

1. RODZAJE BAZ DANYCH I ICH BUDOWA

2.1. Pojęcie bazy danych

W codziennym życiu można się spotkać w różnych miejscach i w różnych sytuacjach z przechowywaniem zbiorów informacji w celu ich udostępniania różnym użytkownikom. Informacje przechowywane są za zwyczaj w sposób uporządkowany, co ułatwia ich wyszukiwanie.

Przykładem zbioru informacji jest dziennik lekcyjny zawierający listy uczniów i ich dane personalne. Przy każdym uczniu są gromadzone uzyskane przez niego oceny z różnych przedmiotów oraz informacje o jego obecnościach, udziale w lekcji itp.. Przykładem miejsca gdzie wykorzystuje się różne zbiory informacji może być też biblioteka. Czytelnik korzysta przede wszystkim z katalogu książek. Są w nim podane szczegółowe informacje o każdej książce (grupie identycznych książek). Do identyfikowania konkretnych egzemplarzy książek stosuje się oznaczenia zwane sygnaturami. Jest ponadto spis książek wypożyczonych; przy każdej książce znajduje się informacja, kto tę książkę wypożyczył. Jest też w bibliotece rejestr osób, które mogą z niej korzystać. Przy każdej osobie są zgromadzone informacje o książkach przez nią wypożyczonych.

Informacje zamieszczone w poszczególnych spisach (rejestrach, katalogach) są ze sobą powiązane. W obrębie poszczególnych spisów są one rozmieszczone w określonym porządku, zazwyczaj alfabetycznym lub według numerów.

W trakcie użytkowania zbiory informacji są uzupełniane i korygowane. Z gromadzonych informacji bywają sporządzane raporty dotyczące całości lub części zbioru. Zbiory danych o określonej strukturze, umożliwiającej spełnianie tych funkcji, określa się mianem baz danych. W informatyce pojęcie to odnosi się do zbiorów informacji zapisanych w pamięci zewnętrznej komputera (np. na dysku twardym).¹

Baza danych (angielskie *database*) jest to rodzaj komputerowego zbioru kartotek, magazyn danych o określonej budowie. Baza danych jest modelowym ujęciem fragmentu rzeczywistości będącego przedmiotem zainteresowania (*universe of discours*) osób, instytucji, organizacji, firm, zakładów itp., reprezentującym fakty dotyczące tej rzeczywistości w formie umożliwiającej ich przetwarzanie w komputerze. Istotne obiekty danego przedmiotu zainteresowania określa się jak encje lub klasy. Klasą lub encją w rejestrze samochodów są poszczególne samochody, a także ich właściciele lub użytkownicy. Projekt bazy danych określa jej strukturę (część intensjonalną) i zawartość (część ekstensjonalną). Dane przechowywane w bazie są trwałe, co nie oznacza, że nie ulegają zmianom. W każdej chwili baza danych znajduje się w określonym stanie. Operacje powodujące zmianę stanu bazy danych noszą nazwę transakcji.²

W rozumieniu ustawy z dnia 27 lipca 2001 r. o ochronie baz danych, termin **baza danych** to zbiór danych lub jakichkolwiek materiałów i elementów zgromadzonych według określonej systematyki lub metody, indywidualnie dostępnych w jakikolwiek sposób, w tym środkami elektronicznymi, wymagający istotnego, co do jakości lub ilości, nakładu inwestycyjnego w celu sporządzenia, weryfikacji lub prezentacji jego zawartości.³

¹ Płoszajski G; Elementy informatyki użytkowanie komputera, , wyd. WSiP, Warszawa 1998.

² Płoski Z; Słownik Encyklopedyczny – Informatyka., wyd. Europa., Wrocław 1999.

³ Ustawa z dnia 27 lipca 2001 r. o ochronie baz danych (Dz.U. z 2001 r. Nr 128, poz. 1402)

Baza danych służy, zatem do tworzenia i zarządzania danymi zapisanymi w systemie komputerowym. Prosta baza obejmuje jedną lub kilka tabel, natomiast bardziej rozbudowane bazy posiadają dziesiątki jak nie setki tabel zawierających mnóstwo rekordów. Do identyfikacji rekordów baza wykorzystuje jeden lub kilka indeksów. Dane w bazie są przechowywane w postaci pól i rekordów. Pole w bazie odpowiada pojedynczemu elementowi np. pola (Imię, nazwisko, adres). Komplet pól tworzy rekord. Bazy danych operują głównie na danych tekstowych i liczbowych, lecz większość współczesnych baz umożliwia przechowywanie danych binarnych typu: grafika, muzyka itp.

Cechy poprawnej bazy danych

1. niepodzielność danych pierwotnych - dane przechowywane w jednostkach najniższego rzędu są traktowane jako niepodzielne.
2. brak nadmiarowości - każda informacja jest zapisana dokładnie jeden raz; do aktualizacji jednej informacji wystarczy modyfikacja jednego wpisu do bazy.
3. integralność - informacje przechowywane w bazie nie mogą być wzajemnie sprzeczne.
4. spójność - wszystkie składniki bazy zawierają wyłącznie sensowne wartości. Kryterium sensowności winno być ustalone osobno dla każdego składnika bazy (np. data urodzenia nie powinna być datą z przyszłości, data zgonu nie powinna być wcześniejsza od daty urodzenia).
5. przemyślany układ tabel - pojedyncza tabela bazy danych winna przechowywać dane opisujące tylko jeden „rodzaj zależności” (np. w tabeli danych osobowych nie umieszczamy pola `Imie-psa` — ktoś może nie mieć psa, a ktoś inny może mieć ich więcej). Układ tabel powinien odzwierciedlać relacje — czyli mówiąc z grubsza zależności między danymi — w jednej z ich tzw. *postaci normalnych*. Zgodność taka umożliwia przechowywanie minimalnego zestawu danych i unikanie komplikacji związanych z zarządzaniem danymi.

2.2. Rodzaje baz danych

Biorąc pod uwagę miejsce przechowywania bazy danych, możemy wśród nich wyróżnić:

1.. **Lokalne bazy danych** - są to najprostsze bazy danych, które w całości znajdują się na jednym komputerze np. prosta baza zawierająca spis ludności w danym mieście. Baza ta będzie się zawierać tylko w jednej tabeli. Wszelkie zmiany użytkownik będzie bezpośrednio nanosił na tą tabelę np. dodawanie, usuwanie lub aktualizację poszczególnych rekordów. Przykładem lokalnych baz danych mogą tu być: Access, Paradox, dBase.

2.. **Bazy typu klient-server** - główna baza tego typu jest przechowywana w zasobach serwera, który to na ogół jest wydzielony jako osobny komputer. Dostęp do niego jest realizowany za pośrednictwem innych komputerów – oczywiście przez sieć, zatem nie muszą się znajdować blisko siebie by korzystać z takiej bazy. Użytkownicy korzystając z takiej bazy nie korzystają bezpośrednio z jej zasobów, ponieważ odbywa się to za pośrednictwem programów zwanych klientami. Jeśli chodzi o serwery to najbardziej popularne na rynku obecnie są produkty firm: InterBase, Oracle, Sybase, Informix oraz Microsoft. Bardzo ważną cechą serwera jest możliwość korzystania wielu użytkowników, a wszystko to związane jest z licencją sprzedawaną przez producentów tego oprogramowania.

Ze względu na architekturę baz danych można wyróżnić:

1. **Bazy jednowarstwowe**, które wykonują natychmiast wszelkiego rodzaju zmiany, zaś program, który udostępnia użytkownikowi zawartość bazy ma z nim bezpośredni kontakt.

2. **Bazy dwuwarstwowe**, w których klient porozumiewa się z serwerem za pomocą specjalnych sterowników. Jeśli chodzi o samo połączenie to jest ono zależne od samego

serwera, natomiast kontrolowanie poprawności danych zależy od klienta. Rozwiązanie takie wiąże się ze sporym obciążeniem programu klienckiego.

Większość lokalnych baz danych opiera się na jednowarstwowym modelu, natomiast bazy typu klient-serwer na wielowarstwowym modelu architektury baz danych.

Obecnie ludzie, którzy zajmują się na co dzień bazami danych mogą łatwo stwierdzić, że różnią się one pomiędzy sobą w wielu aspektach. Praktycznie każda baza tworzy swoją własną kategorię i trudno je zasznuflakować. Jako cechy charakterystyczne dla danej grupy baz wyodrębnić można:

- a. model danych (*data model*)
- b. język zapytań (*query language*),
- c. model obliczeniowy (*computational model*).

Bazy danych można podzielić według struktur danych, których używają:

1. **Bazy proste** (kartotekowe) - każda tablica danych jest samodzielnym dokumentem i nie może współpracować z innymi tablicami. Do baz tego typu należą liczne programy typu - książka telefoniczna, książka kucharska, spis książek, kaset lub płyt. Wspólną cechą tych baz jest ich zastosowanie w jednym wybranym celu.

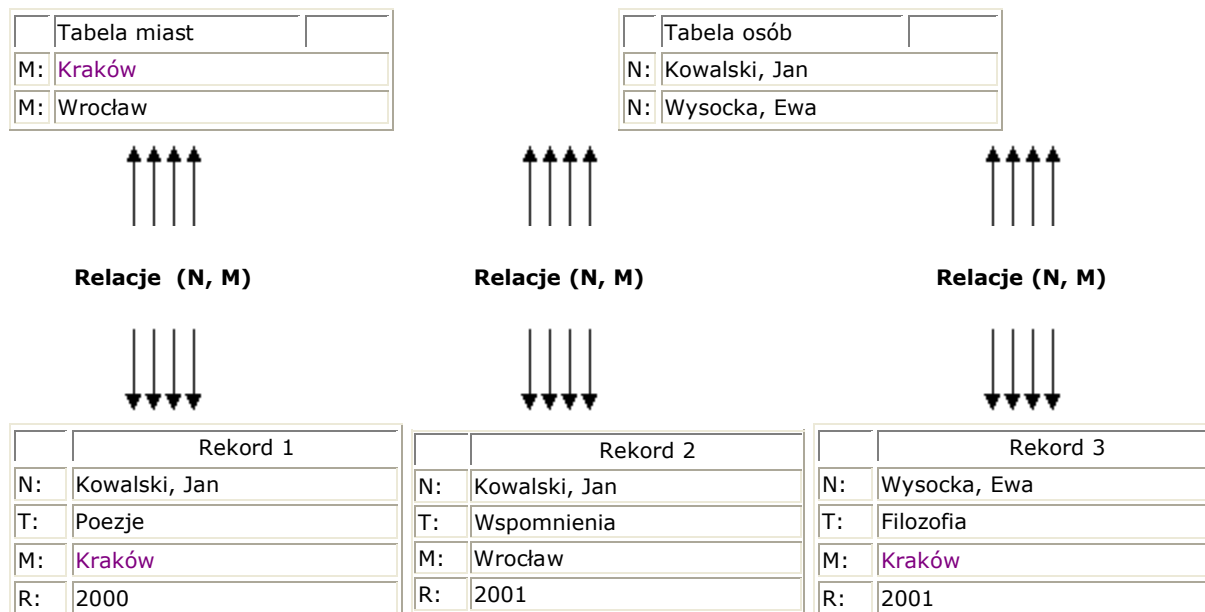
Tabela 1: Przykład bazy kartotekowej

Rekord 1	Rekord 2	Rekord 3
N: Kowalski, Jan T: Poezje M: Kraków R: 2000	N: Kowalski, Jan T: Wspomnienia M: Wrocław R: 2001	N: Wysocka, Ewa T: Filozofia M: Kraków R: 2001

2. **Relacyjne bazy danych** (RDB) - wiele tablic danych może współpracować ze sobą. Bazy relacyjne posiadają wewnętrzne języki programowania (SQL), za pomocą których możemy tworzyć własne menu oraz zaawansowane funkcje obsługi danych. Relacyjne bazy danych (jak również przeznaczony dla nich standard (SQL) oparte są na kilku prostych zasadach:

- a. Wszystkie wartości danych oparte są na prostych typach danych.
- b. Wszystkie dane w bazie relacyjnej przedstawiane są w formie dwuwymiarowych tabel (w matematycznym żargonie noszących nazwę „relacji”). Każda tabela zawiera zero lub więcej wierszy (w tymże żargonie - „krotki”) i jedną lub więcej kolumn („atrybuty”). Na każdy wiersz składają się jednakowo ułożone kolumny wypełnione wartościami, które z kolei w każdym wierszu mogą być inne.
- c. Po wprowadzeniu danych do bazy możliwe jest porównywanie wartości z różnych kolumn, zazwyczaj również z różnych tabel, i scalanie wierszy, gdy pochodzące z nich wartości są zgodne. Umożliwia to wiązanie danych i wykonywanie stosunkowo złożonych operacji w granicach całej bazy danych.
- d. Wszystkie operacje wykonywane są w oparciu o logikę, bez względu na położenie wiersza tabeli. Wiersze w relacyjnej bazie danych przechowywane są w porządku zupełnie dowolnym - nie musi on odzwierciedlać ani kolejności ich wprowadzania, ani kolejności ich przechowywania.
- e. Z braku możliwości identyfikacji wiersza przez jego pozycję pojawia się potrzeba obecności jednej lub więcej kolumn niepowtarzalnych w granicach całej tabeli, pozwalających odnaleźć konkretny wiersz. Kolumny te określa się jak „klucz podstawowy” (*primary key*) tabeli.

Tabela 2: Przykład bazy relacyjnej



3. **Obiektowe bazy danych (ODB)** - nie są one zdefiniowane żadnym oficjalnym standardem. Obowiązujący obecnie standard opracowany przez ODMG został opublikowany w 1993 roku. Jednym z podstawowych celów modelu obiektowego jest bezpośrednie odwzorowanie obiektów i powiązań między nimi wchodzących w skład aplikacji na zbiór obiektów i powiązań w bazie danych. Dzięki mechanizmom obiektowym można też zwiększyć niezależność danych od aplikacji poprzez przeniesienie procedur obsługi danych (w postaci metod) do systemu zarządzania bazą. Model danych w obiektowych bazach danych posługuje się pojęciami takimi jak klasy, atrybuty, metody, udostępnia identyfikatory obiektów (OID), hermetyzacje danych oraz metod i wielokrotnego dziedziczenia. Obiektowe bazy danych łączą własności obiektowości i obiektowych języków programowania z możliwościami systemów bazodanowych. Rozszerzają możliwości obiektowych języków programowania (takich jak C++, Java czy Smalltalk) czyniąc z nich narzędzia do łatwego i efektywnego tworzenia systemów baz danych zmniejszając stopień złożoności i ilość kodu programów.

4. **Strumieniowe bazy danych** to bazy danych, w których dane są przedstawione w postaci zbioru strumieni danych. System zarządzania taką bazą nazywany jest strumieniowym systemem zarządzania danymi (*Data Stream Management System*). Większość strumieniowych baz danych w chwili obecnej znajduje się w fazach prototypowych i nie powstały dotychczas komercyjne rozwiązania.

Różnice pomiędzy poszczególnymi rodzajami baz danych są bardzo widoczne szczególnie w przypadku baz relacyjnych i baz obiektowych.

Tabela 3: Porównanie relacyjnej i obiektowej baz danych

	RELACYJNE	OBIEKTOWE
Cechy podstawowe	<ul style="list-style-type: none"> • Dane zawarte w tabelach • Tabele składają się z kolumn • Typy - predefiniowalne 	<ul style="list-style-type: none"> • Obiekt w bazie reprezentuje obiekt w świecie rzeczywistym • Typ obiektowy (klasa): <ul style="list-style-type: none"> ○ definicja złożonego typu

	<ul style="list-style-type: none"> • Liczba wierszy zmienna • Value-based • Nie ma wskaźników lecz klucze zewnętrzne 	<p>danych (może zawierać inne typy obiektowe lub ich kolekcje)</p> <ul style="list-style-type: none"> ○ procedury (metody) i operatory do manipulowania tymi danymi <ul style="list-style-type: none"> • Identity-based • Enkapsulacja • Dziedziczenie: <ul style="list-style-type: none"> ○ strukturalne: potomek dziedziczy strukturę danych ○ behawioralne: potomek dziedziczy metody i operatory
Przykłady systemów	<u>Oracle, Informix, Sybase, Ingres, DB2, Progress, Gupta, Access</u>	<u>GemStone, O2, Persistence, Versant, POET, Objectivity, ODI</u>
Stan na dzisiaj	Dominuje w zastosowaniach komercyjnych (ok. 95% rynku baz danych)	Mniej popularne, jednak dobrze rokują na przyszłość
Zalety	<ul style="list-style-type: none"> • niezależność od języka programowania • sprawdzone, dobrze zdefiniowana teoria • możliwość zarządzania wielką ilością danych • możliwość złożonych kryteriów wyszukiwawczych • możliwość dostępu do danych fizycznych • dobre mechanizmy kontroli dostępu do danych • mechanizmy perspektyw 	<ul style="list-style-type: none"> • dość łatwa reprezentacja świata • dokładnie reprezentuje złożone zależności między obiektami • łatwość działania na złożonych obiektach • duża podatność na zmiany • możliwość definiowania własnych typów, metod • dobra integracja z językami programowania ogólnego przeznaczenia (np. C++, Smalltalk) • ujednolicony model pojęciowo - obiektowe podejście do analizy, projektowania i implementacji
Wady	<ul style="list-style-type: none"> • brak bezpośredniej reprezentacji n-m • dla trudniejszych problemów bardzo dużo tabel • mało naturalna reprezentacja danych • ograniczona podatność na zmiany • brak złożonych typów danych • trudne operowanie na danych złożonych 	<ul style="list-style-type: none"> • powiązanie z jednym językiem programowania • słaba obsługa przeszukiwania danych • brak powszechnie zaakceptowanego języka zapytań • brak możliwości optymalizacji zapytań • trudny lub nawet niemożliwy dostęp do fizycznych danych • słaba kontrola dostępu • małe możliwości optymalizacji

	<ul style="list-style-type: none"> • trudne operowanie na danych rozproszonych w sieci heterogenicznej • niezgodność z modelem używanym przez języki ogólnego przeznaczenia (<i>impedance mismatch</i>) 	pracy serwera
Lepsze gdy...	<ul style="list-style-type: none"> • dane są proste, niezagnieżdżone, łatwe do umieszczenia w tablicy • dane mają postać bierna, a procesy korzystające z danych stale się zmieniają • często potrzeba wyszukiwać dane spełniające różnorodne warunki 	<ul style="list-style-type: none"> • dane mają złożoną lub zagnieżdżoną strukturę • zdefiniowana przez użytkownika • dane tworzą hierarchie • dane są rozproszone w sieci heterogenicznej • dane dynamicznie zmieniają rozmiar

2.3. Zapis informacji w bazie danych

Encja (*ang. entity*), jest to termin ukuty w środowiskach i na potrzeby specjalistów z dziedziny baz danych. Oznacza przedmiot fizyczny (lub abstrakcja) rozpoznawalny jako samodzielny i niezależny w organizacji rzeczywistości byt; rzecz znacząca. Encja jest to jednoznacznie identyfikowany składnik badanej rzeczywistości, o którym informacja jest lub może być zbierana i przechowywana. Przykładami encji są: PRACOWNIK, KLIENT, DOSTAWCA, ZAMÓWIENIE, MAGAZYN, FAKTURA, POZYCJA, PRZECENA, KONTO. Encja jest urzeczywistniona poprzez wystąpienie. Przykładowo, wystąpieniami encji KLIENT są: *Nowak, Dobrowolski, Kwiatkowski* itd. Jak widać z powyższych przykładów encje opisuje się za pomocą rzeczowników lub wyrażeń rzeczownikowych w liczbie pojedynczej.

Sygnatura	Tytuł	Wydaw	Rok wyd.
100	TCR/IP	ReadMe	1996
200	Ogólne struktury danych	NAKOM	1995
300	Turbo Pascal	NAKOM	1994
400	Turbo Vision	NAKOM	1994
500	Programowanie obiektowe w c++	LUPUS	1994

Diagram labels and their corresponding table components:

- Klucz**: Points to the 'Sygnatura' column.
- Rekord**: Points to the entire row for 'Ogólne struktury danych'.
- Pole**: Points to the 'Tytuł' column.
- Wystąpienie encji**: Points to the 'Tytuł' cell 'Turbo Vision'.
- Encja**: Points to the 'Wydaw' column.
- Atrybut**: Points to the 'Rok wyd.' column.

Rysunek 1: Tabela bazy danych

Atrybut – jest cechą, elementem charakteryzującym encje i związki w badanej dziedzinie przedmiotowej. Zestaw atrybutów, który jednoznacznie opisuje encję, nazywa się **wiązką atrybutów**. Wiązka powinna składać się, z co najmniej dwóch atrybutów opisujących daną encję.

Klucz - zwany identyfikatorem pełni szczególną rolę z zakresie atrybutów encji. Pozwala on na jednoznaczne określenie wystąpienia encji. Jeśli używa się jednego atrybutu dla określenia encji, to mamy do czynienia z kluczem pojedynczym, a jeśli w tym celu używa się więcej niż jednego atrybutu, to z kluczem złożonym.

Rekord - jest jednostką zapisu danych w tabeli. Jeden zapis odnosi się do pojedynczego *wystąpienia encji* i składa się z ciągu wartości *pól*, odpowiadających standaryzowanym *cechom* encji. Podczas prezentacji tabeli rekordom zazwyczaj nadaje się postać wierszy, może też być podawany w formie formularza. W zależności od struktury rekordu można wyróżnić:

► rekordy jednostopniowe

Tabela 4: Przykładowy rekord jednostopniowy

	<i>Autor</i>	<i>Tytuł</i>	<i>Podtytuł</i>	<i>Miasto</i>	<i>Rok</i>	<i>.....</i>
Rekordy	Kowalski, Jan	Poezje		Warszawa	2001	
	Kowalski, Jan	Wspomnienia		Wrocław	2001	
	Wysocka, Ewa	Wspomnienia	Młodość	Wrocław	1999	

► rekordy wielostopniowe

Tabela 5: Przykładowy rekord wielostopniowy

<i>Hasło</i>	<i>Strefa 1</i>	<i>Strefa 2</i>	<i>Strefa 4</i>
Kowalski	Poezje	Wyd. 3 zm.	Warszawa
Jan			PIW
(1945-1998)			2001

Lub

Hasło	<i>Nazwisko</i>	Kowalski	<i>Imię</i>	Jan	<i>Data</i>	(1945-1998)
Strefa 1	<i>Tytuł</i>	Poezje	<i>Podtyt.</i>			
Strefa 2	<i>Wyd.</i>	Wyd. 3	<i>Równol.</i>			
Strefa 4	<i>Miejsce</i>	Warszawa	<i>Wydawca</i>	PIW	<i>Rok</i>	2001

Pole - jest podstawowym składnikiem budowy rekordu. Służy do przechowywania wartości pojedynczych atrybutów. Podczas prezentacji tabeli polem poszczególnych

rekordów zazwyczaj nadaje się postać rubryk (komórek). Pola dotyczące tego samego atrybutu są zazwyczaj zgrupowane w jednej kolumnie tabeli.

Tabela - stanowi zasadniczy element budowy bazy danych. Pojedyncza tabela składa się z rekordów. Wszystkie rekordy tabeli mają identyczną budowę, pozwalającą na przechowywanie informacji o wybranych cechach. Zatem jeżeli weźmiemy pojedynczą tabelę, to każde ujęte w niej wystąpienie encji będzie opisane za pomocą właściwych dla niego wartości z góry ustalonej wiązki atrybutów. Zespół cech pełni rolę nagłówka tabeli. Jego postać jest uwarunkowana projektem tabeli i całej bazy danych.

Dane pierwotne - nazwa ta oznacza zespół wszystkich cech, których wartości nie da się wywieść z innych cech opisanych w tabeli. Zestaw danych pierwotnych zależy od projektu tabeli. Dane pierwotne muszą być wprowadzone do bazy danych z zewnątrz i muszą być w niej przechowywane.

Dane pochodne - nazwa ta oznacza zespół tych cech, których wartości da się wywnioskować (np. wyliczyć) z innych cech, wychodząc od cech pierwotnych. Dane pochodne powinny być wyznaczane, a nie wprowadzane do bazy danych z zewnątrz. Przechowywanie danych pochodnych zamiast wyliczania ich za każdym razem, kiedy są potrzebne, może usprawnić użytkowanie bazy danych. Jednak każda modyfikacja danych pierwotnych wymaga aktualizacji zależnych od nich danych pochodnych. Aktualizację taką przeprowadza się często za pomocą niewielkich programów uruchamianych automatycznie w ściśle określonych sytuacjach w sposób niedostrzegalny dla „zwykłego użytkownika” — noszą one nazwę *wyzwalaczy*.

Bieżący stan bazy danych - zespół informacji o wszystkich wystąpieniach encji ujętych w danej chwili w tabelach bazy danych. Stan bazy danych na ogół ulega zmianie podczas jej użytkowania (aktualizacja).

Zapytanie - sformalizowana postać pytania, na które da się udzielić odpowiedzi w oparciu o bieżący stan bazy danych. Pytania formułuje się w jednym z języków obsługi baz danych. Obecnie najszerzej stosowanym językiem tego typu jest SQL (ang. Structured Query Language). Pytania można zadawać bezpośrednio (formułując samodzielnie zapytanie w odpowiednim języku i przekazując je procesorowi zapytań) lub pośrednio (posługując się interfejsem użytkownika związanym np. z obsługą programu użytkowego).

2.4. Języki zarządzające bazami danych

Do zarządzania bazami danych stosuje się języki, które można podzielić na cztery zasadnicze grupy:

1. Język definiowania danych (Data Definition Language - DDL), który umożliwia definiowanie struktury danych zawartych w bazie.
2. Język manipulowania danymi (Data Manipulation Language - DML), który umożliwia wypełnienie, modyfikowanie i usuwanie danych z bazy.
3. Język sterowania danymi (Data Control Language), który umożliwia sterowanie transakcjami tj. akceptacja lub wycofanie.
4. Język zapytań (Query Language), który umożliwia pobieranie informacji z bazy za pośrednictwem określonych zapytań, warunków.

W praktyce te cztery języki są ze sobą zintegrowane. Takim zintegrowanym językiem jest m.in. SQL (Structured Query Language). Do listy języków można tu jeszcze dodać rozszerzenia proceduralne stosowane przez różne firmy produkujące SZBD: pl/pgsql w PostgreSQL, PL/SQL w Oracle i inne.

2.5. Elementy budowy relacyjnej bazy danych

Analizując strukturę budowy bazy danych można wyróżnić w niej szereg obiektów:

1. **Obiekt** - każdy element bazy danych, który ma swoje właściwości, przechowuje lub przetwarza informacje. Obiektem będzie tabela danych, kwerenda, a także kreator ułatwiający tworzenie tabel, kwerend i raportów.
2. **Tabele** – są najprostszą formą przedstawiania danych zawartych w bazach. Poszczególne kolumny reprezentują pola, a wiersze reprezentują rekordy bazy danych, co pozwala na pokazanie jednocześnie wielu rekordów i pól. Baza danych może zawierać wiele tabel.
3. **Kwerenda** – to, wg Słownika Wyrazów Obcych, poszukiwanie informacji (w aktach, archiwach, bibliotekach) potrzebnych do wyjaśnienia jakiejś kwestii. Słowo to pochodzi od łacińskiego *querenda*, które z kolei oznacza rzeczy poszukiwane, to czego należy poszukiwać. To samo pochodzenie ma także angielskie słowo *query*, tłumaczone jednak najczęściej jako zapytanie. Dzieje się tak dlatego, że oprócz operacji wyszukiwania informacji, a więc takich, które nie zmieniają stanu bazy danych, język QBE dysponuje także operacjami wprowadzania, usuwania i aktualizacji danych.

Rodzaje kwerend:

- × **Kwerendy wybierające** - najczęściej używane, umożliwiają wybieranie rekordów, tworzenie nowych pól obliczeniowych i podsumowywanie danych. Kwerendy tego typu są podobne do filtrów, jednak umożliwiają również:
 - ▶ Przeprowadzanie zapytań w więcej niż jednej tabeli,
 - ▶ Tworzenie nowych pól obliczeniowych,
 - ▶ Podsumowywanie i grupowanie danych,
 - ▶ Wyznaczanie pól do ukrycia lub pokazania.
 - ▶ Kwerendy modyfikujące
- × **Kwerendy modyfikujące** - pozwalają na wprowadzanie zmian w wielu rekordach. Istnieją cztery rodzaje kwerend modyfikujących:
 - ▶ Kwerenda usuwająca - usuwa grupę rekordów z jednej lub kilku tabel. Kwerenda usuwająca może być użyta na przykład do usunięcia produktów, które zostały wycofane z oferty i na które nie ma zamówień. Użycie kwerendy usuwającej powoduje usunięcie całych rekordów, nie zaś wybranych pól w rekordach.
 - ▶ Kwerenda aktualizująca - dokonuje globalnych zmian w grupie rekordów w tabeli lub kilku tabelach. Na przykład można

podwyższyć o 10% ceny wszystkich produktów mlecznych lub dać pięcioprocentową podwyżkę wszystkim pracownikom należącym do określonej kategorii zawodowej. Za pomocą kwerend aktualizujących można zmieniać dane w istniejących tabelach.

- ▶ Kwerenda dołączająca - dodaje grupę rekordów z tabeli lub tabel na końcu innej tabeli lub tabel. Na przykład jeśli firmie przybywają nowi klienci, których dane znajdują się w osobnej bazie danych, aby uniknąć wpisywania wszystkich tych informacji, można dołączyć je do tabeli "Klienci".
 - ▶ Kwerenda tworząca tabelę - tworzy nową tabelę z wszystkich lub części danych znajdujących się w jednej lub kilku tabelach. Kwerendy tworzące tabele są przydatne w następujących sytuacjach:
 - Tworzenie tabel, które mają być eksportowane do innych baz danych programu Microsoft Access.
 - Tworzenie raportów zawierających dane od określonego momentu.
 - Tworzenie kopii zapasowej tabeli.
 - Tworzenie tabeli archiwalnej, zawierającej nieaktualne rekordy.
- × **Kwerendy krzyżowe** - dają możliwość gromadzenia danych lub wykonywania akcji na danych z kilku tabel lub kwerend. Na przykład można obejrzeć informacje o klientach wraz z zamówieniami, które zostały przez nich złożone.⁴

4. **Raporty** – wykorzystuje się do wydruku informacji z bazy. Są to zbiorcze zestawienia danych, które można uzupełnić opisami, rysunkami i wykresami.

5. **Formularze** – służą przedstawianiu w czytelny sposób zawartości pojedynczych rekordów. Umożliwiają i ułatwiają wprowadzanie, zmianę i przeglądanie danych w bazie. Przypominają neoklasyczne formularze z rubrykami i objaśnieniami.

6. **Filtry** – służą wyselekcjonowaniu z bazy danych tylko tych informacji, które spełniają określone warunki. Po włączeniu filtra, aż do czasu jego wyłączenia baza jest widziana tak, jakby posiadała tylko rekordy spełniające warunki podane przy budowie filtru.

7. **Sortowanie** – służy przedstawianiu rekordów w innej kolejności niż były one wprowadzone do bazy (np. alfabetycznie względem wybranego pola). Sortowanie nie zmienia fizycznej kolejności rekordów, a tylko sposób ich prezentacji użytkownikowi.

8. **Relacja** jest to sposób powiązania tabel w bazie danych. Pozwala korzystać z danych zawartych w wielu tabelach tak, jakby to była jedna tabela. Powiązania mogą być typu **jeden-do-jednego**, gdy każdemu wierszowi z tabeli głównej może odpowiadać dokładnie jeden wiersz tabeli związanej. W praktyce częstsze są powiązania typu **jeden-do-wielu**, gdy każdemu rekordowi tabeli głównej może odpowiadać wiele rekordów tabeli związanej. Powiązania **wiele-do-wielu** nie są realizowane bezpośrednio; są natomiast dekomponowane na powiązania jeden-do-wielu z użyciem tabeli pośredniej.

▪ Relacja jeden-do-jednego

Charakteryzuje się tym, że każdemu wystąpieniu jednej z dwóch encji istnieje dokładnie jedno wystąpienie drugiej encji pozostające z nim w równoważnym związku np. czek i opłata (opłata jest realizowana za pomocą jednego czeku i za pomocą jednego czeku można zrealizować tylko jedną opłatę). Ten typ relacji spotyka się rzadko, ponieważ większość informacji powiązanych w ten sposób byłoby zawartych w jednej tabeli. Relacji jeden-do-jednego można używać do podziału tabeli z wieloma polami, do odizolowania

⁴ Sikora M.; New Accessmaster; <http://wr.atr.bydgoszcz.pl/kiz/access/>

części tabeli ze względów bezpieczeństwa, albo do przechowania informacji odnoszącej się tylko do podzbioru tabeli głównej.

Rysunek 2: Schemat relacji jeden-do-jednego



- **Relacja jeden-do-wielu**

Charakteryzuje się tym, że dla każdej instancji jednej encji istnieje wiele instancji drugiej encji pozostającej z nią w rozważanym związku. Relacja jeden-do-wielu jest realizowana poprzez utworzenie atrybutu w encji po stronie wiele aby umieścić w nim klucz encji znajdującej się po stronie jeden. Tak utworzony atrybut encji po stronie wiele nosi nazwę klucza obcego ponieważ jest on głównym kluczem w innej tabeli. Relacja jeden-do-wielu jest najbardziej powszechnym typem relacji.

Rysunek 3: Schemat relacji jeden-do-wielu



- **Relacja wiele-do-wielu**

W relacji wiele-do-wielu, rekord w tabeli A może mieć wiele dopasowanych do niego rekordów z tabeli B i tak samo rekord w tabeli B może mieć wiele dopasowanych do niego rekordów z tabeli A. Jest to możliwe tylko przez zdefiniowanie trzeciej tabeli (nazywanej tabelą łącząca), której klucz podstawowy składa się z dwóch pól - kluczy obcych z tabel A i B. Relacja wiele-do-wielu jest definiowana jako dwie relacje jeden-do-wielu z trzecią tabelą. Na przykład, tabele „Zamówienia” i „Produkty” są powiązane relacją wiele-do-wielu zdefiniowaną przez utworzenie dwóch relacji jeden-do-wielu z tabelą „Opisy zamówień”.

Rysunek 4: Schemat relacji wiele-do-wielu

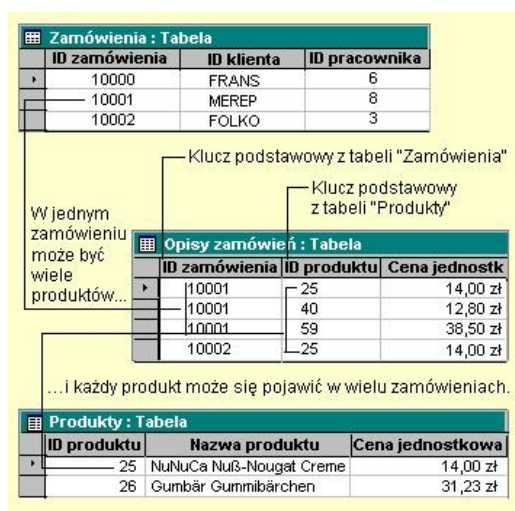


Tabela 6: Przykłady typów relacji

Relacja	Przykład	Znaczenie
Jeden-do-jednego	Dziekan-Wydział	Każde wystąpienie encji Dziekan jest powiązane tylko z jednym wystąpieniem encji Wydział . Zatem jeden Dziekan kieruje jednym Wydziałem
Jeden-do-wielu	Wydział-Student	Każde wystąpienie encji Wydział powiązane jest jednym lub wieloma wystąpieniami encji Student , przy czym każde wystąpienie encji Student powiązane jest tylko jednym wystąpieniem encji Wydział . Zatem Wydział posiada wielu Studentów , natomiast Student studiuje wyłącznie na jednym Wydziale
Wiele-do-wielu	Książka - Autor	Każde wystąpienie encji Książka powiązane jest z wieloma wystąpieniami encji Autor i odwrotnie każde wystąpienie encji Autor powiązane jest z wieloma wystąpieniami encji Książka . Jest to sytuacja, gdzie Książka może być napisana przez jednego lub wielu autorów i jeden Autor jest podpisany pod jednym lub

2.6. Integralność danych

Relacyjny model bazy danych dostarcza dodatkowych, specyficznych dla siebie postaci reguł integralności:

Integralność encji: każda tabela musi posiadać klucz główny, a wartości klucza głównego muszą być w ramach tabeli unikalne i nie równe *NULL*. W szczególności, zapobiega to wystąpieniu w tabeli powtórzeń wierszy.

Integralność referencyjna: każda wartość klucza obcego może być albo równa jakiejś wartości klucza głównego występującej w tabeli powiązanej, lub (ewentualnie) *NULL*. Pociąga to za sobą konieczność określenia reguły postępowania w wypadku usuwania wiersza z tabeli powiązanej, co mogłoby unieważnić niektóre wartości kluczy obcych w tabelach do niej się odnoszących. W grę wchodzi trzy postacie takiej reguły:

1. *Restricted:* usunięcie wiersza jest zabronione, dopóki nie zostaną usunięte lub odpowiednio zmodyfikowane wiersze z innych tabel, których wartości kluczy obcych stałyby się wskutek tej operacji nieważne;
2. *Cascades:* usunięcie wiersza powoduje automatyczne usunięcie z innych tabel wszystkich wierszy, dla których wartości kluczy obcych stały się nieważne;
3. *Nullifies:* nieważne wartości kluczy obcych ulegają zastąpieniu przez *NULL*.

2. SYSTEMY ZARZĄDZANIA BAZAMI DANYCH

3.1. Rodzaje systemów zarządzania bazami danych

Baza danych to nie tylko dane w niej zawarte, jest to również program komputerowy wyspecjalizowany do gromadzenia i przetwarzania tych danych. Program taki (lub zestaw programów) nazywany jest „Systemem zarządzania bazą danych” - z angielskiego *Database Management System* (DBMS).

Systemy DBMS profesjonalne:

1. **Oracle Corporation** - Oracle to druga co do wielkości po Microsoftzie firma zajmująca się oprogramowaniem. Oracle DBMS - na około 90 platformach sprzętowych, Personal Oracle na PC, Oracle Media Server, Oracle Video Server, obsługa hurtowni danych.
2. **PROGRESS** Application Development Environment - jeden z najbardziej popularnych w Polsce.
3. **Ingres** (Computer Associates, CA),
4. **Sybase PL**
5. **Adabas C**, Siemens Nixdorf Polska
6. **Gupta SQLBase** (Centura Corporation)
7. **Microsoft SQL Server** for Windows NT/2000/XP

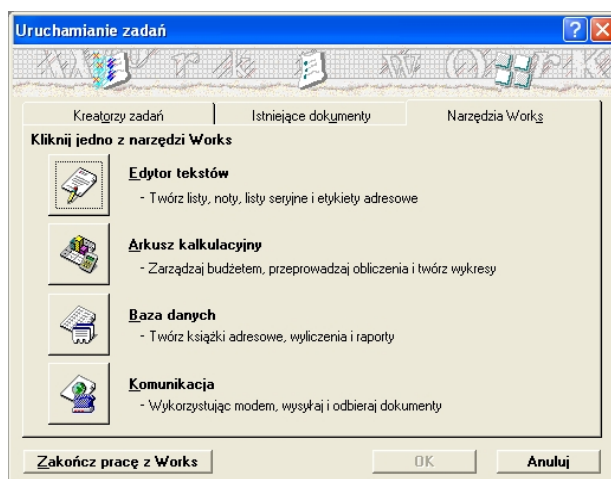
Systemy i programy na komputery osobiste i stacje robocze DBMS dla mniejszych firm:

1. **Microsoft Access** (Windows) - łatwy, SQL, język Access Basic
2. **FoxPro** (Microsoft) pod DOS, Windows, Mac, Unix, wersja polska.
3. **Paradox** (Borland) - Query by Example, pytania przez analogie.
4. **dBase** (Borland), od 1981 roku (dBase II pod CP/M), wersja polska ISIS wzorowany na dBase.
5. **HyperCard** (MacIntosh) - prosta baza hipertekstowa
6. **Microsoft Works** – pakiet zintegrowany

3.2. Zintegrowany pakiet Microsoft Works

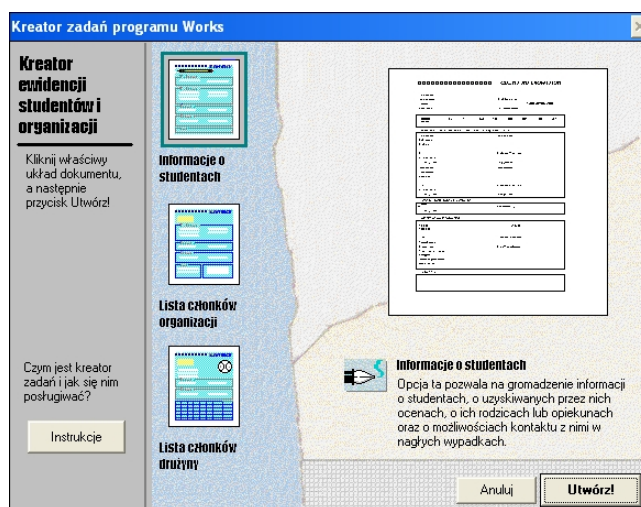
Microsoft Works 4.0 jest pakietem typowo sekretarskim, do którego wprowadzono niektóre ułatwiające pracę funkcje obecne w Office. Pakiet zawiera uproszczony, w porównaniu z Word, ale całkiem sprawny edytor tekstów, prosty arkusz kalkulacyjny, kartotekową bazę danych i narzędzia do obsługi modemu. Pakiet Works służy do szybkiego tworzenia najczęściej używanych dokumentów, takich jak listy, faktury, zestawienia wydatków, spisy telefonów i adresów itp. Pomaga w tym kilkadziesiąt kreatorów podzielonych na kategorie, aby łatwiej odnajdywać właściwy typ dokumentu. Okno uruchamiania zadań, które pokazuje się przy starcie programu, pozwala szybko wybrać właściwy kreator, a następnie zmodyfikować wygląd dokumentu. Po prawej stronie okna dokumentu użytkownik ma do dyspozycji zestaw tematów pomocy adekwatnych do wykonywanej pracy.

Rysunek 5: Okno dialogowe „Uruchamianie zadań” MS Works



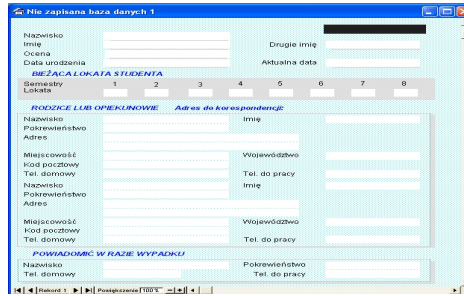
Baza danych pakietu MS Works jest w stanie przechowywać informacje tylko w jednej tabeli, często określanej jako plik płaski. Użytkownik ma tu możliwość tworzenia prostych, kartotekowych baz danych samodzielnie albo z pomocą programu, po wybraniu opcji zwanej kreatorem. Pomoc programu polega na zaoferowaniu dość ogólnej struktury bazy danych przeznaczonej do typowych zastosowań, której nadaje konkretną postać.

Rysunek 6: Okno dialogowe „Kreator zadań programu Works”



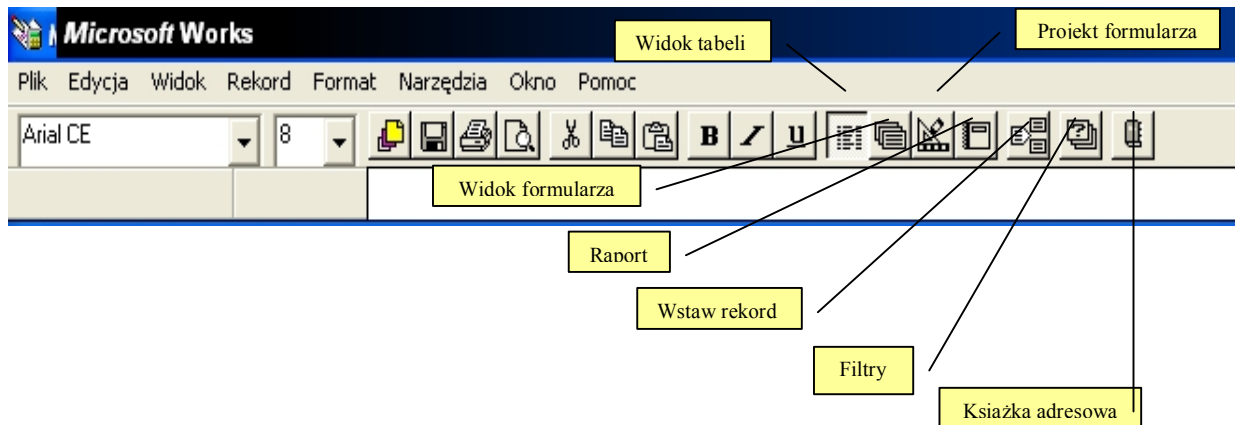
Po wybraniu układu jednego z proponowanych dokumentów kreator otwiera bazę danych w postaci formularza, do którego można wpisywać dane do poszczególnych rekordów tabeli.

Rysunek 7: Formularz wpisywania danych do poszczególnych rekordów tabeli



Korzystając z ikony umieszczonej na pasku narzędzi utworzoną bazę danych można pokazać w widoku tabeli.

Rysunek 8: Pasek menu i pasek narzędzi MS Works



Rysunek 9: Widok tabeli bazy danych MS Works

Nie zapisana baza danych 1					
<input checked="" type="checkbox"/>	Nagłówek	Nazwisko	Imię	2. imię	<input type="checkbox"/>
<input type="checkbox"/>	1				
<input type="checkbox"/>	2				
<input type="checkbox"/>	3				
<input type="checkbox"/>	4				
<input type="checkbox"/>	5				
<input type="checkbox"/>	6				
<input type="checkbox"/>	7				
<input type="checkbox"/>	8				
<input type="checkbox"/>	9				
<input type="checkbox"/>	10				
<input type="checkbox"/>	11				
<input type="checkbox"/>	12				
<input type="checkbox"/>	13				
<input type="checkbox"/>	14				
<input type="checkbox"/>	15				
<input type="checkbox"/>	16				
<input type="checkbox"/>	17				
<input type="checkbox"/>	18				

Bazę danych można utworzyć też korzystając z okna dialogowego: „Utwórz bazę danych”. W oknie tym istnieje możliwość określenia formatu poszczególnych pól tworzonych w bazie danych

Rysunek 10: Okna dialogowe do tworzenia bazy danych MS Works

Pole w formacie ogólnym

Utwórz bazę danych

Dodaj pola, aby utworzyć bazę danych. Aby dodać pole, napisz nazwę i zaznacz format dla danego pola.

Nazwa pola: Pole 1

Format

- Ogólny
- Liczba
- Data
- Godzina
- Tekst
- Ułamek
- Serwisy

Wygląd

Dla tego formatu nie istnieją dodatkowe opcje. Format ogólny wyświetla liczby tak dokładnie, jak jest to możliwe. Tekst jest wyrównywany do lewej, a liczby do prawej.

Automatycznie wprowadź wartość domyślną

Dodaj Anuluj

Pole w formacie data

Utwórz bazę danych

Dodaj pola, aby utworzyć bazę danych. Aby dodać pole, napisz nazwę i zaznacz format dla danego pola.

Nazwa pola: Data urodzenia

Format

- Ogólny
- Liczba
- Data
- Godzina
- Tekst
- Ułamek
- Serwisy

Wygląd

- 2004-01-29
- 01-29
- 2004-01
- 29 stycznia 2004
- styczeń 2004

Automatycznie wprowadź wartość domyślną

Dodaj Wykonane

Pole w formacie liczba

Utwórz bazę danych

Dodaj pola, aby utworzyć bazę danych. Aby dodać pole, napisz nazwę i zaznacz format dla danego pola.

Nazwa pola: Wynagrodzenie za godzinę

Format

- Ogólny
- Liczba
- Data
- Godzina
- Tekst
- Ułamek
- Serwisy

Wygląd

- 1234,56
- 1 234,56
- 1 234,56 zł
- 123456,00%
- 1,23E+03
- 01235

Miejsca dziesiętne: 2

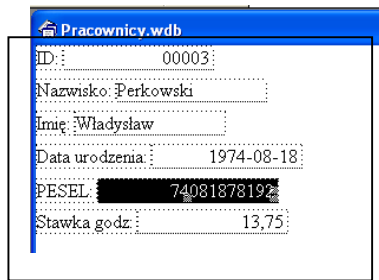
Liczby ujemne na czerwono

Automatycznie wprowadź wartość domyślną

Dodaj Wykonane

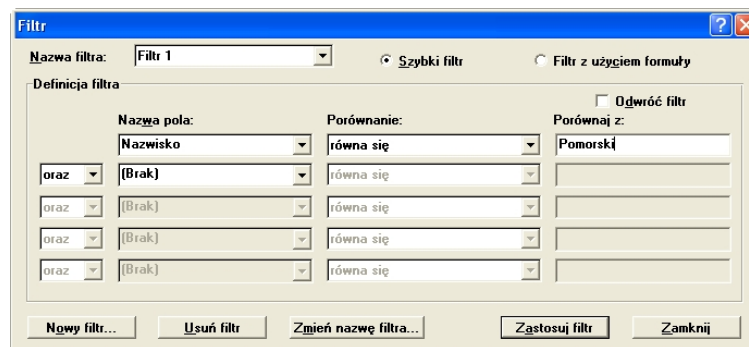
Określone w powyższy sposób pola i ich formaty stworzą bazę danych w postaci tabeli. Przechodząc do opcji projekt formularza można w dowolny sposób kształtować wygląd formularza. Works daje duże możliwości kształtowania wyglądu formularza, zaczynające się od typowych możliwości umieszczania przy poszczególnych polach napisów lepiej określających ich zawartość niż sama nazwa i określania formatu danych liczbowych, poprzez swobodę rozmieszczania napisów na ekranie, po dodawanie elementów graficznych, z obrazkami włącznie.

Rysunek 11: Formularz MS Works w widoku projektu



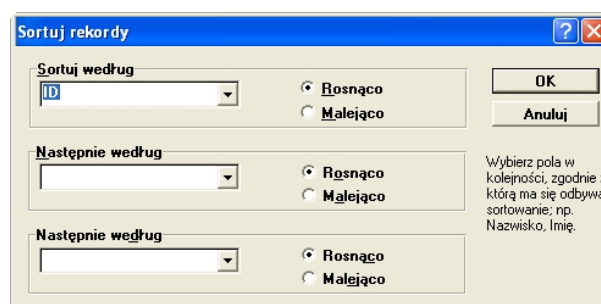
Program Works pozwala na dokonywanie filtrowania danych zawartych w tabeli wg określonego porównania. W oknie filtr, warunki są formułowane za pomocą słów, które wybiera się z pola okna (np. równa się, zawiera itp.). W oddzielnym polu wpisuje się wartość do porównań. Można sformułować kilka warunków i określić wymagane związki logiczne między nimi (np. oraz, lub). Po zastosowaniu filtra na ekranie w widoku tabeli pokazywane są tylko te rekordy, które spełniają ustalony warunek.

Rysunek 12: Okno dialogowe tworzenia filtra w bazie danych MS Works



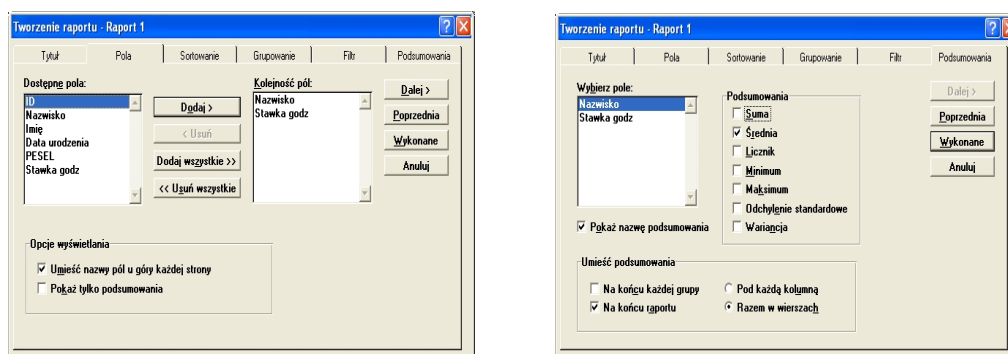
Program Works pozwala na sortowanie rekordów. W oknie dialogowym „Sortuj rekordy” należy podać nazwy pól, według których następuje sortowanie. Podczas sortowania mogą być uwzględniane trzy pola jednocześnie. Operacja sortowania jest przeprowadzana bez konieczności utworzenia nowego pliku bazy. Sortowanie można wykonywać zarówno w porządku rosnącym jak i malejącym

Rysunek 13: Okno dialogowe sortowania rekordów MS Works



Ważną zdolnością programu Works jest możliwość przygotowywania informacji o poszczególnych rekordach lub grupach rekordów w postaci raportów. Korzystając z okna dialogowego „Tworzenie raportu” dokonuje się wyboru pól, które mają być przedstawione w raporcie. Okno to pozwala również wprowadzić w raporcie podsumowania bądź dane statystyczne (np. średnia stawka godzinowa).

Rysunek 14: Okno dialogowe tworzenia raportów MS Works



Po pełnym sformatowaniu raportu można przejść do obejrzenia definicji raportu bądź obejrzeć raport w podglądzie wydruku.

Rysunek 15: Widok definicji raportu MS Works

Tytuł	A	B	C	D	E
Tytuł			Pracownicy.wdb - Stawki prac		
Nagłówki	Nazwisko	Imię	Stawka godz		
Nagłówki					
Rekord	=Nazwisko	=Imię	=Stawka godz		
Podsumowanie			WARTSR:		
Podsumowanie			=WARTSR(Star		

Pracownicy.wdb - Stawki prac		
Nazwisko	Imię	Stawka godz
Grodzki	Zygmunt	7,50
Szczesiński	Waldemar	10,25
Pomorski	Grzegorz	11,25
Perkowski	Władysław	13,75
		WARTSR:
		10,69

Rysunek 16: Podgląd wydruku raportu MS Works

Przedstawiony program MS Works jest prostym programem pozwalającym tworzyć kartotekowe bazy danych, dzięki czemu, zaspokaja on zapewne potrzeby pewnej grupy użytkowników, eliminuje konieczność stosowania wielu drogich i skomplikowanych programów specjalistycznych.

3.3. MS Access jako narzędzie konstrukcji baz danych

3.3.1. Charakterystyka programu MS Access

Microsoft Access 2002 jest prawdziwie zaawansowanym i profesjonalnym programem baz danych. Jest on dostępny również dla niedoświadczonych użytkowników. Microsoft Access nie jest, jak często się mylnie uważa, bazą danych, lecz systemem zarządzającym bazą danych (DBMS - Database Management System), za pomocą którego można tworzyć i administrować bazy danych. Access należy przy tym do systemów zarządzania relacyjnymi bazami danych. Oznacza to, że poszczególne tabele mogą być ze sobą powiązane. Za pomocą tego programu można zarządzać wszystkimi niezbędnymi informacjami używając jednego pliku bazy danych. W pliku takim można dzielić dane na odrębne części zwane tabelami. Za pomocą formularzy elektronicznych można oglądać, dodawać i aktualizować dane w tabelach; za pomocą kwerend można znajdować i pobierać tylko te dane, które są rzeczywiście aktualnie niezbędne; za pomocą raportów można analizować i drukować dane w określonym przez siebie układzie. Pomimo tego, że wszystkie obiekty bazy danych

znajdują się w jednym pliku istnieje możliwość importowania /eksportowania/ pojedynczych tabel. Program MS Access łączy w sobie funkcje:

- ▶ programu dla projektanta,
- ▶ interakcyjnego narzędzia do operowania danymi,
- ▶ interpretera wykonującego gotowe aplikacje.

Wielką zaletą programu Access 2002, składnika pakietu Microsoft Office XP, jest to, że jest on narzędziem do tworzenia zarówno prostych, jak i złożonych baz danych. Nie bez znaczenia jest również fakt, że bazę danych utworzoną za pomocą tego programu, można w dowolnej chwili rozbudować lub zmodyfikować (oczywiście, jeżeli ma się do tego uprawnienia), a nawet przystosować do pracy z Microsoft SQL Server.⁵

Microsoft Access 2.0 oferuje szeroką gamę narzędzi służących zwiększeniu wydajności produkcyjnej każdego użytkownika. Zwiększona zgodność z produktami z zestawu Microsoft Office oraz nowe formy kreatorów (Wizards) oraz podpowiedzi CueCards upraszczają wykonywanie pracy. Na przykład Table Wizard tworzy różnorodne tabele, ułatwiając pokonanie początkowych trudności z tworzeniem bazy. Twórcy programów korzystają z tej samej techniki w celu szybkiego stworzenia aplikacji.

W dotychczasowych wersjach Microsoft Access znajdują się elementy wykazane w tabeli 7.

Tabela 7: Elementy zawarte w starszych wersjach MS Access⁶

Kreator bazy danych	Automatycznie tworzy tabele, kwerendy (zapytania), formularze i raporty z ponad 20 różnych szablonów, z możliwością późniejszej ich modyfikacji.
Kreator prostego zapytania (kwerendy)	Sortuje informacje z różnych tabel bazy danych oraz określa sposób ich prezentacji w odpowiedzi na Twoje zapytanie.
Filtrowanie według wyboru	Filtrowanie i wyświetlanie danych zgodnie z wyborem.
Filtrowanie według formularza	Wprowadzenie kryteriów poszukiwania w formularzach oraz prezentacja tylko tych danych, które spełniają wymagania.
Kreator tabeli	Automatycznie identyfikuje relacje w niestrukturalnych danych, a następnie reorganizuje je w relacyjną bazę danych.
Kreatorzy formularzy i raportów	Generują raporty i formularze tak, aby umożliwić umieszczenie wszystkich planowanych danych.
Publikacja w WWW	Przekształcenie danych i raportów do formatu HTML, aby tworzyć interakcyjne strony baz danych i dzielić się nimi przez intranet lub Internet.
Odnosiniki dostępne przez jedno kliknięcie myszy	Przemieszczanie się między raportami, formularzami i tabelami bazy danych Microsoft Access, dokumentami Microsoft Office, oraz między serwerami intranetu lub Internetu.
Kreator importu / eksportu	Krok po kroku, prowadzi Cię przez procesy importu lub eksportu danych z bazy do innych formatów, pokazując jednocześnie jak dane te będą wyglądać po przekształceniu.

⁵ Sikora M.; New Accessmaster; <http://wr.atr.bydgoszcz.pl/kiz/access/>

⁶ Sikora M.; New Accessmaster; <http://wr.atr.bydgoszcz.pl/kiz/access/>

Replikacja bazy danych	Pozwala na zdalne odświeżanie danych w bazie przez wielu użytkowników oraz na synchronizację tych zmian przez Internet.
Zoptymalizowany 32-bitowy kod aplikacji	Szybsza praca aplikacji Microsoft Access 97, dzięki mniejszym formularzom, lepszej kompilacji i manipulacji 32-bitowymi danymi.
Microsoft Visual Basic® for Applications	Bogate środowisko IDE, zawierające między innymi: kontekst kodowany kolorem, poprawione okno odpluskwania (debugging), przeglądanie obiektów, obsługa sterowników ActiveX oraz odnośniki do innych aplikacji Microsoft Office.
Kreator analizy wydajności	Dokonuje analizy bazy danych oraz proponuje najlepsze sposoby optymalizacji szybkości pracy oraz automatycznie dokonuje zmian.
Programowalne paski poleceń	Oferuje pełną kontrolę nad elementami sterującymi interfejsu (paski zadań i menu) poprzez udostępnienie jednego, centralnego typu dla całej rodziny aplikacji Office 97.

System Microsoft Access 2.0 wyposażony jest w kilka nowych cech:

- Baza zawiera ponad 30 kreatorów (Wizards), łącznie z bardzo ważnym Table Wizard. Wystarczy odpowiedzieć na kilka prostych pytań, kliknąć na przycisku Finish i tabela gotowa. Query Wizards tworzy skomplikowane zapytania.
- Command Button Wizard jest w stanie utworzyć ponad 40 typów przycisków, do których dołącza odpowiedni kod. Wystarczy podać, jaki typ przycisku jest potrzebny i reszta odbywa się automatycznie. Obecnie można tworzyć przyciski do przygotowania formularzy, znajdowania rekordów i wielu innych zadań, bez potrzeby pisania kodu. Twórcy oprogramowania mogą przygotowany kod dowolnie modyfikować.
- Narzędzie do tworzenia zależności graficznych (Graphical Relationship Design) pozwala na graficzne utworzenie struktury bazy przez przeciąganie i upuszczanie.
- Narzędzie do generacji wyrażeń (Expression Builder) wykonuje pracę programową. W celu stworzenia skomplikowanego zapytania wystarczy wybrać odpowiednie elementy z listy.

System oferuje lepsze niż dotychczas możliwości dla programisty:

- Polepszony model obsługi zdarzeń, dający precyzyjną kontrolę nad zdarzeniami i ich właściwościami.
- Wbudowane kompletne środowisko opracowania aplikacji zawierające debugger, moduł do tworzenia dokumentacji DataBase Documentor i zarządzania dodatkowymi elementami Add-in Manager.
- Potężny motor bazy Microsoft Access, teraz zawiera nieograniczoną liczbę kaskadowych uaktualnień i usuwań, zapewniając integralność danych, przez co oszczędza się setki wierszy kodu.
- Technika optymalizacji zapytań Rushmore przyspiesza ich szybkość wykonywania. Na przykład zapytanie do bazy złożonej ze 100 tys. rekordów daje 1052 rekordy wyniku w czasie 2,85 s.
- Access Basic, sterowany zdarzeniami język programowania zapewnia dostęp do wszystkich obiektów w bazie.

- Polepszone możliwości obsługi aplikacji typu klient/serwer i dostępu do danych na innych komputerach pozwalają na korzystanie z dowolnych źródeł danych, zwłaszcza w sieciach z serwerem.
- Twórcy aplikacji mogą także rozszerzyć możliwości Microsoft Access przez wykorzystanie obiektów OLE 2.0 i elementów kontrolnych OLE. Użycie OLE 2.0 umożliwia integrację z innymi aplikacjami, np. z zestawu Microsoft Office.

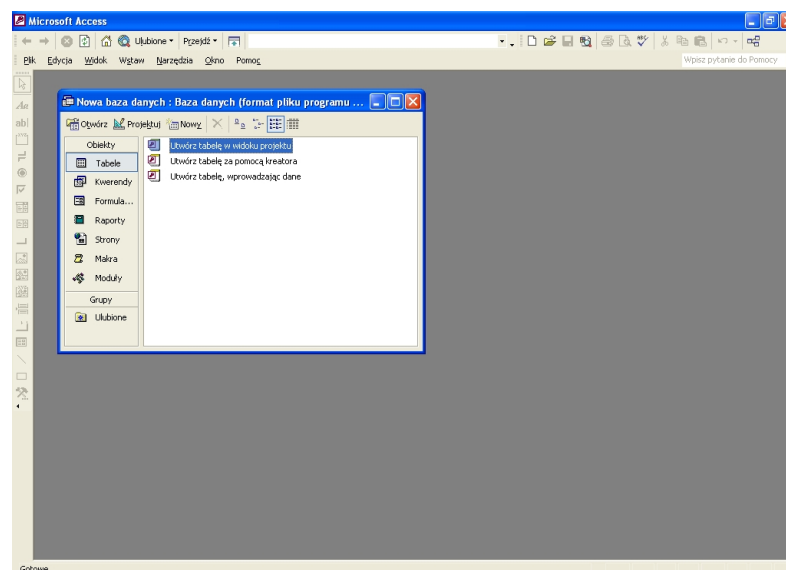
3.3.2. Budowa bazy danych przy użyciu programu MS Access

Przystępując do budowy bazy danych należy podjąć szereg działań, które można podzielić na następujące etapy:

- × Definiowanie struktury danych:
 - ▶ definiowanie tablic (struktura, typy danych, klucz główny, ograniczenia, indeksy);
 - ▶ określenie relacji (więzów integralności);
 - ▶ wprowadzenie danych testowych
- × Wykonanie formularzy:
 - ▶ zaprojektowanie zapytania na którym będzie oparty formularz;
 - ▶ wykonanie graficznego układu formularza;
 - ▶ dodatkowe oprogramowanie formularza.
- × Wykonanie raportów:
 - ▶ zaprojektowanie zapytania dla raportu;
 - ▶ wykonanie graficznego układu raportu;
 - ▶ określenie sortowania i grupowania;
 - ▶ dodatkowe oprogramowanie raportu.
- × Wykonanie formularzy i sekcji sterujących (przyciski).
- × Dodatkowe oprogramowanie aplikacji.

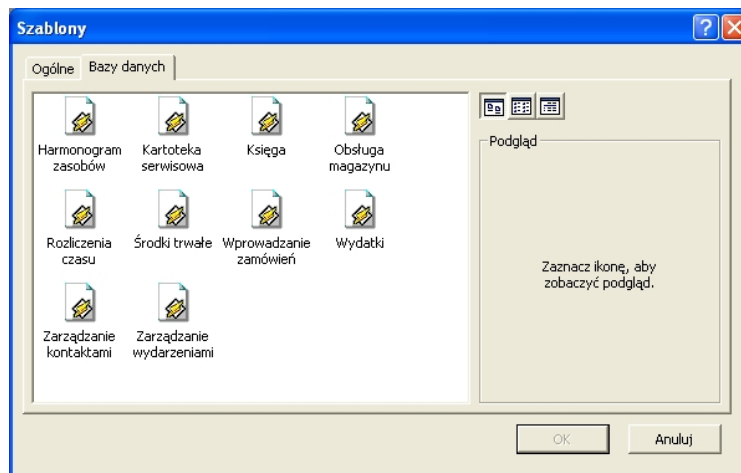
Rozpoczynając pracę z programem Access należy otworzyć okno pozwalające na utworzenie elementów nowej bazy danych, bądź na otwarcie już istniejącej.

Rysunek 17: Pulpit nowo otwartej bazy danych w programie Access



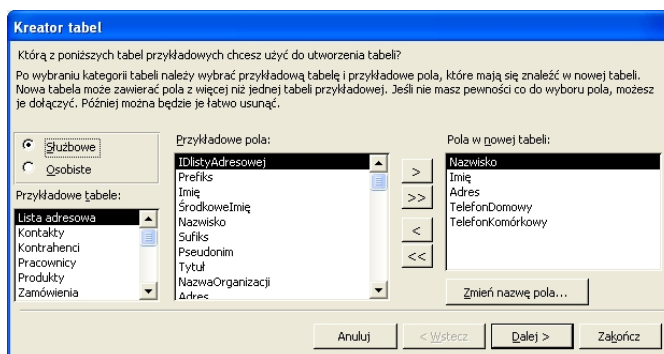
Tworząc nową bazę danych można skorzystać z gotowych szablonów, które proponuje program. Dzięki nim w sposób bardzo szybki można uzyskać bazę z określonej dziedziny, którą w dalszej kolejności można zmodyfikować stosownie do potrzeb użytkownika bazy.

Rysunek 18: Szablony baz danych proponowane przez Access



Tabele bazy danych można też tworzyć, korzystając z kreatora tabel, który krok po kroku przeprowadza przez kolejne etapy tworzenia tabeli. Kreator proponuje przykładowe pola, które można zamieścić w nowej tabeli.

Rysunek 19: Okno dialogowe kreatora tabel Access



Innym sposobem na utworzenie nowej tabeli jest tworzenie jej w widoku projektu. W oknie wpisuje się nazwy pól przyszłej tabeli oraz określa się typy danych, które w tych polach będą zapisywane. Możliwe jest wpisanie następujących typów danych:

- × **Tekst** - jest to pole przeznaczone do wpisywania tekstu, liczb, znaków ANSI [Alt+numer]. Jego maksymalna długość to 255 znaków.
- × **Memo** - pole przeznaczone dla dłuższych tekstów, jak np. długi opis, bądź instrukcja. Maksymalna długość 64000 znaków.
- × **Liczba** - pole typu liczbowego o 5 rodzajach:
 - ▶ bajt (0 - 255),
 - ▶ liczba całkowita (od -32768 do +32767),
 - ▶ liczba całkowita długa (4 bajty),
 - ▶ pojedyncza precyzja (liczba rzeczywista - 4 bajty),
 - ▶ podwójna precyzja (liczba rzeczywista - 8 bajtów).
- × **Data/Godzina** - pole to pozwala na przechowywanie daty bądź godziny w kilku formatach. Ważną cechą jest, że rok może zawierać się w granicach 100-9999.

- * **Waluta** - jest to pole o zwiększonej dokładności zaokrągleń. Jak sama nazwa wskazuje, jest ono przeznaczone dla dokładnych danych, pozwala ono na szybkie operacje na danych (sumowanie, mnożenie).
- * **Autonumerowanie** - pole typu licznikowego, program sam generuje kolejny numer, przy czym numer ten jest niepowtarzalny i nie można go edytować.
- * **Tak/Nie** - pole, którego wartość może być Tak lub Nie. Jest ono przydatne dla zaznaczenia jakiejś wartości w bazie, jak na przykład tego, czy pracownik ma własny samochód. Fizycznie w tabeli opcja Tak ma wartość (-1), a opcja Nie wartość (0).
- * **Obiekt OLE** - przeznaczone do osadzania bądź dołączania obiektów OLE jak np. obrazek BMP.⁷

W polu „Typ danych” można też wstawić **hiperłącze** lub **odnośnik** za pomocą kreatora.

Widok projektu tabeli daje szeroką możliwość formatowania tworzonych pól tabeli i danych w nich zawartych:

- * **Rozmiar pola** - określa długość pola. W przypadku pola tekstowego maksymalną wartością jest 255.
- * **Format** - określa format pola, z tym, że zmieniają się możliwe opcje w zależności od typu danej. Dla liczby może to być to liczba ogólna, waluta, stałoprzecinkowy, wykładniczy, a dla daty/godziny może być np. 19 VI 94 lub Niedziela, czerwca 19, 1997.
- * **Maska wprowadzania** - określa sposób wprowadzania wszystkich danych do danego pola. Jest to podstawowe kryterium dotyczące wprowadzanych danych. Do zapisania maski wprowadzania używane są poniższe znaki:
 - ▶ 0 - oznacza obowiązkowe wprowadzenie cyfry
 - ▶ 9 - opcjonalne wprowadzenie cyfry
 - ▶ L - obowiązkowe wprowadzenie litery
 - ▶ ? - opcjonalne wprowadzenie litery
 - ▶ A - obowiązkowe wprowadzenie cyfry lub litery
 - ▶ a - opcjonalne wprowadzenie cyfry lub litery
 - ▶ & - dowolny znak lub spacja, wprowadzenie obowiązkowe
 - ▶ < - litery za tym znakiem są przekształcane na małe
 - ▶ > - litery za tym znakiem są przekształcane na duże
 - ▶ \ - znaki za \ będą wyświetlane jako alfanumeryczne
 - ▶ ! - znaki będą wypełniane od prawej strony w polu
 - ▶ Hasło - tworzy pole, w którym każdy wpisany znak widziany jest jako gwiazdka.
- * **Tytuł** - jest to wartość, która będzie pojawiała się jako etykieta podczas tworzenia formularzy. Jeżeli nie podamy tutaj nic, to jako etykieta pojawi się w formularzu nazwa pola.
- * **Wartość domyślna** - jest to wartość wstawiana podczas dodawania nowego rekordu. Jeżeli podczas wpisywania będziemy chcieli wprowadzić inną wartość niż wartość domyślna, to wystarczy przejść na pole i wpisać odpowiednią wartość.
- * **Reguła poprawności** - jest to kolejny warunek dotyczący wpisywanych danych. Określa na bieżąco po wpisaniu rekordu czy dana jest poprawna, a w przypadku wprowadzenia niepoprawnej danej wyświetli komunikat. W tym polu można wprowadzać różne warunki:
 - ▶ np: =„Warszawa” - pole musi być równe Warszawa;
 - ▶ In („Warszawa”, „Poznań”, „Wrocław”) - pole musi zawierać jedną z tych wartości; „Poznań” OR „Warszawa” OR „Wrocław”;

⁷ Daniszewski Piotr Projektowanie bazy danych <http://www.enter.pl/ent97.11/access.htm>

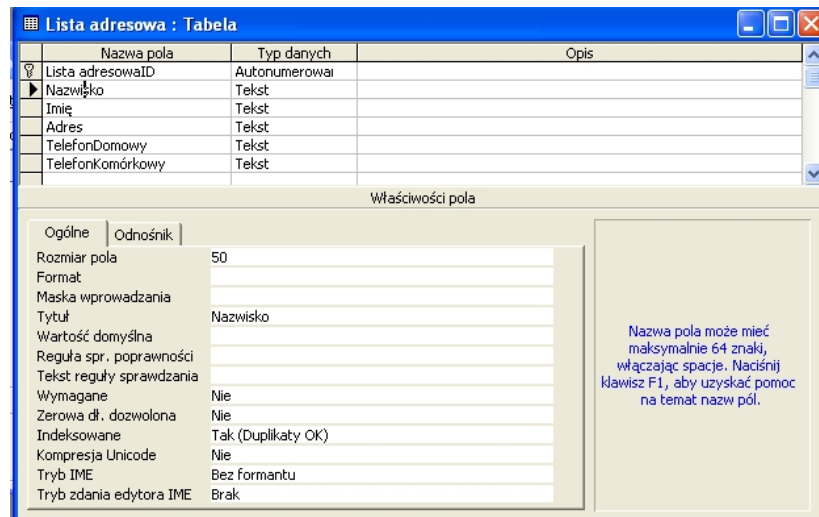
- ▶ ≥ 23 And < 34 - pole musi być większe lub równe 23 i mniejsze niż 34;
- ▶ Between 100 And 200 - liczba musi być w zakresie 100-200 włącznie.

W celu stworzenia odpowiedniej reguły poprawności można skorzystać z wbudowanego generatora wyrażeń do określenia jakiejś wartości, np. Matematycznej, lub wpisać odpowiednią funkcję Visual Basica.

- * **Komunikat o błędzie** – tekst, jaki ma się pojawić w oknie komunikatu w przypadku wprowadzenia błędnego rekordu związanego z tym, że wartość wpisana nie pasuje do maski wprowadzania, bądź też nie spełnia kryteriów reguły poprawności.
- * **Wymagane** - określa, czy wpisanie danych do tego pola jest wymagane. Do wyboru dwie opcje Tak/Nie.
- * **Zerowa długość** - określa, czy można dopuścić ciągi znaków o zerowej długości.
- * **Indeksowane?** - do wyboru są 3 opcje: Nie, Tak(Duplikaty OK), Tak (Bez duplikatów). Indeksy są bardzo ważnym elementem w dużych bazach danych. Są to dodatkowe informacje w postaci liczb (nie dostępne dla użytkownika), określające pozycję dla każdego rekordu w danym polu. Indeksowanie polega na nadawaniu numerów każdemu rekordowi podczas wpisywania. Numer ten jest adekwatny do jego wartości i w przypadku pola tekstowego jest to jego pozycja według sortowania alfabetycznego. Stosowanie indeksów powoduje zwiększenie szybkości działania sortowania dla pól mających indeksy, jednakże spowalnia pracę podczas wprowadzania nowych rekordów, gdyż za każdym razem należy zmodyfikować indeksy dla tego pola. Czas sortowania pól z indeksami jest dużo szybszy niż dla pól bez indeksów. Wskazane jest tworzenie indeksów dla pól, od których oczekuje się szybkiego sortowania. Niektóre pola nie mogą mieć indeksów (np. pole memo czy obiekt OLE).
- * **Miejsca dziesiętne** - określa liczbę miejsc po przecinku dla pola liczbowego.

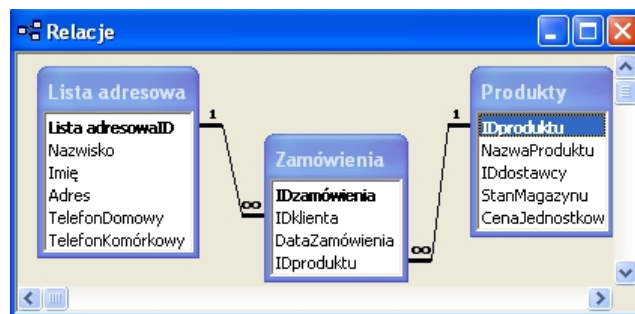
8

Rysunek 19: Widok projektu tabeli Access



Po zaprojektowaniu różnych tabel bazy danych, potrzebny jest sposób na powiązanie zawartych w nich informacji. Pierwszym krokiem w tym procesie jest zdefiniowanie relacji pomiędzy tabelami. Gdy się to zrobi, można tworzyć kwerendy, formularze i raporty pozwalające wyświetlać za jednym razem informacje z różnych tabel.

Rysunek 20: Widok relacji utworzonych w bazie danych Access

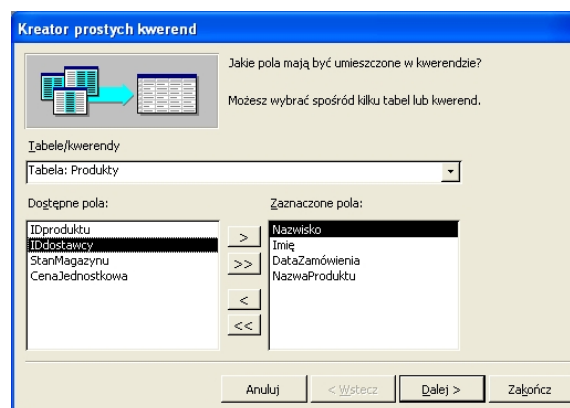


Obok tabel zawierających dane podstawowymi obiektami definiującymi schematy wyszukiwania pożądanej informacji są kwerendy (*ang. query*). W MS Access istnieją dwa języki definiowania kwerend, język SQL (*ang. Structured Query Language*) oraz język QBE (*ang. Query By Example*).

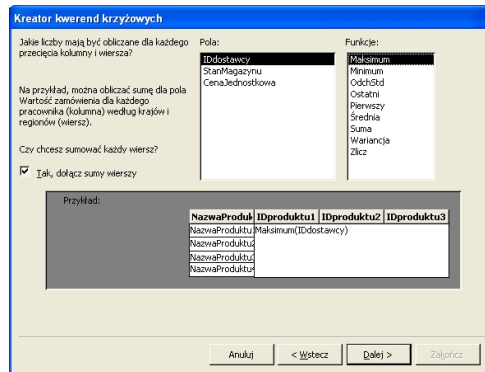
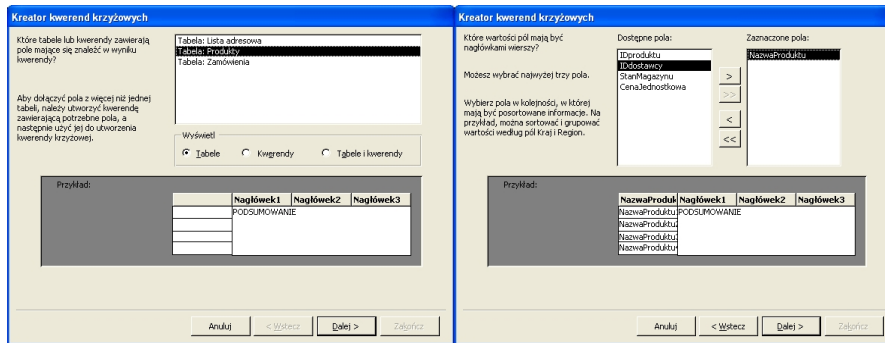
Kwerendy umożliwiają oglądania danych we właściwej kolejności. Umożliwiają również przeprowadzanie obliczeń na podstawie zebranych danych w celu tworzenia źródła danych dla formularzy, raportów i innych kwerend i aby wprowadzić zmiany globalne w tabelach i tworzyć nowe tabele. Przy uruchamianiu większości kwerend lub stosowaniu filtrów, Access zbiera w zestawieniu dane, o które pytamy. Chociaż to zestawienie wygląda i działa jak tabela, jest teraz dynamicznym, "żywym" widokiem jednej lub więcej tabel. W programie MS Access kwerendy można tworzyć dwoma sposobami:

- × Utworzyć za pomocą kreatora kwerend,
- × Utworzyć kwerendę w widoku projektu.

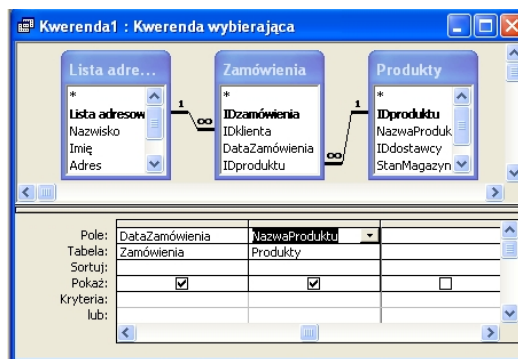
Rysunek 21: Okno dialogowe kreatora prostych kwerend Access



Rysunek 22: Okna dialogowe kreatora kwerend krzyżowych Access



Rysunek 23: Widok kwerendy utworzonej w widoku projektu

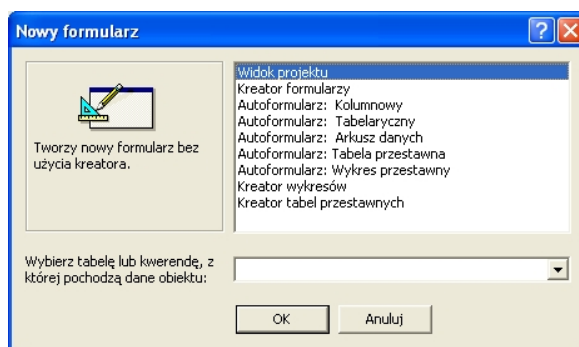


Przystępując do tworzenie formularza należy :

- ▶ zdefiniować zapytania
- ▶ użyć *kreatora formularzy*
- ▶ przeprowadzić modyfikację typów pól i wyglądu formularza
- ▶ przeprowadzić modyfikację własności formularza i pól
- ▶ utworzyć przyciski funkcyjnych (z użyciem *kreatora formantów*)
- ▶ oprogramowanie zdarzeń formularza

Program MS Access stwarza możliwość utworzenia formularza za pomocą kreatora, który pozwala na przygotowanie różnych formularzy w tym nawet i wykresów.

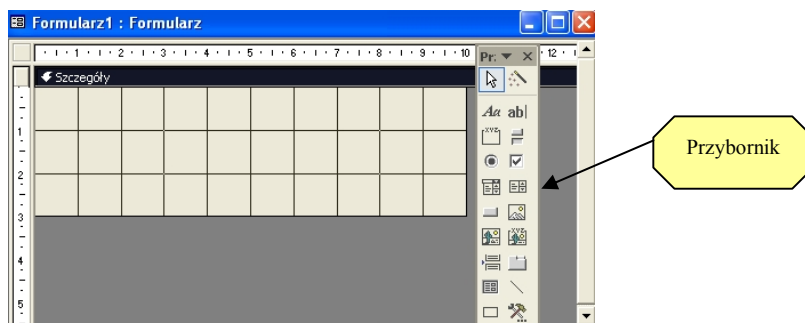
Rysunek 24: Okno do tworzenia nowego formularza



Istnieje też możliwość tworzenia formularza w widoku projektu, co stwarza dość duże pole do popisu dla osoby projektującej formularz. Pomocą przy projektowaniu jest przybornik. Na formularzach dostępne są następujące obiekty:

- ▶ etykiety
- ▶ pola tekstowe
- ▶ linie itp.
- ▶ grupy opcji i przycisków
- ▶ pola kombi, listy, wyboru
- ▶ obiekty OLE stałe, powiązane
- ▶ podformularze
- ▶ wykresy
- ▶ Inne: kalendarz itp.

Rysunek 25: Widok formularza utworzonego w widoku projektu

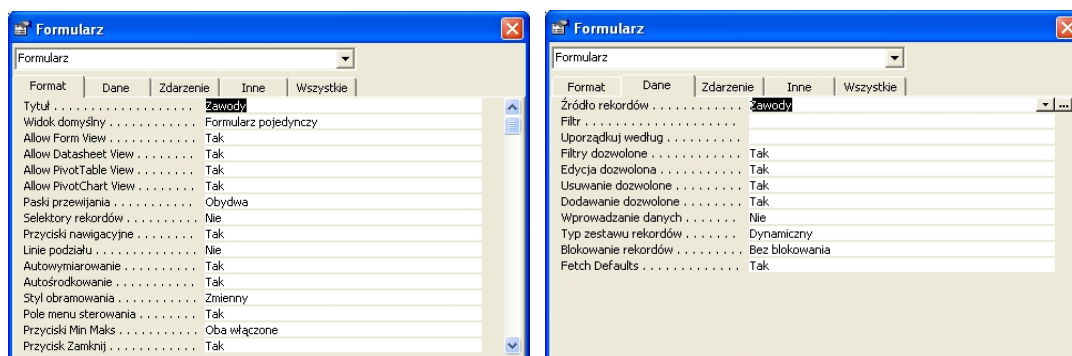


Formularze i poszczególne ich pola posiadają określone własności, które można kształtować w zależności od potrzeb są między innymi własności:

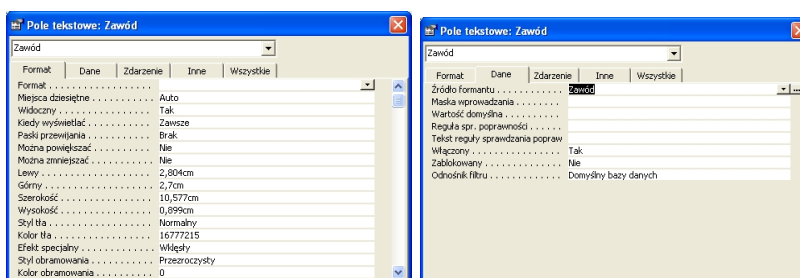
- × formularza
 - ▶ Dane: *źródło rekordów*
 - ▶ Wygląd: *widok domyślny*
 - ▶ Inne: *edycja dozwolona, usuwanie dozwolone, dodawanie dozwolone, modalny*
- × pól

- ▶ Dane: *źródło formantu* (może być wyrażenie - obliczenia automatyczne), *wartość domyślna*, *reguła poprawności*, *komunikat o błędzie*
- ▶ Wygląd: *widoczny*, *kiedy wyświetlać*, własności określające czcionkę, justowanie, kolory i obramowanie
- ▶ Inne: *nazwa*, *indeks kolejności dostępu*, *zatrzymanie przechodzenia*
- ▶ *indeks kolejności dostępu*, *zatrzymanie przechodzenia*

Rysunek 26: Okno właściwości formularza

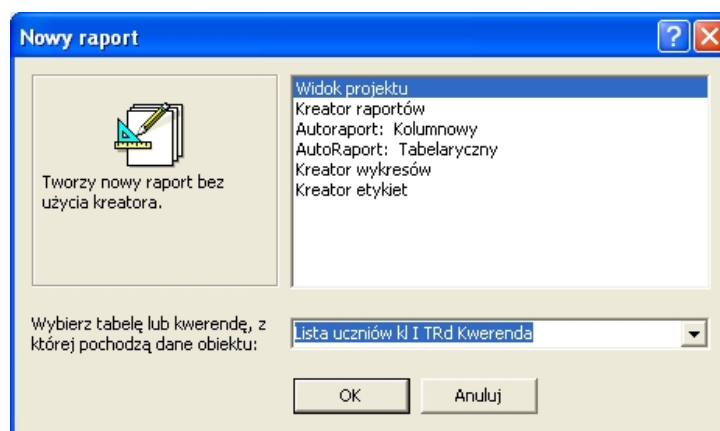


Rysunek 27: Okno właściwości pola tekstowego

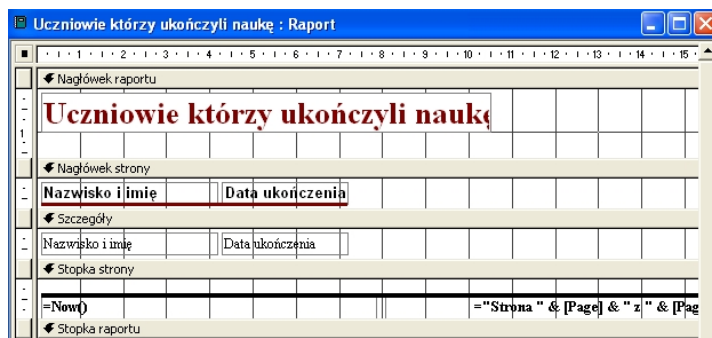


Podobnie jak i dotychczasowe elementy bazy danych tak też i formularze mogą być tworzone w dwojaki sposób, zarówno z wykorzystaniem kreatora jak też i w widoku projektu

Rysunek 28: Okno tworzenia raportu przy pomocy kreatora



Rysunek 29: Okno tworzenia raportu w widoku projektu



Makropolecenia (makra) są źródłem bardzo wielu usprawnień i ułatwień w obsłudze bazy danych. Są one zapisem sekwencji operacji wykonywanych przy obsłudze bazy. Sekwencje te mogą być przypisywane do przycisku, po naciśnięciu, którego następuje uruchomienie makra. Makro może służyć do otwierania formularzy, tabel i kwerend oraz inicjowania innych działań.

Rysunek 30: Okno tworzenia makra

