



MS Access

Średnio zaawansowany

Materiały szkoleniowe

Wersja demonstracyjna



KONTAKT

Adres

Expose sp. z o. o.
ul. Skierniewicka 10a
01-230 Warszawa



Telefon

+ 48 22 465 88 88
+ 48 22 240 19 99

Online

biuro@expose.pl
www.expose.pl
www.chcesieuczyc.pl

Konspekt kursu MS Access – Podstawowy

1. Zarządzanie tabelami
 - Zaawansowane właściwości pól tabeli
 - Relacje – opcje zaawansowane
 - Sprzężenia między tabelami
 - Optymalizacja projektu tabeli
 - Analiza wydajności
2. Tworzenie kwerend
 - Kwerendy wybierające, usuwające i aktualizujące
 - Wykorzystanie operatorów AND i OR
 - Parametryzacja zapytań
 - Łączenie danych za pomocą operacji UNION
 - Sprzężenia wewnętrzne i zewnętrzne
 - Obliczenia w kwerendach
3. Praca z kwerendami funkcjonalnymi
 - Kwerendy aktualizujące
 - Kwerendy dołączające
 - Usuwanie danych
 - Kwerendy tworzące tabelę
4. Podstawy pracy z językiem SQL w programie Microsoft Office Access
 - Kwerendy SQL (konstrukcja SELECT, klauzule WHERE, ORDER BY, INSERT, UPDATE, DELETE)
 - Kwerenda składająca
 - Kwerenda definicji danych
5. Zaawansowane projektowanie formularzy i raportów
 - Formularze oparte o kwerendy
 - Praca z podformularzami
 - Formularze wielostronicowe
 - Wykorzystywanie szablonów raportów
 - Raporty wielokolumnowe
 - Grupowanie danych
 - Kontrolki ActiveX
 - Generowanie raportów w oparciu o kwerendy
6. Bezpieczeństwo i usprawnienia baz danych
 - Zabezpieczanie bazy hasłem
 - Panel sterowania bazą danych
 - Zestaw makropoleceń usprawniających pracę z bazą

7. Podstawy języka Visual Basic

- Automatyzacja aplikacji za pomocą makr
- Tworzenie makr związanych z elementami interfejsu użytkownika
- Środowisko projektowe
- Zmienne i stałe
- Deklarowanie zmiennych
- Funkcje i procedury
- Przykłady wykorzystania VBA w MS Access

1. Wprowadzenie do baz danych

Definicja bazy danych

Zbiór danych zapisanych w ściśle określony sposób w strukturach odpowiadających założonemu modelowi danych. W potocznym ujęciu obejmuje dane oraz program komputerowy wyspecjalizowany do gromadzenia i przetwarzania tych danych. Program taki (często pakiet programów) nazywany jest "Systemem zarządzania bazą danych" (ang. DataBase Management System, DBMS). W ścisłej nomenklaturze baza danych oznacza zbiór danych, który zarządzany jest przez system DBMS.

Bazy danych operują głównie na danych tekstowych i liczbowych, lecz większość współczesnych baz umożliwia przechowywanie danych binarnych typu: grafika, muzyka itp.

Program Access pozwala wykonywać następujące czynności:

- Dodawanie nowych danych, takich jak nowy element inwentarza, do bazy danych
- Edytowanie istniejących danych w bazie danych, na przykład zmienianie bieżącej lokalizacji elementu
- Usuwanie informacji, na przykład po sprzedaniu elementu lub wycofaniu go ze sprzedaży
- Organizowanie i przeglądanie danych na różne sposoby
- Udostępnianie danych innym osobom za pomocą raportów, wiadomości e-mail, sieci intranet lub Internet

Geneza baz danych

- Najwcześniejsze znane użycie terminu baza danych miało miejsce w listopadzie 1963, kiedy odbyło się sympozjum pod nazwą "Development and Management of a Computer-centered Data Base", sponsorowane przez System Development Corporation. Termin ten stał się powszechnie używany w Europie we wczesnych latach siedemdziesiątych XX wieku.
- Pierwszy system zarządzania baz danych został opracowany w latach sześćdziesiątych XX wieku. Pionierem był Charles Bachman. Wczesne opracowanie Bachmana pokazywały, że jego celem było bardziej efektywne użycie nowych urządzeń bezpośredniego dostępu do składowanych danych, które wtedy zaczynały być dostępne. Jak dotąd, przetwarzanie danych było oparte na kartach dziurkowanych i taśmach magnetycznych.
- W 1970 E. F. Codd zaproponował relacyjny model danych. Krytykował on istniejące modele danych za mieszanie abstrakcyjnego opisu struktury informacyjnej z opisami mechanizmów fizycznego dostępu. Jednak przez dłuższy czas model relacyjny pozostawał tylko w sferze rozważań akademickich. Pierwsze komercyjne rozwiązania, Oracle i DB2 nie były dostępne aż do roku około 1980. Natomiast pierwszym udanym produktem tego typu dla mikrokomputerów był dBASE dla systemów operacyjnych CP/M i PC-DOS/MS-DOS.
- W latach dziewięćdziesiątych XX wieku, uwaga badaczy przesunęła się w kierunku obiektowych baz danych. Były one z powodzeniem zastosowane tam, gdzie konieczne była obsługa bardziej skomplikowanych danych niż mogły z nimi dać łatwo rady relacyjne bazy danych. Część z tych idei była zaadaptowana przez producentów relacyjnych baz danych. Te lata były też okresem rozprzestrzeniania się baz danych Open Source, takich jak PostgreSQL i MySQL.
- Pierwsze lata XXI wieku są okresem dużego zainteresowania bazami danych XML. W tym czasie, podobnie jak to było w przypadku obiektowych baz danych, powstało sporo nowych firm-producentów tych baz, ale kluczowe ich elementy są wbudowywane także w istniejące relacyjne bazy danych.

Typy baz danych

Bazy danych można podzielić według struktur organizacji danych, których używają:

- Bazy proste :
 - o bazy kartotekowe
 - o hierarchiczne bazy danych
- Bazy złożone :
 - o bazy relacyjne
 - o bazy obiektowe
 - o bazy relacyjno-obiektowe
 - o strumieniowe bazy danych
 - o temporalne bazy danych

Bazy kartotekowe

W bazach kartotekowych każda tablica danych jest samodzielnym dokumentem i nie może współpracować z innymi tablicami. Z baz tego typu korzystają liczne programy typu: książka telefoniczna, książka kucharska, spisy książek, kaset i inne. Wspólną cechą tych baz jest ich zastosowanie w jednym wybranym celu.

Sieciowe bazy danych

Model historyczny, pozwalał tylko na związki binarne; wiele do jeden.

Hierarchiczne bazy danych

Przykładem hierarchicznej bazy danych jest opracowana przez IBM baza IMS (ang. Information Management System).

Bazy relacyjne

W bazach relacyjnych wiele tablic danych może współpracować ze sobą (są między sobą powiązane). Bazy relacyjne posiadają wewnętrzne języki programowania, wykorzystujące zwykle SQL do operowania na danych, za pomocą których tworzone są zaawansowane funkcje obsługi danych.

Bazy obiektowe

W bazach obiektowych dane przechowywane są w strukturach obiektowych (zdefiniowanych jako klasy). Koncepcje akademickie dotyczące baz obiektowych były najbardziej popularne w latach 90. Współcześnie popularność tego tematu zmalała, choć prace badawcze nad nimi nadal trwają, a na rynku pojawiły się obiektowe SZBD (np. Versant, db4o, LoXIM). Prace nad obiektowymi bazami danych ponowiło międzynarodowe konsorcjum OMG

Bazy relacyjno-obiektowe

Bazy relacyjno-obiektowe pozwalają na manipulowanie danymi jako zestawem obiektów, posiadają jednak bazę relacyjną jako wewnętrzny mechanizm przechowywania danych.

Strumieniowe bazy danych

Baza danych, w której dane są przedstawione w postaci zbioru strumieni danych

Systemy zarządzania baz danych i ich cele

System zarządzania bazą danych, SZBD (ang. Database Management System, DBMS) nazywany też serwerem baz danych, SBD to oprogramowanie bądź system informatyczny służący do zarządzania komputerowymi bazami danych. Systemy baz danych mogą być sieciowymi serwerami baz danych lub udostępniać bazę danych lokalnie.

Cele systemów zarządzania bazą danych:

- Zarządzanie plikami
 - dodawanie nowych plików do bazy,
 - usuwanie plików z bazy danych,
 - modyfikowanie struktury istniejących plików,
 - wstawianie nowych danych do istniejących plików,
 - aktualizowanie danych w istniejących plikach,
 - usuwanie danych z istniejących plików.
- Wyszukiwanie informacji
 - wydobywanie danych z istniejących plików w odpowiedzi na żądania użytkowników,
 - wydobywanie informacji dla potrzeb programów użytkowych.
- Zarządzanie bazą danych
 - tworzenie i monitorowanie użytkowników bazy danych,
 - ograniczanie dostępu do plików w bazie danych,
 - monitorowanie działania bazy danych. Podstawowe pojęcia stosowane w bazach danych

Podstawowe pojęcia stosowane w bazo znawstwie

Tabela jest po prostu opakowanie na dane w naturalnej postaci – odpowiednikiem teczki zawierającej arkusze w fizycznych systemach. Każda tabela w bazie danych Accessa zawiera informacje opisujące oddzielny podmiot – na przykład osobę lub produkt, a dane są zorganizowane w postaci wierszy i kolumn.



Cont	Contact Type	First Name	Last Name	Company	Address	
1	Buyer	John	Jones	Nopa Auto Parts	11253 Main Street	Springfield
2	Seller	Hank	Masters	Jiffy Auto Sales	623 Field Road	Springfield
3	Both	Larry	Minkler	All Start Autos	971 E Main St	Detroit
4	Both	Joe	Hammerman	Columbia Chevy	105 Main Street	Arlington
5	Buyer	Cary	James	James Auto Parts	59 South Street	West Palm Beach
6	Buyer	Mark	Uno	Fillion Sales And Service	8908 North Park Ave	Ft Lauderdale
7	Buyer	Brandon	Aley	Tip Top Chevy	1916 Erickson Drive	Fairbanks
8	Both	Michael	Dennis	Newbury Auto	75 Main Street	Macon
9	Both	Mark	Martin	Peekskill Sales	51 Tolland Tnpk	Savannah
10	Both	Karl	Johnson	KJ Auto Repair	350 Broadway St	Augusta
11	Both	William	Gleason	R & G Monda Inc	196 East Street	Derby
12	Both	Alex	Tomaso	Tires National	46 School Street	Stratford
13	Both	Karla	Hayes	Hayes Auctions	54 E Center Street	Granby
14	Seller	Teresa	Aikins	Middletown Auto Parts	100 Northfield Dr	Middletown
15	Both	John	Marino	Bill Thomas Sales, Inc	986 Buckingham St	Valdosta
16	Both	Donald	Peterson	Yantic Auto Parts	8 Oak Street	Yantic
17	Seller	Dennie	Parkson	ACC Car Sales	963 New England Turnpike	Atlanta
18	Buyer	Ann	Bond	A-1 Auto Sales	54 South Main	Ft Lauderdale
19	Seller	Joe	Crook	Main Street Used Auto Parts	61 North Main Street	Key West
20	Both	Jeffrey	Lan	LAN Trucking	108 Thomas Road	North Branch
21	Buyer	Matt	Smith	ABC Trucking	7 Depot Rd	Orlando
22	Both	David	Smith	Fordman Colonial Times	123 Federal Street	Quincy
24	Both	Cindy	Casey	Circle Auto Sales	123 South Street	Newington
25	Both	Karen	Bailey	Sammy Ford	59 West Church	Westborough
26	Both	Alvin	Schindler	Pine Plains BMW	Rt 9	Atlanta
27	Both	John	Bright	Pleasantville Monda Inc	46 Pleasantville RD	Arlington
28	Both	George	Sampson	Newton Monda Inc	560 Broadway	Salem

Rekordy i pola. Jak widać na rysunku arkusz danych jest podzielony na **wiersze** (poziome) zwane **rekordami** oraz **kolumny** (pionowe) zwane **polami**, gdzie pierwsze wiersz (nagłówki wszystkich kolumn) zawierają **nazwy pól** w bazie danych. Każdy wiersz reprezentuje pojedynczy rekord zawierający pola charakteryzujące rekord, W systemach fizycznych wiersze można porównać do pojedynczych formularzy (kartek papieru), natomiast pola do pustych pól formularz, których zawartość wypełnia się samemu.

Wartości. Na przecięciu wiersza (rekordu) i kolumny (pola) znajduje się wartość – właściwy element danych. Na przykład „John”, imię w pierwszym rekordzie, reprezentuje pojedynczą wartość danych.

Formularz są też określane jako „ekrany wprowadzania danych”. Stanowią one interfejs do pracy z danymi i często zawierają przyciski umożliwiające wykonywanie różnych poleceń. Można utworzyć bazę danych bez korzystania z formularzy, edytując dane w arkuszach danych tabel. Większość użytkowników baz danych woli jednak używać formularzy do przeglądania, wprowadzania i edytowania danych przechowywanych w tabelach.

Raporty służą do podsumowywania i wyświetlania danych zawartych w tabelach. Zazwyczaj raport umożliwia odpowiedź na określone pytanie dotyczące danych, na przykład „Ile pieniędzy otrzymaliśmy od poszczególnych klientów w tym roku?” lub „W jakich miastach mają swoje siedziby nasi klienci?” Każdy raport można sformatować w sposób zwiększający czytelność danych.

Kwerendy stanowią podstawowe narzędzie baz danych, które może spełniać wiele różnych funkcji. Najczęściej są one używane do pobierania określonych danych z tabel. Często dane, które mają być pobrane, znajdują się w kilku tabelach – kwerendy pozwalają przeglądać takie dane w jednym

arkusza danych. Poza tym nie zawsze jest konieczne wyświetlanie wszystkich rekordów, dlatego przy użyciu kwerend można określać odpowiednie kryteria „filtrowania” danych, aby uzyskać dostęp tylko do potrzebnych rekordów. Kolejnym zastosowaniem kwerend jest dostarczanie danych dla formularzy i raportów.

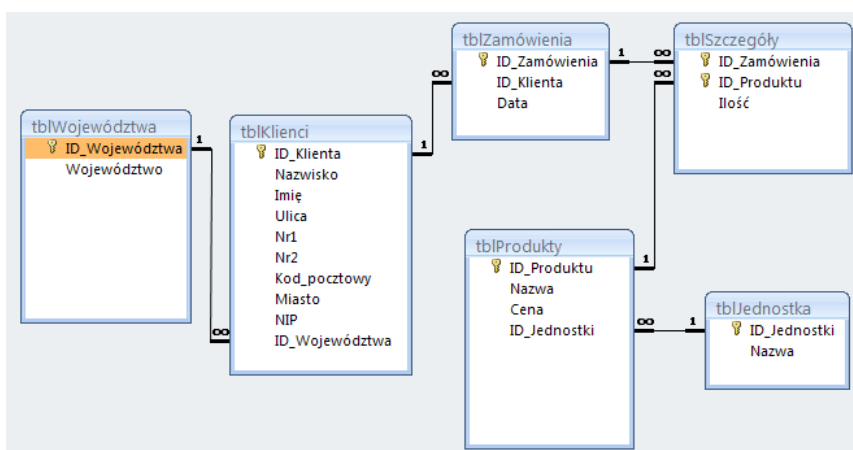
Makra programu Access stanowią uproszczony język programowania, który pozwala zwiększyć funkcjonalność bazy danych. Można na przykład dołączyć makro do przycisku polecenia w formularzu, dzięki czemu będzie ono uruchamiane przy każdym kliknięciu tego przycisku. Makra zawierają opis akcji służących do wykonywania zadań, takich jak otwieranie raportu, uruchamianie kwerendy lub zamykanie bazy danych. Makra pozwalają zaoszczędzić czas, gdyż za ich pomocą można zautomatyzować większość operacji bazy danych wykonywanych ręcznie.

Moduły, podobnie jak makra, są obiektami umożliwiającymi rozszerzenie funkcjonalności bazy danych. Makra są tworzone przez wybieranie akcji z listy, podczas gdy moduły są pisane w języku programowania. Moduł jest zbiorem deklaracji, instrukcji i procedur przechowywanych razem jako jednostka. Istnieją dwa typy modułów: moduły klas i moduły standardowe. Moduły klas są dołączane do formularzy lub raportów i najczęściej zawierają procedury dotyczące formularza lub raportu, do którego je dołączono. Moduły standardowe zawierają procedury ogólne, które nie są powiązane z żadnymi innymi obiektami. W przeciwieństwie do modułów klas moduły standardowe są wyświetlane w obszarze Moduły okienka nawigacji.

Model relacyjny

Model organizacji danych bazujący na matematycznej teorii mnogości, w szczególności na pojęciu relacji. Na modelu relacyjnym oparta jest relacyjna baza danych (RDBMS ang. Relational Database Management Systems) – baza danych, w której dane są przedstawione w postaci relacyjnej.

W najprostszym ujęciu w modelu relacyjnym dane grupowane są w relacje, które reprezentowane są przez tablice. Relacje są pewnym zbiorem rekordów o identycznej strukturze wewnętrznie powiązanych za pomocą związków zachodzących pomiędzy danymi. Relacje zgrupowane są w tzw. schematy bazy danych. Relacją może być tabela zawierająca dane teleadresowe pracowników, zaś schemat może zawierać wszystkie dane dotyczące firmy. Takie podejście w porównaniu do innych modeli danych ułatwia wprowadzanie zmian, zmniejszenie możliwości pomyłek, ale dzieje się to kosztem wydajności.



Postulaty Codd'a

12 postulatów Codd'a – jest to zestaw 13 zasad numerowanych od 0 do 12 stworzonych przez Edgara F. Codd'a pioniera relacyjnych baz danych. Każda relacyjna baza danych musi je spełniać.

- 0) System musi być kwalifikowany jako relacyjny, jako baza danych i jako system zarządzania
- 1) Postulat informacyjny – dane są reprezentowane jedynie poprzez wartości atrybutów w wierszach tabel,
- 2) Postulat dostępu – każda wartość w bazie danych jest dostępna poprzez podanie nazwy tabeli, atrybutu oraz wartości klucza podstawowego,
- 3) Postulat dotyczący wartości NULL – dostępna jest specjalna wartość NULL dla reprezentacji wartości nieokreślonej jak i nieadekwatnej, inna od wszystkich i podlegająca przetwarzaniu
- 4) Postulat dotyczący katalogu – wymaga się, aby system obsługiwał wbudowany katalog relacyjny z bieżącym dostępem dla uprawnionych użytkowników używających języka zapytań,
- 5) Postulat języka danych – system musi dostarczać pełnego języka przetwarzania danych, który może być używany w trybie interaktywnym jak i w obrębie programów aplikacyjnych, obsługuje operacje definiowania danych, operacje manipulowania danymi, ograniczenia związane z bezpieczeństwem i integralnością oraz operacje zarządzania transakcjami,
- 6) Postulat modyfikowalności perspektyw – system musi umożliwiać modyfikowanie perspektyw, o ile jest ono (modyfikowanie) semantycznie realizowalne,
- 7) Postulat modyfikowalności danych – system musi umożliwiać operacje modyfikacji danych, musi obsługiwać operatory INSERT, UPDATE oraz DELETE,
- 8) Postulat fizycznej niezależności danych – zmiany fizycznej reprezentacji danych i organizacji dostępu nie wpływają na aplikacje,
- 9) Postulat logicznej niezależności danych – zmiany wartości w tabelach nie wpływają na aplikacje,
- 10) Postulat niezależności więzów spójności – więzy spójności są definiowane w bazie i nie zależą od aplikacji,
- 11) Postulat niezależności dystrybucyjnej – działanie aplikacji nie zależy od modyfikacji i dystrybucji bazy,
- 12) Postulat bezpieczeństwa względem operacji niskiego poziomu – operacje niskiego poziomu nie mogą naruszać modelu relacyjnego i więzów spójności.

Etapy projektowania baz danych

- Określenie celu, któremu ma służyć baza danych

Pierwszym krokiem w projektowaniu bazy danych jest określenie celu, któremu ma służyć baza i sposobu jej używania. Trzeba wiedzieć, jakich informacji ma dostarczyć baza danych. Na tej podstawie można określić, jakie zagadnienia będą analizowane w bazie danych (tabele) i jakie informacje mają określać każde zagadnienie (pola w tabelach). Należy porozmawiać z przyszłymi użytkownikami bazy danych. Zorganizować burzę mózgów dotyczącą pytań, na które baza danych miałaby odpowiadać. Naszkicować wzory raportów, które powinna wytwarzać. Zgromadzić formularze do wpisywania danych, które są używane obecnie. Zapoznać się z działaniem dobrze zaprojektowanych baz danych, podobnych do tej, która ma być utworzona.

- Określenie tabel, które są potrzebne w bazie danych

Określenie tabel może być najtrudniejszym etapem procesu projektowania bazy danych. Powodem tego jest fakt, że wyniki, których ma dostarczyć baza danych – raporty gotowe do wydrukowania, formularze przeznaczone do wykorzystania, pytania, na które trzeba odpowiedzieć – nie zawsze zawierają bezpośrednie wskazówki o strukturze tworzących je tabel. Tabele nie

muszą być projektowane za pomocą programu Microsoft Access. W istocie, lepiej jest najpierw wykonać szkic i przemyśleć projekt korzystając z kartki papieru. Projektując tabele należy podzielić posiadane informacje pamiętając o przedstawionych poniżej podstawowych zasadach projektowania:

- o Tabela nie powinna zawierać powtarzających się informacji, a informacje nie powinny powtarzać się w różnych tabelach. Pod tym względem tabela w relacyjnej bazie danych różni się od tabeli w aplikacji "płaskiego pliku", takiej jak arkusz kalkulacyjny. Jeżeli informacje są przechowywane tylko w jednej tabeli, można je zaktualizować zmieniając dane tylko w jednym miejscu. Tak jest łatwiej, a ponadto unika się wystąpienia niezgodności danych zapisanych w różnych miejscach. Na przykład, dobrze jest zapisać numer telefonu i adres klienta tylko raz, w jednej tabeli.
- o Każda tabela powinna zawierać informacje tylko na jeden temat. Jeżeli tabela zawiera dane na temat jednego zagadnienia można przetwarzać je niezależnie od danych dotyczących innych zagadnień. Na przykład, adresy klientów będą przechowywane w innej tabeli niż zamówienia. W ten sposób można usunąć jakieś zamówienie, zachowując dane o kliencie.

– Określenie pól, które są potrzebne w tabelach

Każda tabela zawiera informacje na temat jednego zagadnienia, a każde pole w tabeli zawiera jedną daną dotyczącą tego zagadnienia, któremu poświęcona jest tabela. Na przykład, tabela poświęcona klientom może zawierać pola z nazwą firmy, adresem, miastem, krajem i numerem telefonu. Zabierając się do projektowania pól w tabelach dobrze jest pamiętać o następujących wskazówkach:

- o Należy powiązać każde pole bezpośrednio z zagadnieniem, którego dotyczy tabela.
- o Nie należy wprowadzać danych pośrednich lub obliczonych (danych, które są wynikiem wyrażenia).
- o Należy uwzględnić wszystkie potrzebne informacje.
- o Informacje należy przechowywać w możliwie najmniejszych jednostkach logicznych (na przykład Imię oraz Nazwisko, a nie Dane personalne).

– Przypisanie polom jednoznacznych wartości w każdym rekordzie

Aby program Microsoft Access mógł powiązać informacje przechowywane w różnych tabelach – na przykład, aby powiązać klienta ze wszystkimi jego zamówieniami – każda tabela w bazie danych musi zawierać pole lub zbiór pól, które jednoznacznie określają każdy rekord. Takie pole lub zbiór pól nazywany jest kluczem podstawowym.

– Określenie relacji między tabelami

Po podzieleniu danych na tabele i zdefiniowaniu pól kluczy podstawowych trzeba wprowadzić do programu Microsoft Access informacje na temat sposobu poprawnego łączenia powiązanych danych w logiczną całość. W tym celu definiuje się relacje między tabelami w bazie danych Microsoft Access. Być może warto sprawdzić, jak utworzono te relacje w dobrze zaprojektowanej bazie danych.

– Udoskonalenie projektu

Po zaprojektowaniu tabel, pól i łączących ich relacji przychodzi czas na sprawdzenie projektu i wykrycie ewentualnych usterek. Łatwiej poprawić bazę danych teraz niż po wypełnieniu tabel danymi. Wykorzystaj program Microsoft Access do utworzenia tabel, określ łączące je relacje i

wprowadź do tabel dostatecznie dużo przykładowych danych, aby można było sprawdzić projekt. Aby przetestować relacje w bazie danych, sprawdź, czy można utworzyć kwerendy w celu uzyskania odpowiedzi na stawiane pytania. Utwórz w zarysie formularze i raporty i sprawdź, czy przedstawiają te dane, których oczekujesz. Poszukaj niepotrzebnych powtórzeń danych i usuń je. Jeśli napotkasz problemy, popraw projekt.

– Wprowadzenie danych i utworzenie innych obiektów bazy danych

Kiedy okaże się, że struktura tabel spełnia opisane powyżej zasady, można wprowadzić wszystkie dane do tabel i utworzyć dowolne kwerendy, formularze, raporty, strony dostępu do danych, makra i moduły.

– Zastosowanie narzędzi analizy programu Microsoft Access

Program Microsoft Access jest wyposażony w dwa narzędzia, które pomagają udoskonalić projekt bazy danych Microsoft Access. Kreator analizy tabel potrafi sprawdzić projekt przeglądając jednorazowo jedną tabelę, zaproponować nową strukturę tabeli i – jeśli to jest potrzebne – podzielić tabelę na kilka tabel połączonych relacjami. Analizator wydajności potrafi przeprowadzić analizę całej bazy danych, sformułować zalecenia i propozycje dotyczące wprowadzenia ulepszeń. Analizator wydajności może też wprowadzić te zalecenia i propozycje.

Modelowanie E-R

Model związków encji ERM – Entity Relationship Model. Najpopularniejszy konceptualny model danych.

ERD – Entity Relationship Diagram to graficzny odpowiednik ERM. W diagramach związków encji obszar analizy reprezentowany jest przez:

- Encje
- Związki
- Atrybuty

- Encja:
 - Oddzielna jednostka zawartości BD
 - Obiekt materialny lub niematerialny, fakt, zdarzenie itp., o którym informacje mają być przechowywane w bazie
 - Jest jednoznacznie identyfikowana za pomocą nazwy
 - Encję w modelu danych stworzonym w oparciu o diagramy związków encji należy uczynić te rzeczy, dla których trzeba przechowywać dane na temat wielu ich właściwości

- Związek
 - Istotne powiązanie pomiędzy dwiema encjami
 - Przedstawia zależności zachodzące pomiędzy obiektami
 - Rozpatrywane są jedynie zależności bezpośrednie

- Pomiędzy dwoma encjami może zachodzić więcej niż jeden związek
 - Zwykle określany jest czasownikiem
 - Związek oznaczany jest na diagramie jako linia łącząca encje
 - Przykłady związków encji: jeden do jednego, jeden do wielu, wiele do wielu.
- **Atrybut**
 - Zbiór cech, właściwości
 - Charakteryzuje dany typ encji
 - Zestaw atrybutów, który jednoznacznie opisuje encję, nazywa się wiązką atrybutów
 - Ma swoją wartość
 - Przykłady wad baz danych
 - model danych (szczególnie relacyjny) jest zbyt prosty do zamodelowania złożonych, zagnieżdżonych encji (np. w przypadku występowania zależności wielowartościowych)
 - systemy baz danych nie obsługują ogólnych typów danych występujących w językach programowania
 - model danych nie zawiera kilku często używanych pojęć semantycznych (np.: generalizacja, agregacja)
 - zbyt wolne działanie systemów baz danych z programami użytkowymi wymagającymi szybkich i skomplikowanych obliczeń
 - systemy baz danych nie dostarczają narzędzi do reprezentowania i zarządzania temporalnymi aspektami baz danych (m.in.: pojęciem czasu, wersjami obiektów i schematu)



Skierniewicka 10A
01-230 Warszawa
tel.: +48 22 465 88 88
biuro@expose.pl
www.expose.pl