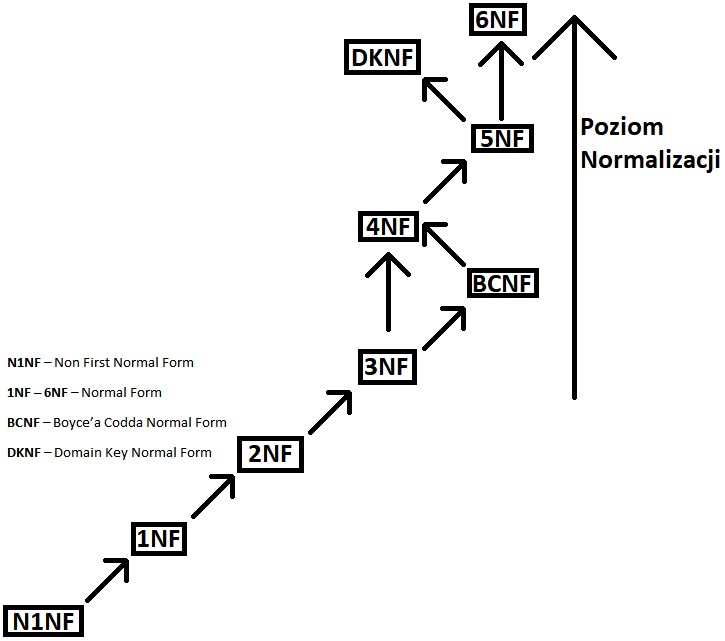
**Oznaczenie:**



**Normalizacja** jest procesem mającym na celu redukowanie anomalii, zachowań relacyjnych baz danych poprzez tworzenie mniejszych tabel opartych na dobrze zaprojektowanych relacjach.

**Celem normalizacji** jest również izolowanie danych. Pojęcie to oznacza, że modyfikacje danych (usunięcie, dodanie informacji) odbywa się w jednym polu skąd zmiana przenoszona jest za pomocą zdefiniowanych związków do pozostałych tabel baz danych.

**Kierunek normalizacji relacyjnej bazy danych:**



Powyższy schemat przedstawia kolejne etapy normalizacji bazy danych. W powszechnie przyjętych praktykach baza danych sprowadzana jest do 3 postaci normalnej uznana jest za znormalizowaną.

**1NF powinna spełnić następujące kryteria:**

* Rozpatrywane z góry na dół wiersze (krotki) nie muszą być ustawione w określonym porządku (kolejność wierszy dowolna)
* Kolejność kolumn dowolna
* Wiersze tabeli nie mogą się powtarzać
* Każdy wiersz i każda kolumna na swoim przecięciu zawierają dokładnie jedną wartość z zakresu swojej dziedziny i nic ponad to
* Wszystkie kolumny są symetryczne i równomierne (tzn., że w obrębie wierszy nie ma dodatkowych ukrytych składników takich jak mniejsze kolumny, podkolumny itp.)

**2NF powinna spełniać następujące warunki:**

* Tabela, która znajduje się w 2NF musi spełniać wymogi 1NF
* Wszystkie kolumny zawierające elementy nienależące do klucza głównego musza od niego zależeć i być przez niego jednoznacznie identyfikowane
* Aby doprowadzić tabele do drugiej postaci normalnej, należy usunąć kolumny (???)

**3NF tabela jest zgodna z 3NF, gdy:**

* Znajduje się w 2NF to znaczy, że każda kolumn nie wchodzi w skład klucza głównego jest zależna od niego
* Każda kolumna nie wchodząca w skład klucza głównego nie może zależeć od innej kolumny nie wchodzącej w skład klucza głównego

**Doprowadzenie tabel do 3NF** uzyskujemy zwykle za pomocą tworzenia tabel słownikowych oraz tabel łącznikowych (tabele łącznikowe używane są przy budowaniu relacji typu wiele do wielu).

**Wnioski:**

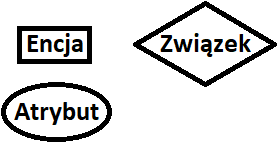
* Normalizacja polega na przeprowadzeniu tabel przez kolejne postacie normalne
* Znormalizowana baza danych jest bardziej elastyczna pod kątem zapytań

**4NF** po w prowadzeniu postaci normalnej Boyce’a Codda okazało się, że istnieją przypadki które sprawiają, że w bazie danych zdarzają się **retundancje** (nadmierność informacji – powtarzanie się tych samych danych w więcej niż jednym rekordzie) Postacią normlaną silniejszą od BCNF nazwano czwartą postacią normlaną.

**5NF** występuje, kiedy relacja nie zawiera zależności złączeniowych.

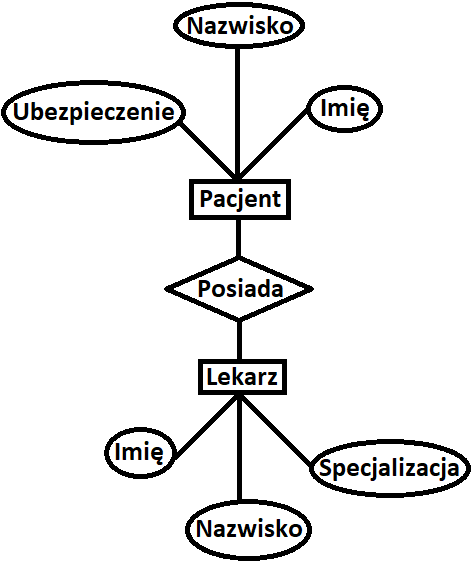
**6NF** relacja R jest w 6NF, jeżeli nie posiada (nie posiada i nie będzie posiadać) żadnych trywialnych zależności złączeniowych.

**Bloki diagramu ER**

****

**Diagramy ER** (Chena, Entity Relatioship) są abstrakcyjno pojęciowo reprezentacją danych i pozwalają w sposób graficzny przedstawić model projektu bazy danych i związków zachodzących pomiędzy encjami.

**Przykładowy diagram**

****

**Typy związków:**

* 1:1
* 1:M (jeden do wielu)
* M:N (wiele do wielu)

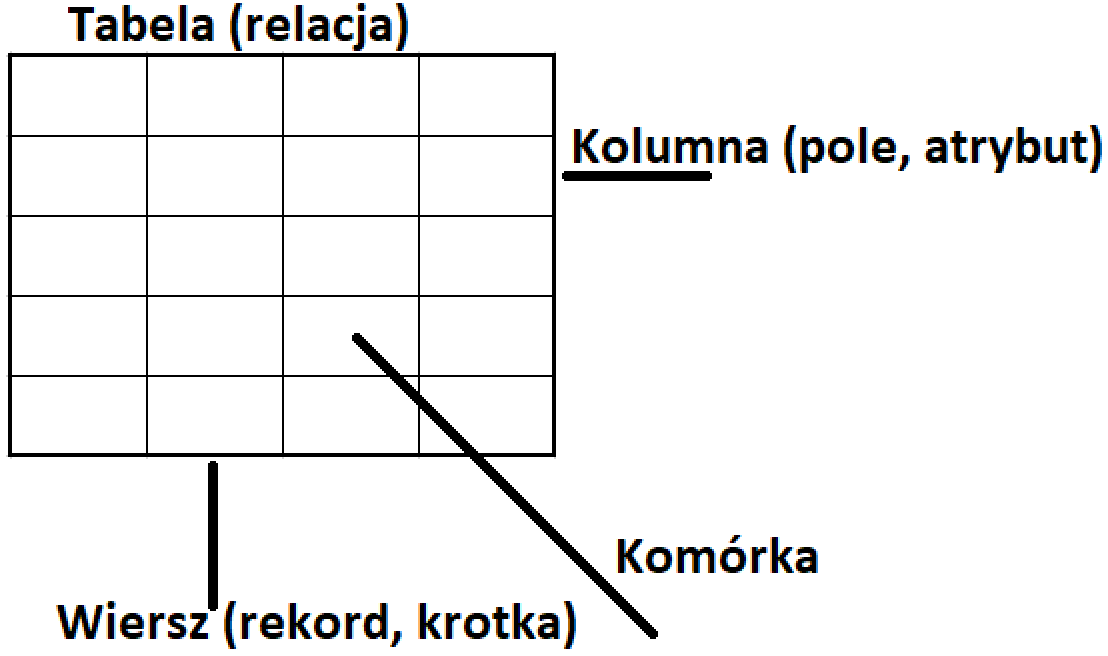
**Związek jeden do jednego (1:1)** – występuje wówczas, gdy jednej instancji zbioru A np. każdemu kierowcy odpowiada jeden samochód, a każdemu samochodowi odpowiada jeden kierowca

**Związek jeden do wielu (1:M)** – występuje wtedy, gdy jednej instancji zbioru A odpowiada więcej niż jedna instancja zbioru B np. jeden kierownik ma wielu podwładnych, ale jeden podwładny ma jednego kierownika

**Związek wiele do wielu (M:N)** – zachodzi w sytuacji gdy wielu instancjom encji zbioru A odpowiada wielu instancji encji zbioru B np. jeden lekarz może mieć wielu pacjentów i jeden pacjent może być leczony przez wielu lekarzy

**Więzy propagacji** (???) wskazują jakie działanie zostanie podjęte podczas modyfikowania wierszy w tabeli powiązanej. Możliwe przypadki:

* Zablokowanie możliwości usunięcia rekordu do czasu wystąpienia powiązanego z nim wiersza w drugiej tabeli
* Kaskadowe usuwanie występuje, gdy podczas usuwania krotki pozostałe połączone z nią krotki w innych tabelach również zostaną automatycznie usunięte
* Wstawienie (???) wartości NULL

Tabele bazy danych możemy połączyć relacjami. Relacje tworzymy łącząc pola tabeli które muszą być tego samego typu i przechowywać dane z tej samej dziedziny. Relacje tworzymy poprzez połączenie klucza głównego z jednej tabeli z tzw. kluczem obcym z tabeli drugiej. Podczas tworzenia relacji możemy wyświetlić tzw. więzy integralności, które powodują, że dane w tabelach są spójne to znaczy na przykład, że system nie pozwoli nam wypożyczyć książki, która nie jest zdefiniowana w tabeli książka. Nie wymuszając więzów integralności w relacji taka sytuacja mogłaby mieć miejsce.

**Kwerenda** jest zapytaniem do bazy danych realizującym jakąś operacje, najczęściej jest to wybór rekordów do wyświetlenia spełniających określone kryteria. Program Access posiada graficzne narzędzia do tworzenia kwerend, natomiast w rzeczywistości są to polecenia SQL (SQL – Structured Query Language) np.:

SELECT Książka.tytył, Książka.autor, Książka.gatunek

FROM Książka

Where (((Książka.gatunek) like „powieść”));

**Kwerenda jest zapytaniem do bazy danych które może realizować operacje takie jak:**

* Wybór danych spełniających podane kryteria
* Modyfikacja zawartości tabel (dodanie, usunięcie i zmiana rekordów)
* Modyfikowanie struktury tabel (tworzenie, usuwanie i zmiana struktury tabeli)

**Rodzaje kwerend:**

* Kwerenda **wybierająca** (1)
* Kwerenda **funkcjonalna** (2 i 3)

**Kwerendę w programie access możemy tworzyć** korzystając z narzędzia „projekt kwerendy” lub pisząc polecenie języka SQL. Wynikiem działania projektu kwerendy jest automatyczne wygenerowanie polecenia SQL. Realizującego daną kwerendę.

**Przykład polecenia SQL** wyświetlającego informacje o pełnoletnich uczniach z Rzeszowa:

SELECT Uczeń.imię, Uczeń.nazwisko, Uczeń.miejscowość, Uczeń.wiek

FROM Uczeń

WHERE (((Uczeń.miejscowość)=„Rzeszów” AND ((Uczeń.wiek)> 18));

Program Access posiada szereg funkcji wbudowanych, dostępnych w narzędziu konstruktor wyrażeń. Między innymi dostępne są funkcje do wykonywania operacji na dacie.

**Przykładowe wyrażenie**, określenie roku na podstawie daty systemowej:

Year(Date())

**Formularz** jest interfejsem pomiędzy użytkownikiem a programem komputerowym. Formularze w programie Access pozwalają na wykonywanie operacji na bazie danych bez konieczności znajomości jej organizacji i struktury. Formularz składa się z obiektów (w programie nazywanych formantami). Poprzez wykonanie operacji na obiekcie (np. kliknięcie przycisku lub wpisanie tekstu w pole edycyjne). Program wykonuje instrukcje obsługi zapytania. Programy z graficznym interfejsem nie są programami sekwencyjnymi, lecz programami sterowanymi zdarzeniowo.