



Access Zaawansowany

OPIEKA POSZKOLENIOWA

GWARANTOWANA POD ADRESEM:

opieka@cognity.pl



Proces szkoleniowy

Analiza Potrzeb Szkoleniowych

Wypełnione **testy i ankiety** sprawdzające poziom wiedzy i umiejętności oraz potrzeby w zakresie opanowania nowych kompetencji pomogą w określeniu Twoich oczekiwań względem szkolenia.

Praktyczne Szkolenie

Przeprowadzimy dla Ciebie szkolenie, na którym będziesz aktywnie rozwiązywać ćwiczenia oparte o **rzeczywiste sytuacje** spotykane w codziennej pracy zawodowej, realizowane przez doświadczonych **trenerów praktyków**.



Opieka poszkoleniowa

Po szkoleniu możesz skorzystać z **opieki poszkoleniowej**. Masz do wyboru jedną z dwóch możliwych **form kontaktu z trenerem**: konsultację telefoniczną lub e-mailową bądź możliwość spotkania w ramach konsultacji po szkoleniu.

Certyfikaty i dostęp do strefy wiedzy

Na zakończenie zajęć otrzymasz **certyfikat** czyli dokument potwierdzający zdobyte umiejętności na szkoleniu. Jeśli masz życzenie możesz uzyskać certyfikat także **w języku angielskim**. Uzyskasz **dostęp do Strefy Wiedzy Cognito**, gdzie będziesz mógł korzystać z ćwiczeń i materiałów szkoleniowych **w wersji elektronicznej**, które pomogą utrwalić Ci wiedzę zdobytą w trakcie kursu.

Życzymy udanego szkolenia i zapraszamy do kontaktu!

COGNITY Sp. z o.o. - ☎ 12 421 87 54, ✉ biuro@cognito.pl, www.cognito.pl

Praktycznie najlepsze szkolenia



Pełna oferta na:
www.cognity.pl

SPIS TREŚCI

Zasady projektowania baz danych	6
Zasady projektowania baz danych	6
Etap 1. Ogólny projekt systemu - od koncepcji do rzeczywistości.....	6
Etap 2. Projektowanie raportów.....	7
Etap 3. Projektowanie danych: jakie pola są obowiązkowe?.....	8
Etap 4. Projektowanie tabel.....	8
Etap 5. Projektowanie formularzy - wejście	9
Relacje i klucze	10
Jeden do jednego.....	10
Jeden do wielu	11
Wiele do wielu	12
Więzy integralności	14
Normalizacja danych	17
Pierwsza postać normalna	17
Druga postać normalna	19
Trzecia postać normalna	21
Tabele	22
Tabele	22
Istota konwencji nazewnictwa	22
Typy danych dostępnych w Access.....	23
Projektowanie reguł wprowadzania danych	23
Projektowanie tabel odnośników	24
Typy danych pól	24
Dane tekstowe.....	26
Typ danych Nota	26
Typ danych Liczba	27
Typ danych Data/Godzina	27
Waluta	28
Autonumerowanie	28
Maska wprowadzania.....	29
Reguła sprawdzania poprawności.....	31
Pojęcie kluczy.....	31
Określanie klucza głównego.....	32
Kwerendy	33
Typy kwerend	33

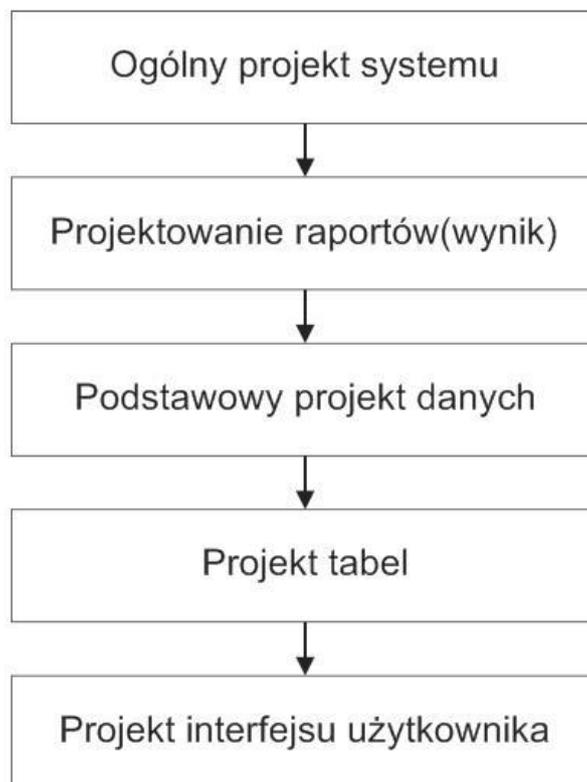
Możliwości kwerend	34
Raporty	35
Raporty tabelaryczne	35
Raporty kolumnowe.....	36
Etykiety pocztowe.....	38
Różnice między raportami a formularzami	39
Procedura tworzenia raportu	39
Definiowanie układu raportu	40
Gromadzenie danych	40

ZASADY PROJEKTOWANIA BAZ DANYCH

ZASADY PROJEKTOWANIA BAZ DANYCH

Aby utworzyć obiekty bazy danych, takie jak tabele, formularze czy raporty, najpierw należy wykonać szereg czynności określanych wspólnym mianem projektu. Im lepszy projekt, tym lepsza aplikacja, która na nim bazuje. Im więcej czasu programista poświęci na etapie projektowania, tym szybciej uda mu się stworzyć cały system. Jedynekim celem zaprojektowania obiektu jest stworzenie jednoznacznej charakterystyki, na podstawie której można go później zaimplementować.

Na poniższym rysunku 1 zaprezentowano wersję metody projektowania specjalnie przystosowaną do wykorzystania w Accessie. Zastosowano w niej metodę góra-dół, która zaczyna się od ogólnego projektu systemu, a kończy na projekcie formularzy. Metoda ta składa się z pięciu etapów.



Rys. 1. Diagram pięciostopniowej metody projektowania bazy danych w Access.

ETAP 1. OGÓLNY PROJEKT SYSTEMU - OD KONCEPCJI DO RZECZYWISTOŚCI

Pięciostopniowa metoda projektowania przedstawiona na rysunku 1 umożliwi stworzenie żądanego systemu za przystępną cenę. Na przykład może to być baza danych Komis

samochodowy która będzie systemem umożliwiającym sprzedawanie towarów (pojazdów i części) klientom. Baza danych Komis samochodowy powinna automatyzować następujące zadania:

- Wprowadzanie i zarządzanie informacjami o klientach i sprzedawcach (nazwisko, adres, historia finansowa).
- Wprowadzanie i zarządzanie informacjami o dokonanych transakcjach (data sprzedaży, metoda płatności, kwota brutto, identyfikator nabywcy oraz inne pola).
- Wprowadzanie i zarządzanie informacjami dotyczącymi poszczególnych pozycji transakcji (szczegóły sprzedawanych produktów).
- Przeglądanie informacji pochodzących z wszystkich tabel (transakcji, klientów, poszczególnych pozycji transakcji oraz informacji o płatnościach).
- Zadawanie wszelkiego rodzaju pytań na temat informacji zawartych w bazie danych.
- Tworzenie aktualnego katalogu klientów.
- Tworzenie miesięcznego raportu sprzedaży.
- Tworzenie historii sprzedaży dla poszczególnych klientów.
- Tworzenie etykiet adresowych oraz raportów korespondencyjnych.

Powyższe operacje które musi umożliwiać aplikacja Komis samochodowy, wynikają z analizy potrzeb klienta. W trakcie projektowania może również pojawić się potrzeba uwzględnienia innych zadań. Większość informacji potrzebnych do stworzenia systemu pochodzi od jego docelowych użytkowników. Oznacza to, że trzeba z nimi usiąść i dowiedzieć się, jak wygląda i działa istniejący dotąd system fizyczny. W tym celu należy przeprowadzić analizę wymagań tego systemu oraz określić, w jaki sposób można go zautomatyzować.

Po zakończeniu przygotowań do następnych etapów, nie należy zrywać kontaktu z klientem – należy go na bieżąco informować o wykonanych czynnościach, pytać o kolejne szczegóły, sprawdzać, czy wykonywane działania spełniają jego wymagania.

ETAP 2. PROJEKTOWANIE RAPORTÓW

Chociaż rozpoczynanie od raportów może się wydawać dziwne, w wielu przypadkach użytkownicy są bardziej zainteresowani wydrukami z bazy danych niż wszelkimi innymi elementami aplikacji. Raporty bardzo często zawierają niemal wszystkie dane zarządzane przez aplikację. Ponieważ raporty zwykle są obszerne, często są najlepszym sposobem zbierania informacji dotyczących wymagań dla bazy danych. W przypadku bazy danych Komis samochodowy drukowane raporty będą zawierać szczegółowe informacje niemal o wszystkich danych w bazie wraz z odpowiednimi podsumowaniami.

Po zdefiniowaniu ogólnego projektu systemu Komis samochodowy wraz z funkcjami, które ma spełniać, można przystąpić do projektowania raportów.

Podczas analizy raportów tworzonych w tym punkcie nasuwa się pytanie czy najpierw należy opracować układ raportu, czy też wcześniej należy zdecydować o elementach danych

i etykietach tekstowych, które go tworzą? W rzeczywistości te dwa elementy są analizowane w tym samym czasie.

Sposób rozmieszczenia pól w raporcie nie jest zbyt ważny. Jednak im więcej czasu poświęcimy teraz, tym łatwiej będzie nam skonstruować raport. Projekt i rozmieszczenie pól w każdym z raportów zależy od przeznaczenia raportu i zawartych w nim danych.

ETAP 3. PROJEKTOWANIE DANYCH: JAKIE POLA SĄ OBOWIĄZKOWE?

Następnym etapem procesu projektowania jest zgromadzenie informacji na temat wszystkich danych, które są konieczne do utworzenia żądanych raportów. Jedną z najlepszych metod jest wypisanie wszystkich elementów znajdujących się na poszczególnych raportach. W przypadku podjęcia decyzji o takim sposobie, należy zwracać uwagę na elementy znajdujące się na więcej niż jednym raporcie. Należy pamiętać, aby nadać im te same nazwy - w rzeczywistości są to te same elementy.

Inną metodą jest rozdzielenie elementów na pewne logiczne układy. Później elementy te będą zgrupowane w logiczne struktury tabel, na podstawie których zostaną utworzone właściwe ekrany do wprowadzania danych (formularze). Na przykład dane dotyczące klientów (sprzedawców czy nabywców) powinny być wprowadzane podczas rejestrowania klientów, a nie podczas wpisywania danych dotyczących transakcji.

ETAP 4. PROJEKTOWANIE TABEL

Teraz najtrudniejsza część: trzeba określić pola w tabelach, na podstawie których będą generowane raporty. Po przeanalizowaniu wielu pól i obliczeń potrzebnych do utworzenia żądanych dokumentów, można wywnioskować, że poszczególne pola raportów powinny pochodzić z różnych tabel w bazie danych.

Należy pamiętać o konieczności uwzględnienia w tabelach bazy danych wszystkich, nawet najmniejszych szczegółów. Na przykład może się zdarzyć, że użytkownik zażąda umieszczenia w bazie danych informacji o urlopach i innych dniach wolnych w celu łatwego wyszukiwania informacji o tym, którzy pracownicy są dostępni określonego dnia. Łatwo może jednak dojść do zaciemnienia ogólnego projektu aplikacji z powodu uwzględnienia zbyt wielu wątków podczas wstępnej fazy projektowania. Ponieważ tabele Accessa można z łatwością modyfikować w dowolnym momencie, najlepszą strategią jest odłożenie mniej ważnych elementów na późniejsze etapy projektowania - do czasu, kiedy wstępny projekt będzie gotowy. Ogólnie rzecz biorąc, uwzględnienie żądań użytkowników po zakończeniu fazy projektowania bazy danych nie jest szczególnie trudne.

Po wyodrębnieniu danych ze wszystkich raportów, należy je scalić według funkcji, które pełnią (czyli, na przykład, podzielić na logiczne grupy), a następnie porównać dane w ramach poszczególnych funkcji. W tym celu najpierw należy sięgnąć do informacji związanych z klientami i połączyć wszystkie występujące tam pola w jeden zbiór elementów danych. Następnie to samo

należy wykonać dla informacji dotyczących transakcji oraz informacji dotyczących pojedynczej pozycji transakcji.

Relacje pomiędzy tabelami mogą być różne. Na przykład każdej fakturze sprzedaży odpowiada jeden i tylko jeden klient, natomiast z jednym klientem może być związanych wiele transakcji. Podobna relacja zachodzi pomiędzy fakturą a zawartymi na niej pozycjami.

Zdefiniowanie relacji w bazie danych wymaga istnienia niepowtarzalnego identyfikatora w obu tabelach. Bez unikatowego identyfikatora mechanizm bazy danych nie potrafiłby prawidłowo połączyć i wyodrębnić powiązanych danych. Pole, które w unikatowy sposób identyfikuje każdy wiersz w tabeli, to tzw. klucz główny. Odpowiadające mu pole w powiązanej tabeli nazywa się kluczem obcym. Przykładowo pole Identyfikator klienta w tabeli Klienci może być kluczem głównym, natomiast pole Identyfikator klienta w tabeli Faktury to klucz obcy.

ETAP 5. PROJEKTOWANIE FORMULARZY - WEJŚCIE

Po utworzeniu danych oraz określeniu relacji pomiędzy tabelami, przyszedł czas na zaprojektowanie formularzy. Formularze składają się z pól, poprzez które można wprowadzać dane lub które je wyświetlają w trybie edycji. Jeżeli to możliwe, formularze wyświetlane na ekranie powinny przypominać formularze funkcjonujące w systemie fizycznym. Dzięki temu system będzie bardziej przyjazny dla użytkowników.

Podczas projektowania formularzy na ekranie, należy rozmieścić na nim trzy rodzaje pól:

- Etykiety oraz pola tekstowe służące do wprowadzania danych (pola w formularzach i raportach Accessa zwykle nazywa się formantami).
- Formanty specjalne (wielowierszowe pola tekstowe, przyciski opcji, pola list, pola wyboru, wykresy oraz obrazy).
- Obiekty graficzne poprawiające wygląd ekranu („kolory”, linie, prostokąty oraz efekty trójwymiarowe).

W czasie projektowania formularza można umieszczać pola (tekstowe, pola wyboru, pola list oraz przyciski opcji) w dowolnych miejscach. Idealna sytuacja występuje w przypadku, gdy formularz jest tworzony na podstawie istniejącego, wydrukowanego dokumentu - formularz Accessa powinien przypominać jego wersję papierową. Jego pola powinny wtedy znaleźć się mniej więcej w tych samych miejscach, co na wersji papierowej.

Etykiety służą do wyświetlania komunikatów, tytułów i podpisów. Pola tekstowe udostępniają miejsce, w którym można wpisywać (lub w którym są wyświetlane) tekst lub liczby zawarte w bazie danych. Pola wyboru wskazują na jakiś warunek i mogą być zaznaczone lub niezaznaczone. Inne typy formantów dostępnych w Accessie to, między innymi, pola list, pola kombi, przyciski opcji, przyciski przełączników oraz grupy opcji.

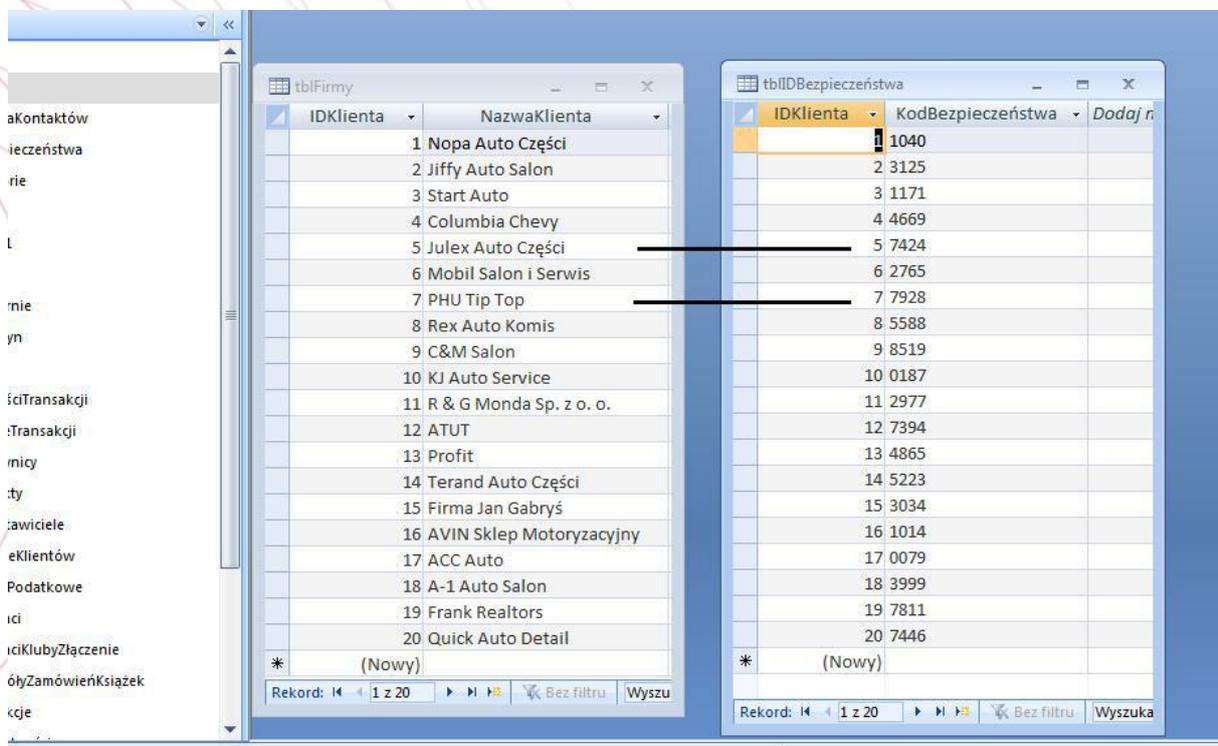
RELACJE I KLUCZE

W literaturze przedmiotu najczęściej są omawiane trzy podstawowe typy relacji pomiędzy tabelami:

- jeden do jednego;
- jeden do wielu;
- wiele do wielu.

JEDEN DO JEDNEGO

Relacja "jeden do jednego" pomiędzy tabelami oznacza, że każdemu rekordowi w pierwszej tabeli odpowiada jeden i tylko jeden rekord w drugiej tabeli. Pojęcie to zilustrowano na rysunku 2.



IDKlienta	NazwaKlienta
1	Nopa Auto Części
2	Jiffy Auto Salon
3	Start Auto
4	Columbia Chevy
5	Julex Auto Części
6	Mobil Salon i Serwis
7	PHU Tip Top
8	Rex Auto Komis
9	C&M Salon
10	KJ Auto Service
11	R & G Monda Sp. z o. o.
12	ATUT
13	Profit
14	Terand Auto Części
15	Firma Jan Gabryś
16	AVIN Sklep Motoryzacyjny
17	ACC Auto
18	A-1 Auto Salon
19	Frank Realtors
20	Quick Auto Detail

IDKlienta	KodBezpieczeństwa	Dodaj r.
1	1040	
2	3125	
3	1171	
4	4669	
5	7424	
6	2765	
7	7928	
8	5588	
9	8519	
10	0187	
11	2977	
12	7394	
13	4865	
14	5223	
15	3034	
16	1014	
17	0079	
18	3999	
19	7811	
20	7446	

Rys. 2. Relacja "jeden do jednego"

Klasycznych relacji „jeden do jednego” nie stosuje się zbyt często w relacyjnych bazach danych. W większości przypadków dane zawarte w drugiej z tabel są umieszczone w pierwszej tabeli. Relacji „Jeden do jednego” raczej się unika, ponieważ naruszają one reguły normalizacji. Zgodnie z regułami normalizacji, jeśli dane opisują pojedynczy podmiot, nie należy ich rozdzielać na wiele tabel. Przykładowo jeśli osoba ma jedną i tylko jedną datę urodzenia, powinna ona się znaleźć w tabeli zawierającej inne dane tej osoby.

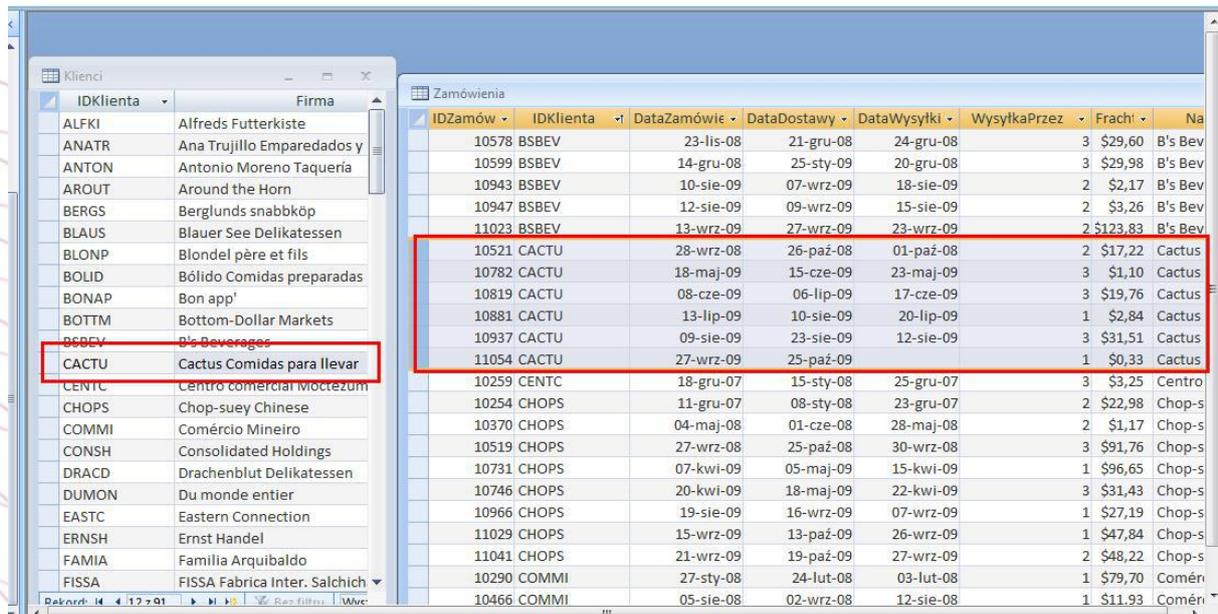
Czasami jednak zapisywanie określonych danych razem z innymi danymi w tabeli nie jest zalecane. Na przykład rozważmy sytuację pokazaną na rysunku 6. Dane zapisane w tabeli tblBezpieczeństwo są poufne. Nie chcielibyśmy, aby wszystkie osoby mające dostęp do publicznych danych dotyczących klienta (nazwisko, adres itp.) miały także dostęp do poufnego kodu bezpieczeństwa używanego podczas zakupów bądź do wystawiania faktur. Gdyby zachodziła konieczność, tabelę tblBezpieczeństwo można by umieścić na innym dysku w sieci lub nawet w celu ochrony przed niepowołanym dostępem przechowywać na przenośnym nośniku informacji.

Innym przypadkiem, kiedy trzeba zastosować relację „Jeden do jednego”, jest sytuacja, gdy liczba pól w tabeli przekroczy dozwoloną liczbę 255. Choć takie sytuacje występują rzadko, czasem może się zdarzyć, że w pojedynczej tabeli znajduje się zbyt wiele pól. Najprostszym rozwiązaniem jest rozdzielenie danych na wiele tabel i powiązanie ich za pomocą klucza głównego (wykorzystując w każdej z tabel tę samą wartość klucza).

JEDEN DO WIELU

Znacznie częstszym typem relacji występujących pomiędzy tabelami w relacyjnych bazach danych jest relacja „Jeden do wielu”. W takiej relacji każdy rekord z pierwszej tabeli (nadrzędnej) jest powiązany z jednym lub większą liczbą rekordów w drugiej tabeli (podrzędnej). Każdy rekord z drugiej tabeli jest powiązany z jednym i tylko jednym rekordem z pierwszej tabeli. Bez wątpienia relacje „Jeden do wielu” są najczęstszym typem relacji spotykanych w systemach relacyjnych baz danych. Przykładami relacji „Jeden do wielu” mogą być:

- Klienci i zamówienia: każdy klient (strona „Jeden”) może złożyć kilka zamówień (strona „wiele”), ale zamówienia te są przesyłane przez tego samego klienta.
- Nauczyciel i uczniowie: każdy nauczyciel ma wielu uczniów, ale każdy uczeń ma jednego nauczyciela (oczywiście w ramach określonej klasy).
- Pracownicy i чеки wypłat: każdy pracownik może otrzymać wiele czeków wypłat, ale określony czek można wręczyć jednemu i tylko jednemu pracownikowi.
- Pacjenci i kuracje: każdy pacjent jest poddawany zerowej bądź większej od zera liczbie kuracji w związku z chorobą.

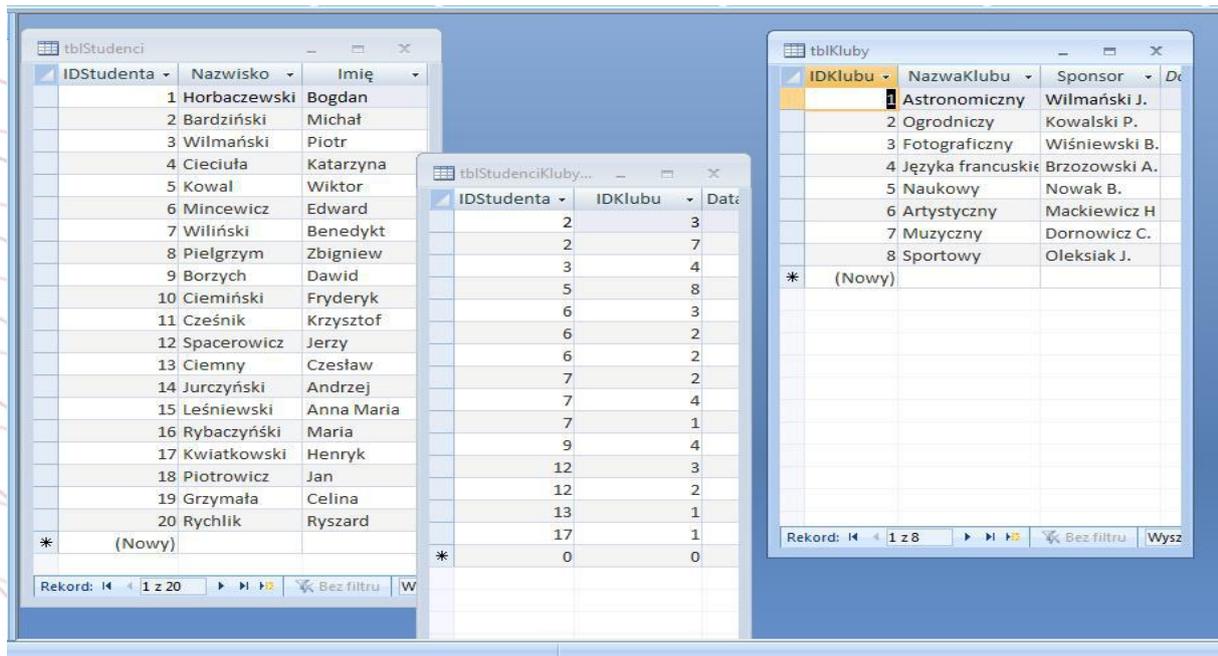


IDKlienta	Firma	IDZamów	IDKlienta	DataZamówię	DataDostawy	DataWysyłki	WysyłkaPrzez	Frach	Na
ALFKI	Alfreds Futterkiste	10578	BSBEV	23-lis-08	21-gru-08	24-gru-08	3	\$29,60	B's Bev
ANATR	Ana Trujillo Emparedados y	10599	BSBEV	14-gru-08	25-sty-09	20-gru-08	3	\$29,98	B's Bev
ANTON	Antonio Moreno Taquería	10943	BSBEV	10-sie-09	07-wrz-09	18-sie-09	2	\$2,17	B's Bev
AROUT	Around the Horn	10947	BSBEV	12-sie-09	09-wrz-09	15-sie-09	2	\$3,26	B's Bev
BERGS	Berglunds snabbköp	11023	BSBEV	13-wrz-09	27-wrz-09	23-wrz-09	2	\$123,83	B's Bev
BLAUS	Blauer See Delikatessen	10521	CACTU	28-wrz-08	26-paź-08	01-paź-08	2	\$17,22	Cactus
BLONP	Blondel père et fils	10782	CACTU	18-maj-09	15-cze-09	23-maj-09	3	\$1,10	Cactus
BOLID	Bólido Comidas preparadas	10819	CACTU	08-cze-09	06-lip-09	17-cze-09	3	\$19,76	Cactus
BONAP	Bon app'	10881	CACTU	13-lip-09	10-sie-09	20-lip-09	1	\$2,84	Cactus
BOTTM	Bottom-Dollar Markets	10937	CACTU	09-sie-09	23-sie-09	12-sie-09	3	\$31,51	Cactus
BSBEV	B's Beverages	11054	CACTU	27-wrz-09	25-paź-09		1	\$0,33	Cactus
CACTU	Cactus Comidas para llevar	10259	CENTC	18-gru-07	15-sty-08	25-gru-07	3	\$3,25	Centro
CENTC	Centro comercial Moccasin	10254	CHOPS	11-gru-07	08-sty-08	23-gru-07	2	\$22,98	Chop-s
CHOPS	Chop-suey Chinese	10370	CHOPS	04-maj-08	01-cze-08	28-maj-08	2	\$1,17	Chop-s
COMMI	Comércio Mineiro	10519	CHOPS	27-wrz-08	25-paź-08	30-wrz-08	3	\$91,76	Chop-s
CONSH	Consolidated Holdings	10731	CHOPS	07-kwi-09	05-maj-09	15-kwi-09	1	\$96,65	Chop-s
DRACD	Drachenblut Delikatessen	10746	CHOPS	20-kwi-09	18-maj-09	22-kwi-09	3	\$31,43	Chop-s
DUMON	Du monde entier	10966	CHOPS	19-sie-09	16-wrz-09	07-wrz-09	1	\$27,19	Chop-s
EASTC	Eastern Connection	11029	CHOPS	15-wrz-09	13-paź-09	26-wrz-09	1	\$47,84	Chop-s
ERNSH	Ernst Handel	11041	CHOPS	21-wrz-09	19-paź-09	27-wrz-09	2	\$48,22	Chop-s
FAMIA	Familia Arquibaldo	10290	COMMI	27-sty-08	24-lut-08	03-lut-08	1	\$79,70	Comér
FISSA	FISSA Fabrica Inter. Salchich	10466	COMMI	05-sie-08	02-wrz-08	12-sie-08	1	\$11,93	Comér

Rys. 3. Relacja "jeden do wielu"

WIELE DO WIELU

Od czasu do czasu w relacyjnych systemach baz danych wykorzystuje się relacje „wiele do wielu”. W układzie „wiele do wielu” każdy rekord w obydwu tabelach może być powiązany z jednym lub większą liczbą rekordów. Mogą też wystąpić rekordy, dla których w drugiej tabeli nie ma żadnego odpowiednika. Przykład takiej relacji pokazano na rysunku 4. Każdy student w tabeli tblStudentci może należeć do więcej niż jednego klubu, a każdy klub z tabeli tblKluby może mieć więcej niż jednego członka.



IDStudenta	Nazwisko	Imię
1	Horbaczewski	Bogdan
2	Bardziński	Michał
3	Wilmański	Piotr
4	Cieciula	Katarzyna
5	Kowal	Wiktor
6	Mincewicz	Edward
7	Wiliński	Benedykt
8	Pielgrzym	Zbigniew
9	Borzych	Dawid
10	Ciemniński	Fryderyk
11	Cześnik	Krzysztof
12	Spacerowicz	Jerzy
13	Ciemny	Czesław
14	Jurczyński	Andrzej
15	Leśniewski	Anna Maria
16	Rybaczyński	Maria
17	Kwiatkowski	Henryk
18	Piotrowicz	Jan
19	Grzymała	Celina
20	Rychlik	Ryszard
*	(Nowy)	

IDKluby	NazwaKluby	Sponsor
1	Astronomiczny	Wilmański J.
2	Ogrodniczy	Kowalski P.
3	Fotograficzny	Wiśniewski B.
4	Języka francuskie	Brzozowski A.
5	Naukowy	Nowak B.
6	Artystyczny	Mackiewicz H.
7	Muzyczny	Dornowicz C.
8	Sportowy	Oleksiak J.
*	(Nowy)	

IDStudenta	IDKluby	Data
2	3	
2	7	
3	4	
5	8	
6	3	
6	2	
6	2	
7	2	
7	4	
7	1	
9	4	
12	3	
12	2	
13	1	
17	1	
0	0	

Rys. 4. Relacja „wiele do wielu”

Jak można wywnioskować na podstawie rysunku 4, relacje „wiele do wielu” są w pewnym sensie trudniejsze do zrozumienia, ponieważ nie można ich bezpośrednio zamodelować w systemach relacyjnych baz danych takich jak Access. Zamiast tego relację „wiele do wielu” rozбивają się na dwie osobne relacje „Jeden do wielu”, połączone ze sobą za pomocą tabeli łączącej. Tabela ta wchodzi w relację „jeden do wielu” z obydwojema tabelami wchodzącymi w skład relacji „wiele do wielu”.

Relacje „wiele do wielu” spotyka się dość powszechnie w środowiskach biznesowych:

- Prawnicy i klienci (lub lekarze i pacjenci): każdy prawnik może prowadzić kilka procesów, a jednocześnie każdego klienta w każdym procesie może reprezentować więcej niż jeden prawnik.
- Pacjenci i ochrona ubezpieczeniowa: wiele osób chroni więcej niż jedna polisa ubezpieczeniowa. Na przykład: jeśli pracodawcy zarówno męża, jak i żony zapewniają ubezpieczenie medyczne, każdy z małżonków jest chroniony dwoma polisami.
- Wypożyczalnie płyt DVD i ich klienci: w ciągu roku każda płyta DVD jest wypożyczana przez wiele osób. Jednocześnie w ciągu roku każdy z klientów wypożycza wiele płyt.
- Prenumeratory czasopism: większość czasopism wychodzi w nakładach liczonych w tysiące lub nawet miliony. Wiele osób prenumeruje więcej niż jedno czasopismo.

WIĘZY INTEGRALNOŚCI

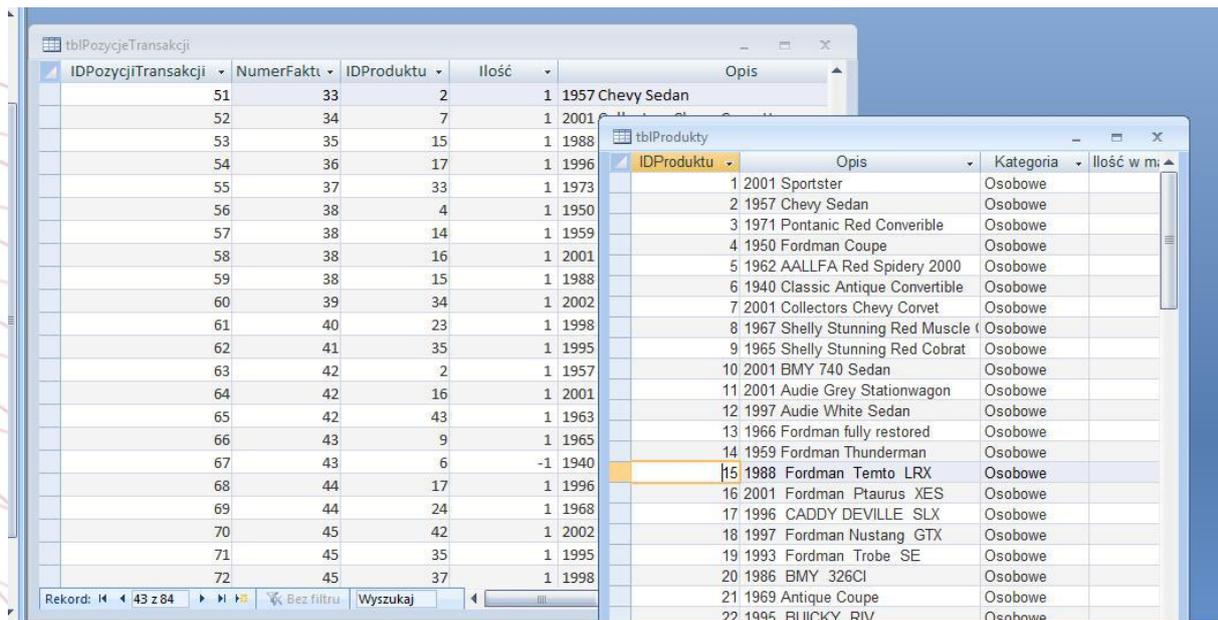
Access zezwala na stosowanie reguł więzów integralności, które chronią dane przed utratą lub zniszczeniem. W modelu relacyjnym zdefiniowano kilka reguł, które mają na celu wymuszenie wymagań więzów integralności relacyjnych baz danych. Dodatkowo Access posiada własny zbiór reguł więzów integralności, nad których przestrzeganiem czuwa aparat bazy danych Jet. Zachowanie więzów integralności oznacza zachowanie relacji pomiędzy tabelami w przypadku aktualizacji, usuwania i innych operacji na rekordach.

Wyobraźmy sobie aplikację płacową, w której nie obowiązują żadne reguły korzystania z danych w bazie danych. W takim systemie można by wystawić czeki niepowiązane z pracownikami. Mogliby również istnieć pracownicy, którym nigdy nie wystawiono czeków. Z punktu widzenia biznesowego wystawianie czeków dla pracowników-zjaw jest znacznie groźniejsze od występowania w bazie danych pracowników, którym nigdy nie zapłacono. Pracownik, któremu zapomniano zapłacić, szybko poinformuje o tej sytuacji, natomiast czeki wystawione pracownikom-zjawom nie spowodują żadnego alarmu. Sytuacja taka może trwać do momentu kontroli, która wykryje niedociągnięcia.

W systemie baz danych muszą obowiązywać reguły określające specyficzne warunki pomiędzy tabelami - reguły wymuszające integralność relacji pomiędzy nimi. Reguły te określa się mianem więzów integralności. Ich zadaniem jest utrzymanie relacji pomiędzy tabelami w systemie zarządzania relacyjnymi bazami danych. Więzy integralności blokują użytkownikom możliwość modyfikowania danych w sposób powodujący naruszenie relacji pomiędzy tabelami.

Reguły więzów integralności wykorzystują pola kluczowe tabel. Sprawdzają pola kluczowe za każdym razem, kiedy do tabeli jest dodawany klucz główny bądź obcy, jest on modyfikowany lub usuwany z tabeli. Jeśli modyfikacja wartości pola kluczowego stanie się przyczyną nieprawidłowej relacji, mówi się o naruszeniu więzów integralności. Tabele można zaprojektować w taki sposób, aby więzy integralności były wymuszane automatycznie.

Na rysunku 5 zilustrowano jedną z kilku relacji w bazie danych Komis samochodowy. Tabela tblProdukty jest powiązana z tabelą tblPozycjeTransakcji za pośrednictwem pola IDProduktu. Pole IDProduktu w tabeli tblProdukty jest kluczem głównym tej tabeli, natomiast pole IDProduktu w tabeli tblPozycjeTransakcji jest kluczem obcym. Relacja łączy każdy produkt z pozycją transakcji na fakturze. W tej relacji tabela tblProdukty jest nadrzędna, natomiast tabela tblPozycjeTransakcji jest podrzędna.



IDPozycjiTransakcji	NumerFaktu	IDProduktu	Ilość	Opis
51	33	2	1	1957 Chevy Sedan
52	34	7	1	2001
53	35	15	1	1988
54	36	17	1	1996
55	37	33	1	1973
56	38	4	1	1950
57	38	14	1	1959
58	38	16	1	2001
59	38	15	1	1988
60	39	34	1	2002
61	40	23	1	1998
62	41	35	1	1995
63	42	2	1	1957
64	42	16	1	2001
65	42	43	1	1963
66	43	9	1	1965
67	43	6	-1	1940
68	44	17	1	1996
69	44	24	1	1968
70	45	42	1	2002
71	45	35	1	1995
72	45	37	1	1998

IDProduktu	Opis	Kategoria	Ilość w m
1	2001 Sportster	Osobowe	
2	1957 Chevy Sedan	Osobowe	
3	1971 Pontiac Red Convertible	Osobowe	
4	1950 Fordman Coupe	Osobowe	
5	1962 AALLFA Red Spidery 2000	Osobowe	
6	1940 Classic Antique Convertible	Osobowe	
7	2001 Collectors Chevy Corvet	Osobowe	
8	1967 Shelly Stunning Red Muscle	Osobowe	
9	1965 Shelly Stunning Red Cobrat	Osobowe	
10	2001 BMY 740 Sedan	Osobowe	
11	2001 Audie Grey Stationwagon	Osobowe	
12	1997 Audie White Sedan	Osobowe	
13	1966 Fordman fully restored	Osobowe	
14	1959 Fordman Thunderman	Osobowe	
15	1988 Fordman Temto LRX	Osobowe	
16	2001 Fordman Ptaurus XES	Osobowe	
17	1996 CADDY DEVILLE SLX	Osobowe	
18	1997 Fordman Nustang GTX	Osobowe	
19	1993 Fordman Trobe SE	Osobowe	
20	1986 BMY 326Ci	Osobowe	
21	1969 Antique Coupe	Osobowe	
22	1945 FIICKY RIV	Osobowe	

Rys. 5. Typowa relacja w bazie danych

Osierocene rekordy w aplikacji bazy danych są bardzo niedobrym symptomem. Ponieważ informacje płacowe są zazwyczaj wyświetlane jako lista czeków płacowych wraz z nazwiskami pracowników, którym je wydano, czek płacowy niepowiązany z żadnym pracownikiem w większości przypadków nie będzie wykryty. Z łatwością można sprawdzić, które чеки wydano Janowi Kowalskiemu, ale na podstawie dowolnego чеку trudno stwierdzić brak istnienia uprawnionego pracownika, któremu ten czek wydano.

Ponieważ nad przestrzeganiem reguł więzów integralności czuwa aparat bazy danych Accessa, integralność danych jest zapewniona w momencie pojawienia się danych w bazie danych: w tabelach, kwerendach bądź formularzach. Po zdefiniowaniu wymagań integralności aplikacji nie trzeba się martwić tym, że dane w powiązanych tabelach zostaną utracone lub zdezorganizowane.

Dzięki ogólnym zasadom więzów integralności modelu relacyjnego rekordy zapisane w relacyjnych tabelach nie mogą być utracone lub pogubione. Z oczywistych powodów ważna jest ochrona kluczy głównych łączących tabele. Również zmiany wprowadzone w tabeli, które mają wpływ na inne tabele powinny być uwzględnione w powiązanej tabeli. W innym przypadku dane w obu tabelach szybko stracą synchronizację.

Zgodnie z pierwszą zasadą więzów integralności żaden klucz główny nie może zawierać wartości nuli. Wartość nuli to taka, która po prostu nie istnieje. Wartość pola, do którego nigdy nie przypisano wartości (nawet domyślnej), wynosi nuli. W tabeli bazy danych nie może być wierszy, które mają wartość null w polu klucza głównego, ponieważ podstawowym przeznaczeniem kluczy głównych jest zapewnienie niepowtarzalności wierszy. Oczywiście jest, że wartości null nie mogą być niepowtarzalne, a model relacyjny nie będzie działał, jeśli klucze główne będą miały wartość null.

Access automatycznie wymusza przestrzeganie pierwszej reguły więzów integralności. Podczas dodawania danych do tabel nie można pozostawić pustych pól kluczy głównych bez wygenerowania ostrzeżenia (jest to jeden z powodów, dla którego pola typu AutoNumerowanie tak dobrze sprawdzają się w roli kluczy głównych). Po wyznaczeniu w tabeli Accessa pola jako klucza głównego Access nie dopuści do usunięcia danych w tym polu ani do ich modyfikacji w taki sposób, że powstaną duplikaty wartości w innym rekordzie.

W przypadku używania złożonych kluczy głównych, składających się z kilku pól, wszystkie pola wchodzące w skład klucza głównego muszą zawierać wartości. Żadne z pól nie może być puste. Kombinacja wartości w złożonym kluczu głównym musi być niepowtarzalna.

Zgodnie z drugą zasadą więzów integralności wartościom kluczy obcych muszą odpowiadać klucze główne. Oznacza to, że każdemu rekordowi w tabeli po stronie „wiele” w relacji „Jeden do wielu” musi odpowiadać rekord w tabeli po stronie „Jeden” relacji. Rekord po stronie „wiele” relacji bez odpowiadającego mu rekordu po stronie „Jeden” to tzw. rekord osierocony, który powinien być usunięty z bazy danych. Identyfikacja osieroconych rekordów w bazie danych może być bardzo trudna, dlatego należy przede wszystkim nie dopuścić do ich pojawienia się w tabeli.

Spełnienie drugiej reguły oznacza, że aparat Jet bazy danych musi obsłużyć następujące sytuacje:

- Nie wolno dodawać wierszy do tabeli po stronie „wiele” (dziecka), jeśli nie istnieją odpowiadające im rekordy po stronie „jeden” (rodzica).
- Jeśli rekord-dziecko zawiera pole IDRodzica, wartość IDRodzica musi pasować do istniejącego rekordu w tabeli-rodzicu.
- Nie można zmienić wartości klucza głównego w tabeli nadrzędnej (rodzicu), jeśli zmiana ta spowodowałaby powstanie osieroconych rekordów w tabeli podrzędnej (dziecku).
- Usunięcie wiersza po stronie „jeden” nie może powodować osierocenia odpowiadających mu rekordów po stronie „wiele”.

Jednym z ciekawych efektów stosowania reguł więzów integralności jest fakt, iż istnieje możliwość występowania rekordu-rodzica, któremu nie odpowiada żaden rekord-dziecko. Intuicyjnie taka sytuacja ma sens. W firmie rzeczywiście mogą być pracownicy, którym jeszcze nie wydano czeków z wypłatą. Może się też zdarzyć, że firma Komis samochodowy zatrudni nowego sprzedawcę, który w określonym momencie nie ma żadnych samochodów na sprzedaż. Większości rekordów-rodziców odpowiada jeden lub kilka rekordów-dzieci, ale nie jest to warunek konieczny dla relacyjnych baz danych.

Nieco mniej oczywistym efektem stosowania reguł więzów integralności jest możliwość istnienia rekordu-dziecka, któremu nie odpowiada rekord-rodzic, pod warunkiem że pole klucza obcego rekordu-rodzica nie zawiera wartości. Jak pamiętamy, zgodnie z zasadą więzów integralności każdej wartości w polu klucza obcego tabeli-dziecka musi odpowiadać ta sama wartość w tabeli-

rodzicu. Jeśli jednak pole klucza obcego w rekordzie-dziecku jest puste, reguła więzów integralności nie jest naruszona.

W praktyce w aplikacjach Accessa sytuacja ta zachodzi dość rzadko. Niemal wszystkie pola bazy danych mają pewną wartość domyślną, na przykład zero lub pusty ciąg znaków (""). W przypadku liczbowego klucza obcego Access domyślnie wstawia zero w polu w momencie dodania nowego rekordu do tabeli. Jeśli zero nie istnieje nigdzie w tabeli nadrzędnej, występuje naruszenie zasady więzów integralności i nowy rekord nie będzie dodany. Jedynym sposobem obejścia tej sytuacji jest usunięcie domyślnej wartości zero dla pola klucza obcego lub zapewnienie ustawienia wartości klucza obcego na prawidłową wartość IDRodzica przed dodaniem nowego rekordu do tabeli-dziecka.

NORMALIZACJA DANYCH

Proces podziału danych na wiele tabel nazywa się normalizacją danych. Jest kilka faz normalizacji. Postaci normalne od pierwszej do trzeciej są najłatwiejsze do zrozumienia i zaimplementowania i ogólnie rzecz biorąc, zwykle wystarczają w większości zastosowań. Choć istnieją wyższe poziomy normalizacji, zazwyczaj ignorują je wszyscy projektanci z wyjątkiem najbardziej doświadczonych i najbardziej drobiazgowych.

PIERWSZA POSTAĆ NORMALNA

Pierwsza faza normalizacji nazywana pierwszą postacią normalną wymaga, aby tabela spełniała następującą regułę: Każda komórka tabeli musi zawierać pojedynczą wartość. W tabeli nie mogą występować powtarzające się grupy danych.

Za tabelę uważa się dwuwymiarową strukturę danych, a zapisywanie wielu wartości w polu lub zezwolenie na powtarzające się grupy wewnątrz tabeli sugeruje występowanie trzeciego wymiaru danych. Pierwszą próbę stworzenia tabeli do zarządzania zamówieniami księgarń (tbl ZamówieniaKsiążek1) pokazano na rysunku 6. Zwróćmy uwagę, że niektóre księgarnie zamówiły więcej niż jedną książkę. Wartość 7 Magia ciastek w polu TytułKsiążki oznacza, że klient zamówił 7 egzemplarzy książki pod tytułem Magia ciastek.

IDZamów	DataZamów	Klient	TytułKsiążki	Dodaj nowe pole
1	2007-05-12	Księgarnia Miejska	2 Łatwe Sushi, 10 Wędzonki z dziczyzny, 5 Dania z fas	
2	2007-05-17	Mania Książek	3 Szalona kapusta	
3	2007-05-23	Księgarnia Miejska	3 Nowe warzywa dla wegetarian, 1 Gotowanie w podróży	
4	2007-05-27	Książki u Jurka	7 Magia ciastek	
5	2007-06-01	Księgarnia "Na rogu"	8 Gotowanie dla dwunastu, 1 Posiłki zdrowotne	
6	2007-06-03	Książki i nie tylko	3 Szybkie obiady, 3 Szybkie kolacje, 6 Szybkie przeką	
7	2007-06-07	Książki u Kowalskiego	1 Odjazdowe potrawy z kurczaka, 1 Wędzonki domowe	
8	2007-06-10	Księgarnia Domowa	2 Wędzonki domowe	
9	2007-06-12	Książki i nie tylko	4 Zasady żywienia domowego	
10	2007-06-13	Księgarnia Uniwersyte	3 Wszystko o makaronach, 2 Smażenie frytek	
*	(Nowy)			

Rys. 6. Nieznormalizowana tabela tblZamówieniaKsiążek

Tabela na rysunku 6 prezentuje zastosowanie kartotekowych baz danych. Dane w kartotekowej bazie danych są zapisane w dwóch wymiarach (wierszach i kolumnach).

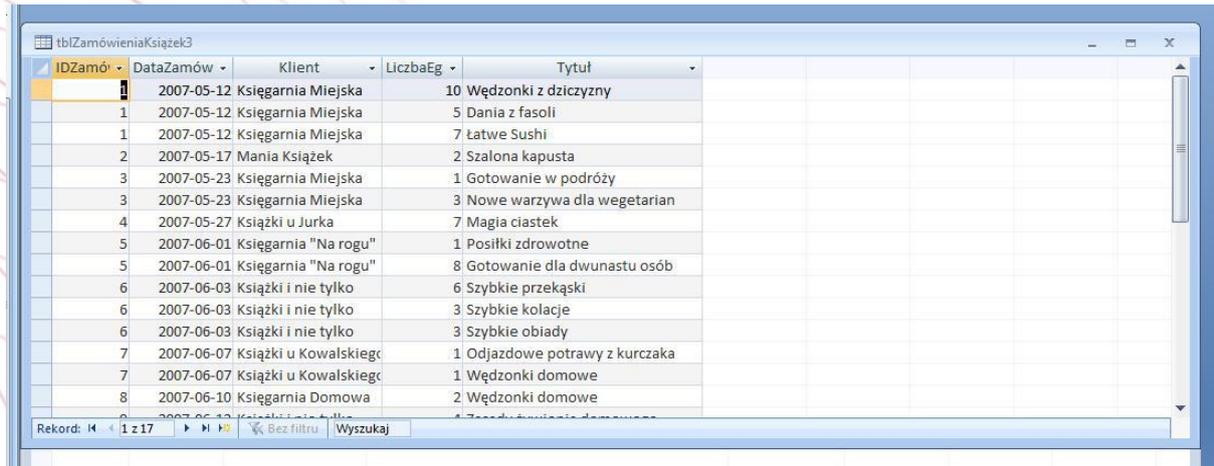
Należy zwrócić uwagę, że tabela na rysunku 6 narusza regułę pierwszej postaci normalnej. Pole TytułKsiążki w wielu rekordach w tej tabeli zawiera więcej niż jedną wartość. Na przykład książka pt. Domowe wędzonki występuje w rekordzie 7. i 8. Nie ma łatwego sposobu obsługi danych w tej postaci przez aparat bazy danych. Na przykład aby odwołać się do książek zamówionych przez księgarnie, trzeba by przeanalizować dane zapisane w polu TytułKsiążki i na tej podstawie stwierdzić, jakie książki zamówiły poszczególne księgarnie.

Nieco lepszy projekt pokazano na rysunku 7. Liczby egzemplarzy poszczególnych książek przeniesiono do osobnych kolumn. Dzięki takiemu projektowi odczytanie informacji o liczbie egzemplarzy określonych tytułów jest nieco łatwiejsze. W dalszym ciągu jednak występują powtarzające się grupy dla liczb i tytułów, które naruszają regułę pierwszej postaci normalnej.

IDZamów	DataZamów	Klient	LiczbaEg	Tytuł1	LiczbaE	Tytuł2	LiczbaEgzer	Tytuł3	LiczbaEgzer	Tytuł4	LiczbaEgzer
1	2007-05-12	Księgarnia Miejska	2	Łatwe Sushi	10	Wędzonki z dziczyzny	5	Dania z fasoli			
2	2007-05-17	Mania Książek	3	Szalona kapusta							
3	2007-05-23	Księgarnia Miejska	3	Nowe warzywa dla wegetarian	1	Gotowanie w podróży					
4	2007-05-27	Książki u Jurka	7	Magia ciastek							
5	2007-06-01	Księgarnia "Na rogu"	8	Gotowanie dla dwunastu	1	Posiłki zdrowotne					
6	2007-06-03	Książki i nie tylko	3	Szybkie obiady	3	Szybkie kolacje	6	Szybkie przekąski			
7	2007-06-07	Książki u Kowalskiego	1	Odjazdowe potrawy z kurczaka	1	Wędzonki domowe					
8	2007-06-10	Księgarnia Domowa	2	Wędzonki domowe							
9	2007-06-12	Książki i nie tylko	4	Zasady żywienia domowego							
10	2007-06-13	Księgarnia Uniwersytecka	3	Wszystko o makaronach	2	Smażenie frytek					
*	(Nowy)		0		0		0		0		0

Rys. 7. Poprawiony projekt tabeli w stosunku do poprzedniego

Na rysunku 8 pokazano tabelę tblZamówieniaKsiążek w pierwszej postaci normalnej. Zamiast umieszczania zamówień wielu książek, dodano drugą tabelę, w której każdy rekord zawiera informację o osobnym tytule zamówionym przez klienta. Potrzebnych jest więcej rekordów, ale obsługa danych jest znacznie łatwiejsza. Pierwsza postać normalna jest bardziej wydajna, ponieważ w tabeli nie ma nieużywanych pól. Każde pole spełnia rolę zgodną z przeznaczeniem tabeli.



IDZamów	DataZamów	Klient	LiczbaEg	Tytuł
	2007-05-12	Księgarnia Miejska	10	Wędzonki z dziczyzny
1	2007-05-12	Księgarnia Miejska	5	Dania z fasoli
1	2007-05-12	Księgarnia Miejska	7	Łatwe Sushi
2	2007-05-17	Mania Książek	2	Szalona kapusta
3	2007-05-23	Księgarnia Miejska	1	Gotowanie w podróży
3	2007-05-23	Księgarnia Miejska	3	Nowe warzywa dla wegetarian
4	2007-05-27	Książki u Jurka	7	Magia ciastek
5	2007-06-01	Księgarnia "Na rogu"	1	Posiłki zdrowotne
5	2007-06-01	Księgarnia "Na rogu"	8	Gotowanie dla dwunastu osób
6	2007-06-03	Książki i nie tylko	6	Szybkie przekąski
6	2007-06-03	Książki i nie tylko	3	Szybkie kolacje
6	2007-06-03	Książki i nie tylko	3	Szybkie obiady
7	2007-06-07	Książki u Kowalskiego	1	Odjazdowe potrawy z kurczaka
7	2007-06-07	Książki u Kowalskiego	1	Wędzonki domowe
8	2007-06-10	Księgarnia Domowa	2	Wędzonki domowe

Rys. 8. Tabela w pierwszej postaci normalnej

Optymalizacja projektu tabeli nie kończy się jednak w tym momencie. Z tabelą ZamówieniaKsiążek, a także z innymi tabelami w tej aplikacji trzeba wykonać jeszcze wiele działań. W szczególności tabela pokazana na rysunku 8 zawiera wiele redundantnych informacji. Tytuły książek powtarzają się za każdym razem, kiedy klienci zamówią tę samą książkę, a numer zamówienia i data zamówienia powtarzają się dla każdego wiersza zawierającego informacje o zamówieniu.

Bardziej subtelny błąd polega na tym, że pola IDKlienta nie można używać jako klucza głównego tabeli. Ponieważ identyfikator zamówienia powtarza się dla każdego tytułu książki w zamówieniu, nie można go używać do identyfikacji każdego rekordu w tabeli. Zamiast tego polem kluczowym jest IDZamówienia. Można z niego korzystać w celu wyszukiwania wszystkich rekordów odpowiadających określonej zamówieniu. Sytuację zostanie naprawiona poprzez wykonanie drugiej fazy optymalizacji.

DRUGA POSTAĆ NORMALNA

Bardziej wydajny projekt można uzyskać w wyniku rozdzielania danych w tabeli tblZamówieniaKsiążek na dwie osobne tabele w celu uzyskania drugiej postaci normalnej. W pierwszej tabeli będą się znajdowały informacje dotyczące zamówienia (na przykład

IDZamówienia, DataZamówienia i IDKlienta), natomiast w drugiej szczegółowe informacje związane z zamówieniem (LiczbaEgzemplarzy i Tytuł). Proces ten bazuje na drugiej regule normalizacji: Dane, które nie zależą bezpośrednio od klucza głównego tabeli, należy przenieść do osobnej tabeli.

Reguła ta oznacza, że tabela powinna zawierać dane reprezentujące jeden podmiot. Tabela pokazana na rysunku 4 narusza tę regułę normalizacji, ponieważ indywidualne tytuły książek nie zależą od pola kluczowego tabeli - IDZamówienia. Każdy rekord zawiera kombinację informacji o książkach i zamówieniach.

Na pierwszy rzut oka mogłoby się wydawać, że tytuły książek zależą od identyfikatora zamówienia. Przecież powodem, dla którego znalazły się w tabeli, jest fakt, iż należą do zamówienia. Wystarczy jednak chwilę pomyśleć, aby zauważyć naruszenie zasad drugiej postaci normalnej. Tytuł książki zupełnie nie zależy od zamówienia, w którym go uwzględniono.

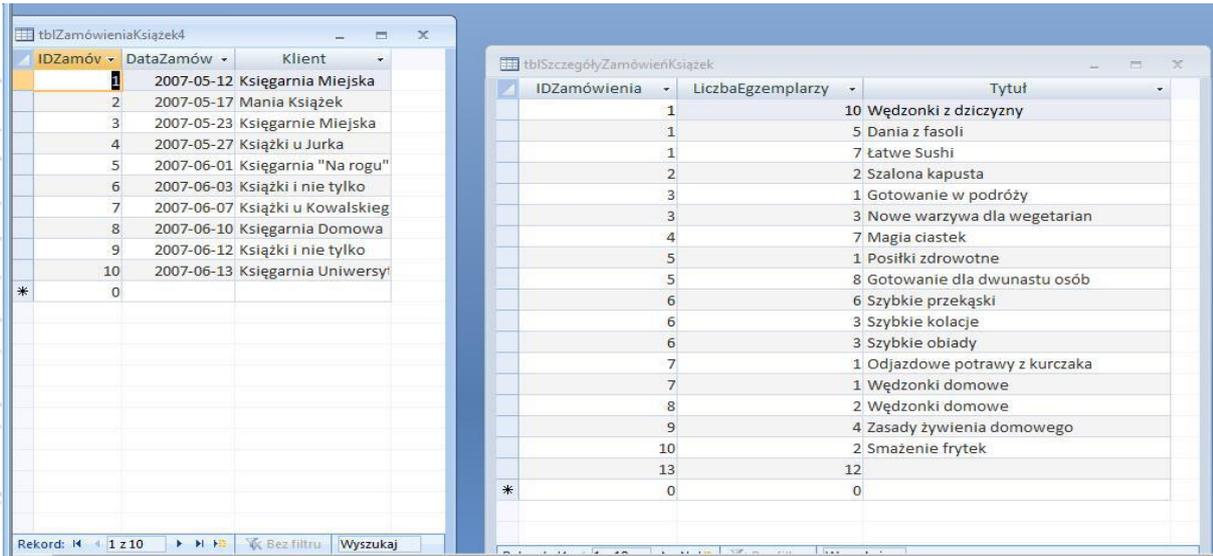
Ten sam tytuł książki pojawia się w wielu zamówieniach, dlatego identyfikator zamówienia nie ma nic wspólnego z tytułem książki. Na podstawie dowolnego identyfikatora zamówienia nie można niczego powiedzieć na temat książek, które są w nim zapisane, poza tym, co znajduje się w tabeli Zamówienia.

Natomiast DataZamówienia całkowicie zależy od identyfikatora zamówienia. Każdemu identyfikatorowi zamówienia odpowiada jedna i tylko jedna data zamówienia. Tak więc każda DataZamówienia zależy od przypisanego do niej identyfikatora zamówienia. Daty zamówień mogą się oczywiście powtarzać w tabeli, ponieważ jednego dnia może wpłynąć wiele zamówień. Natomiast każdemu identyfikatorowi zamówienia odpowiada jedna i tylko jedna wartość daty zamówienia.

W pokazanym przykładzie drugą postać normalną osiągnięto poprzez podział informacji o książkach i zamówieniach na osobne tabele: Zamówienia i SzczegółyZamówień.

Informacje specyficzne dla zamówienia (na przykład data zamówienia, klient, płatność i informacje o wysyłce) są zapisane w tabeli Zamówienia, z kolei szczegóły każdej pozycji zamówienia (tytuł książki, liczba egzemplarzy, cena sprzedaży itp.) są zapisane w tabeli SzczegółyZamówień (nie wszystkie spośród tych danych występują w tabelach użytych w naszych przykładach).

Nowe tabele pokazano na rysunku 9. Kluczem głównym tabeli tblZamówieniaKsiążek4 jest pole IDZamówienia. Pole IDZamówienia w tabeli tblSzczegółyZamówieńKsiążek to klucz obcy, który odwołuje się do klucza głównego IDZamówienia w tabeli tblZamówieniaKsiążek4. Każde pole w tabeli tblZamówieniaKsiążek4 (DataZamówienia i Klient) zależy od klucza głównego tabeli.



IDZamow	DataZamow	Klient
1	2007-05-12	Księgarnia Miejska
2	2007-05-17	Mania Książek
3	2007-05-23	Księgarnie Miejska
4	2007-05-27	Książki u Jurka
5	2007-06-01	Księgarnia "Na rogu"
6	2007-06-03	Książki i nie tylko
7	2007-06-07	Książki u Kowalskiego
8	2007-06-10	Księgarnia Domowa
9	2007-06-12	Książki i nie tylko
10	2007-06-13	Księgarnia Uniwersytetu
*	0	

IDZamowienia	LiczbaEgzemplarzy	Tytuł
1	10	Wędzonki z dziczyzny
1	5	Dania z fasoli
1	7	Łatwe Sushi
2	2	Szalona kapusta
3	1	Gotowanie w podróży
3	3	Nowe warzywa dla wegetarian
4	7	Magia ciastek
5	1	Posiłki zdrowotne
5	8	Gotowanie dla dwunastu osób
6	6	Szybkie przekąski
6	3	Szybkie kolacje
6	3	Szybkie obiady
7	1	Odjazdowe potrawy z kurczaka
7	1	Wędzonki domowe
8	2	Wędzonki domowe
9	4	Zasady żywienia domowego
10	2	Smażenie frytek
13	12	
*	0	

Rys. 9. Druga postać normalna.

Tabele tblZamowieniaKsiazek4 oraz tblSzczegolyZamowienKsiazek powiązane w relację „jeden do wielu”. Tabela tblSzczegolyZamowienKsiazek zawiera tyle rekordów dla każdego zamówienia, ile potrzeba do wypełnienia wymagań zamówienia. Pole IDZamowienia z tabeli tblZamowieniaKsiazek4 jest teraz rzeczywistym kluczem głównym. Każde pole z tabeli tblZamowieniaKsiazek4 zależy od pola IDZamowienia i występuje tylko raz dla każdego złożonego zamówienia. Pole IDZamowienia w tabeli tblSzczegolyZamowienKsiazek nie spełnia roli klucza głównego tej tabeli. W rzeczywistości w tabeli tblSzczegolyZamowienKsiazek w ogóle nie ma klucza głównego, choć bez trudu można go dodać.

TRZECIA POSTAĆ NORMALNA

Ostatnia faza normalizacji, zwana trzecią postacią normalną wymaga usunięcia wszystkich pól, które można uzyskać na podstawie danych zawartych w innych polach tabeli bądź innych tabel bazy danych. Załóżmy, że menedżer handlowy upiera się, aby w tabeli z zamówieniami dodać pole zawierające całkowitą kwotę zamówienia książek. Informacje te można oczywiście obliczyć na podstawie pola Ilość w tabeli tblSzczegolyZamowienia oraz ceny jednostkowej książki pochodzącej z tabeli informacji o książkach.

Nie ma powodu, dla którego należałoby dodawać nowe pole RazemZamowienie do tabeli Zamowienia. Access z łatwością może obliczyć tę wartość na podstawie danych dostępnych w bazie danych. Jedyną zaletą zapisywania całkowitej kwoty zamówienia w bazie danych jest zaoszczędzenie kilku milisekund, jakich Access potrzebuje do odczytania z bazy danych i obliczenia wartości potrzebnych w formularzu lub raporcie.

Usuwanie obliczonych danych z bazy danych nie ma wiele wspólnego z utrzymywaniem bazy danych. Najważniejszą korzyścią jest zaoszczędzenie miejsca na dysku i w pamięci oraz

zmniejszenie intensywności ruchu sieciowego. W zależności od tworzonej aplikacji można znaleźć istotne powody, dla których warto zapisywać obliczane dane w tabelach, zwłaszcza gdy wykonywanie obliczeń trwa długo lub jeśli zapisane wartości są potrzebne do sprawdzenia obliczonych wartości drukowanych na raportach. Może się okazać, że wykonywanie obliczeń podczas wprowadzania danych (kiedy dane są przetwarzane rekord po rekordzie) jest bardziej wydajne niż podczas drukowania raportów (kiedy wygenerowanie pojedynczego raportu wymaga przetworzenia wielu tysięcy rekordów).

Chociaż możliwe są także wyższe poziomy normalizacji, w większości aplikacji baz danych trzecia postać normalna w zupełności wystarcza. Minimalnym wymaganiem jest dążenie do pierwszej postaci normalnej polegające na przeniesieniu redundantnych, powtarzających się danych do innej tabeli.

TABELE

TABELE

Przed przystąpieniem do tworzenia tabel w bazie danych za pomocą narzędzi Accessa warto zaprojektować tabele na papierze. Wiele tabel, zwłaszcza tych nieskomplikowanych, nie wymaga wielkiego planowania przed wprowadzeniem ich do bazy danych. W szczególności nie trzeba zbyt wiele planować, jeśli chce się zaprojektować tabelę zawierającą odnośniki takie jak nazwy miast i województw. Prawidłowe zaprojektowanie tabel z informacjami o bardziej złożonych podmiotach, na przykład klientach lub produktach, wymaga jednak znacznego wysiłku i przemyśleń.

ISTOTA KONWENCJI NAZEWNICTWA

W miarę wzrostu rozmiarów i stopnia złożoności bazy danych wzrasta konieczność ustalenia konwencji nazewnictwa dla obiektów w bazie danych. Jak już dowiedzieliśmy się wcześniej, zmiana nazwy obiektu nie jest propagowana na wszystkie obiekty w bazie danych. Nawet jeśli włączy się Opcje autokorekty nazw {Przycisk Office/Opcje programu Access/Bieżąca baza danych/Opcje autokorekty nazw), Access skoryguje tylko najbardziej oczywiste sytuacje modyfikacji nazw. Zmiana nazwy tabeli powoduje zaprzestanie działania praktycznie każdej kwerendy, formularza czy raportu, które z niej korzystają. Najlepszą linią obrony jest stosowanie sensownych nazw obiektów oraz przyjęcie właściwej konwencji nazewnictwa na wczesnym etapie projektowania baz danych Accessa i przestrzeganie jej w całym projekcie.

W Accessie obowiązuje bardzo niewiele ograniczeń dotyczących nazw przypisywanych obiektom baz danych. W związku z tym istnieje możliwość istnienia dwóch zupełnie różnych obiektów (na przykład formularza i raportu lub tabeli i makra) o tych samych nazwach (nie można

jednak zastosować tej samej nazwy dla tabeli i kwerendy, ponieważ tabele i kwerendy zajmują tę samą przestrzeń nazw w bazie danych).

Najprostszą konwencją nazewnictwa jest poprzedzenie nazw obiektów ciągiem trzech lub czterech znaków, które wskazują na typ obiektu objętego określoną nazwą. Zgodnie z tą konwencją nazwy tabel poprzedza się prefiksem tbl, natomiast kwerend – qry. Prefiksy dla formularzy, raportów, makr i modułów to odpowiednio frm, rpt, mcr i bas lub mod.

Tworzenie struktury tabeli można podzielić na kilka etapów. Wykonując je kolejno, można utworzyć czytelną strukturę tabeli przy minimalnym wysiłku:

1. Utworzenie nowej tabeli.
2. Wpisanie nazwy tabeli, typu danych oraz (opcjonalnie) opisu dla każdego pola.
3. Wpisanie właściwości dla pól.
4. Ustalenie klucza głównego tabeli.
5. Utworzenie indeksów dla wybranych pól.
6. Zapisanie struktury tabeli.

TYPY DANYCH DOSTĘPNYCH W ACCESS

W Accessie można wybrać dla pola dowolne spośród kilku dostępnych typów danych:

- Tekst: znaki alfanumeryczne. Od 0 do 255 znaków.
- Nota: znaki alfanumeryczne, bardzo długie ciągi znaków. Od 0 do 65 536 (64 K) znaków.
- Liczba: wartości liczbowe wielu typów i formatów.
- Data/godzina: data oraz czas.
- Waluta: wartości pieniężne.
- Autonumerowanie: liczba zwiększana automatycznie.
- Tak/Nie: wartości logiczne: Tak/Nie, Prawda/Falsz.
- Obiekt OLE: zdjęcia, wykresy, dźwięki, pliki filmowe, pliki edytorów tekstów i arkusze kalkulacyjne.
- Hiperłącze: łączy do zdjęcia, wykresu, dźwięku, pliku filmowego, pliku edytora tekstu lub arkusza kalkulacyjnego.

Jeden z tych typów danych trzeba przypisać do każdego z pól. Trzeba także określić długość pól typu Tekst lub zaakceptować wartość domyślną, która w przypadku typu Tekst wynosi 255 znaków.

PROJEKTOWANIE REGUŁ WPROWADZANIA DANYCH

Ostatnia ważna decyzja podczas projektowania dotyczy sprawdzania poprawności danych, które ma bardzo istotne znaczenie w czasie, gdy użytkownicy wprowadzają dane. Trzeba zapewnić, aby do systemu przedostały się tylko poprawne dane (tzn. takie, które spełniają określone testy). Jest kilka rodzajów sprawdzania poprawności danych. Można sprawdzać, czy pole zawiera znane, pojedyncze wartości, zastrzegając, że na przykład w polu Płeć dopuszczalne są tylko trzy

wartości Mężczyzna, Kobieta bądź Nieznana. Można również sprawdzać, czy wprowadzona wartość mieści się w dozwolonym zakresie, na przykład czy pole Waga ma wartość z przedziału 0-150 kg.

PROJEKTOWANIE TABEL ODNOŚNIKÓW

Czasami w celu sprawdzenia poprawności danych lub w celu ułatwienia korzystania z systemu trzeba zaprojektować specjalne tabele. Są to tzw. tabele odnośników (ang. lookup tables). Na przykład: ponieważ w systemie Komis samochodowy trzeba wprowadzać stawkę podatkową dla klienta, można użyć tabeli odnośników zawierającej strefę i stawkę podatkową. Innym przykładem, kiedy tabela odnośników może być potrzebna, jest sytuacja, kiedy klient opłaca fakturę, wykorzystując specyficzną metodę - gotówka, karta kredytowa, polecenie zapłaty itp.

Ponieważ stawki podatkowe się zmieniają, bardziej sensowne jest zapisanie ich w bazie danych niż zakodowanie „na sztywno” w aplikacji. Dzięki zastosowaniu tabeli odnośników podczas tworzenia faktury Access będzie poszukiwał bieżącej stawki podatkowej w tabeli tblStawkiPodatkowe. Stawka podatkowa zastosowana do faktury będzie zapisana razem z innymi danymi faktury w tabeli Faktury (Transakcje), ponieważ są to dane zależne od czasu, a wartość zapisana w tabeli tblStawkiPodatkowe może się w przyszłości zmienić.

TYPY DANYCH PÓL

Dziesięć podstawowych typów danych dostępnych w Accessie zestawiono w tabeli 1. Niektóre typy danych (na przykład liczby) mają kilka wariantów.

Typ danych	Rodzaj zapisywanych danych	Rozmiar
Tekst	Znaki alfanumeryczne	0 – 255 znaków
Nota	Znaki alfanumeryczne	0 – 65 536 znaków
Liczba	Wartości liczbowe	1, 2, 4 lub 8 bajtów, 16 bajtów dla identyfikatora replikacji (GUID)
Daty/godziny	Data oraz czas	8 bajtów
Waluta	Wartości pieniężne	8 bajtów
Autonumerowanie	Liczba zwiększana automatycznie	4 bajty, 16 bajtów dla identyfikatora replikacji (GUID)
Tak/Nie	Wartości logiczne: <i>Tak/Nie, Prawda/Falsz</i>	1 bit (0 lub –1)
Obiekt OLE	Zdjęcia, rysunki, dźwięki, pliki filmowe	Do 1 GB (oraz ograniczenie wielkością dysku)
Hiperłącze	Łącze do zasobu internetowego	0 – 64 000 znaków
Załącznik	Specjalne pole (wprowadzone w Accessie 2007) umożliwiające dołączanie zewnętrznych plików do bazy danych Accessa	Różny dla różnych załączników
Kreator odnośników	Wyświetla dane z innej tabeli	Zazwyczaj 4 bajty

Tabela 1. Typy danych dostępne w systemie Microsoft Access

Oto podstawowe reguły, które należy uwzględnić podczas wybierania typu danych dla nowych pól w tabeli:

- **Typ danych powinien odzwierciedlać rodzaj danych zapisanych w polu.** Na przykład należy wybrać jeden z liczbowych typów danych do zapisywania liczb oznaczających ilości bądź ceny. Nie należy jednak zapisywać numerów telefonów bądź numerów NIP jako danych liczbowych. Aplikacja nie będzie wykonywać operacji typowych dla liczb, takich jak dodawanie czy mnożenie, na numerach telefonów, dlatego tych danych nie należy zapisywać w polach typów liczbowych. Zamiast nich dla pól zawierających często wykorzystywane dane, takie jak numery NIP lub numery telefonów, należy wykorzystywać pola tekstowe.
- **Dla wybranego typu danych należy uwzględnić wymagania dotyczące miejsca na dysku.** Chociaż można użyć typu Liczba całkowita długa zamiast typu Liczba całkowita lub Bajt, to trzeba pamiętać, że wymagania dotyczące miejsca na dysku w przypadku liczby całkowitej długiej (4 bajty) to dwa razy więcej w porównaniu ze zwykłą liczbą całkowitą. Oznacza to, że do wykonywania operacji na tych liczbach potrzeba dwa razy tyle pamięci oraz dwa razy tyle miejsca na dysku do ich zapisania. Tam, gdzie to możliwe, do przechowywania prostych danych liczbowych należy używać typu Bajt lub typu Liczba całkowita.

- **Należy zastanowić się, czy będzie potrzebne sortowanie lub indeksowanie pól.** Z uwagi na binarny charakter typów danych Nota i Obiekt OLE nie można indeksować lub sortować pól wykorzystujących te typy. Z pól typu Nota należy korzystać rozważnie. Koszt obliczeniowy związany z zapisywaniem i przetwarzaniem danych tego typu jest znaczny.
- **Należy uwzględnić wpływ typu danych na wymagania dotyczące sortowania.** Dane liczbowe sortuje się inaczej niż dane tekstowe. W przypadku używania liczbowych typów danych sekwencja liczb będzie posortowana zgodnie z oczekiwaniami: 1, 2, 3, 4, 5, 10, 100. Ta sama sekwencja liczb zapisana w postaci tekstu będzie posortowana w następujący sposób: 1, 10, 100, 2, 3, 4, 5. W przypadku potrzeby posortowania danych tekstowych w sposób właściwy dla liczb przed sortowaniem należy zastosować dla danych funkcję konwersji. Jeśli dane tekstowe reprezentujące liczby mają być posortowane w porządku zgodnym z ich wartościami, można je poprzedzić zerami (001, 002 itp). W takim przypadku dane tekstowe będą posortowane zgodnie z oczekiwaniami - 001, 002, 003, 004, 005, 010, 100.
- **Czy dane są tekstem, czy reprezentują daty?** W przypadku przetwarzania dat zazwyczaj lepiej zapisywać je w polu typu Data/Godzi niż w polu typu Tekst. Dane tekstowe są sortowane inaczej niż daty (wewnętrznie daty są przechowywane jako wartości liczbowe). Może to spowodować problemy w raportach lub innych wynikach, które powinny być wyświetlone w sposób chronologiczny.
- **Należy pamiętać o potrzebnych raportach.** W raportach nie można sortować bądź grupować danych typu Nota lub Obiekt OLE. Jeśli trzeba przygotować raport na podstawie danych typu Nota lub Obiekt OLE, należy dodać pole znacznika, na przykład datę lub numer porządkowy, i wykorzystać je jako klucz sortowania tabeli.

DANE TEKSTOWE

Dowolne dane składające się ze znaków (liter, cyfr i znaków przestankowych). Danymi tekstowymi są wszystkie nazwiska, adresy czy opisy, a także dane liczbowe, które nie są wykorzystywane do obliczeń (na przykład numery telefonów, numery NIP, kody pocztowe).

Chociaż dla każdego pola tekstowego w obszarze właściwości określa się jego długość, nie można wpisać do żadnego z nich więcej niż 255 znaków. Do przechowywania danych tego typu Access używa pól o stałej długości. Jeżeli, na przykład, ustalimy długość pola na 25 znaków, ale w poszczególnych rekordach będą występować wpisy nie dłuższe niż 5 znaków, to w przestrzeni bazy danych będzie zajęta ilość miejsca potrzebna do zapisania 5 znaków.

TYP DANYCH NOTA

Typ danych Nota przechowuje w każdym rekordzie zmienną liczbę znaków (od 0 do 65 536). Stąd, jeżeli na przykład w jednym rekordzie wpisano 100 znaków, w innym tylko 10, a jeszcze

w innym 3000, to dla każdego z tych rekordów w bazie wykorzystane będzie dokładnie tyle miejsca, ile danych on zawiera.

Należy zwrócić uwagę, że dla danych typu Nota nie określa się rozmiaru pola. Access przydziela tyle miejsca, ile potrzeba do zapisania danych tego typu.

TYP DANYCH LICZBA

Typ danych Liczba pozwala na przechowywanie danych liczbowych, czyli liczb, które będą brały udział w obliczeniach matematycznych (w przypadku danych wykorzystywanych w obliczeniach finansowych, należy ustalić dla nich typ Walutowy, który pozwala na wykonywanie obliczeń bez błędów zaokrążeń).

Dokładny typ danych liczbowych zapisanych w polu typu Liczba jest określony przez właściwość Rozmiar pola. W tabeli 2 zestawiono różne liczbowe typy danych wraz z ich maksymalną i minimalną wartością, liczbą miejsc dziesiętnych oraz rozmiarem (w bajtach) potrzebnym do przechowywania każdego z typów liczbowych.

Ustawienie w polu Rozmiar pola	Zakres	Miejsca dziesiętne	Rozmiar w pamięci
Bajt	0 do 255	Brak	1 bajt
Liczba całkowita	-32 768 do 32 767	Brak	2 bajty
Liczba całk. długa	-2 147 483 648 do 2 147 483 647	Brak	4 bajty
Podwójna precyzja	$-1,797 \times 10^{308}$ do $1,797 \times 10^{308}$	15	8 bajtów
Pojedyncza precyzja	$-3,4 \times 10^{38}$ do $3,4 \times 10^{38}$	7	4 bajty
Identyfikator replikacji	Nie dotyczy	Nie dotyczy	16 bajtów
Dziesiętne	Dokładność 1 – 28	15	8 bajtów

Tabela 2. Typy danych dostępne w systemie Microsoft Access

TYP DANYCH DATA/GODZINA

Typ danych Data/Godzina to specjalistyczny typ liczbowy do przechowywania dat lub godzin (lub dat razem z godzinami). Po zapisaniu dat w polach typu Data/Godzina można z łatwością obliczyć liczbę dni pomiędzy dwoma datami oraz wykonać inne operacje dotyczące kalendarza. Dаты zapisane w polach typu Data/Godzina można również bez przeszkód sortować i filtrować.

WALUTA

Typ danych Waluta to kolejny, specjalistyczny typ danych liczbowych. Dane typu Waluta nie są zaokrąglane podczas obliczeń i zachowują 15 cyfr precyzji z lewej strony kropki dziesiętnej i 4 cyfry z prawej. Ponieważ w danych typu Waluta pozycja kropki dziesiętnej jest stała, są one szybsze w obliczeniach liczbowych w porównaniu z liczbami typu Podwójna precyzja.

AUTONUMEROWANIE

Typ danych Autonumerowanie to jeszcze jeden specjalistyczny typ danych liczbowych. W przypadku dodania do tabeli pola typu Autonumerowanie Access automatycznie przypisuje liczbę całkowitą do pola (począwszy od 1) i inkrementuje tę wartość każdorazowo po dodaniu rekordu do tabeli. Alternatywnie (w zależności od właściwości Nowe wartości) wartość pola typu Autonumerowanie jest losową liczbą całkowitą automatycznie wprowadzaną do nowych rekordów.

W tabeli może występować tylko jedno pole typu Autonumerowanie. Po przypisaniu wartości Autonumerowanie wewnątrz rekordu nie można go zmienić ani programowo, ani przez użytkownika. Pola typu Autonumerowanie są odpowiednikami typu danych Liczba całk. długa – zajmują cztery bajty, ale za ich pomocą można wyświetlać tylko dodatnie wartości. Zatem zakres dopuszczalnych wartości dla pól Autonumerowanie wynosi od 1 do 4 294 967 296 - to więcej niż wystarczająco do wykorzystania w roli kluczy głównych dla większości tabel.

Tak/Nie

W polach typu Tak/Nie można zapisać tylko jedną z dwóch dozwolonych wartości. Pola te są wewnętrznie zapisane w postaci bitu o wartości 1 lub 0 i służą do wskazywania wartości włączony/wyłączony, tak/nie lub true/false. Pole typu Tak/Nie zajmuje jeden bit.

Obiekt OLE

Pola typu Obiekt OLE służą do zapisywania danych OLE - wysoko specjalizowanych obiektów binarnych, takich jak dokumenty Microsoft Word, arkusze kalkulacyjne Excela, klipy dźwiękowe i filmowe lub zdjęcia. Obiekty OLE tworzy się za pomocą aplikacji serwera OLE. Można ją powiązać z aplikacją macierzystą lub osadzić w tabeli Accessa. W formularzach i raportach Accessa obiekty OLE można wyświetlać wyłącznie w powiązanych ramkach obiektów. Rozmiar obiektów OLE może wynosić 1 GB i więcej. Pól typu Obiekt OLE nie można indeksować.

Załącznik

Typ danych Załącznik został wprowadzony w Accessie 2007. Był jednym z powodów, dla których firma Microsoft zmieniła format pliku danych Accessa. Starszy format pliku – MDB – nie pozwala na zapisywanie załączników.

Typ danych Załącznik w porównaniu z innymi typami pól Accessa jest stosunkowo złożony. Więcej informacji na temat tego interesującego typu danych można znaleźć w podrozdziale „Informacje o typie danych Załącznik” w dalszej części tego rozdziału.

Typ danych Hiperłącze

Typ Hiperłącze przechowuje kombinację tekstu oraz liczb zapisaną jako tekst i wykorzystywaną jako adres hiperłącza. Może zawierać do trzech części:

- widoczną część wyświetlaną w polu (zazwyczaj podkreśloną);
- adres internetowy – ścieżkę dostępu do pliku (w postaci UNC – Universal Naming Convention) lub strony (w postaci URL - Uniform Resource Locator);
- dowolny podadres wewnątrz pliku lub strony; przykładem podadresu jest nazwa raportu czy formularza Accessa 2000; poszczególne części są oddzielone znakiem krzyżyka (#).

Kreator odnośników

Kreator odnośników tworzy pole, które pozwala na wybieranie wartości z innej tabeli lub z listy wyników instrukcji SQL. Wartości mogą być również wyświetlane w postaci pola kombi lub listy. W czasie projektowania tabeli Kreator odnośników przeprowadza programistę przez proces definiowania własności odnośników. Proces ten rozpoczyna się w momencie przypisania tego typu danych do pola.

Przeciągnięcie pola utworzonego za pomocą kreatora odnośników z listy pól podczas tworzenia formularza powoduje automatyczne utworzenie na formularzu pola kombi lub pola listy. Pole listy lub pole kombi wyświetli się również w arkuszu danych kwerendy zawierającej pole.

MASKA WPROWADZANIA

Właściwość Maska wprowadzania ułatwia wprowadzanie danych we właściwym formacie. Maska wprowadzania ogranicza sposób, w jaki użytkownik wprowadza dane do tabeli lub formularza. Na przykład można pozwolić użytkownikowi na wprowadzanie wyłącznie cyfr w polach numeru telefonu, numeru NIP oraz identyfikatora pracownika. Maska wprowadzania dla numeru NIP może mieć następującą postać: 000-000-00-00 Maska ta nakłada obowiązek wpisania wszystkich cyfr numeru NIP bez spacji i innych znaków.

Maska wprowadzania zdefiniowana na poziomie właściwości pola jest wykorzystywana wszędzie tam, gdzie pole jest używane (kwerendy, formularze, raporty).

Wartość właściwości Maska wprowadzania jest ciągiem znaków składającym się z trzech części. Poszczególne części są oddzielone od siebie średnikami.

- Pierwsza część: zawiera właściwą maskę składającą się z symboli zamieszczonych poniżej.
- Druga część: informuje Accessa o tym, czy należy zapisywać literały zawarte w masce razem z pozostałymi danymi. Na przykład w masce wprowadzania mogą się znaleźć

myślniki oddzielające poszczególne części numeru NIP, natomiast maska dla numeru telefonu może zawierać nawiasy i myślniki. Wartość 0 (zero) powoduje zapisywanie literałów wewnątrz danych, natomiast wartość 1 powoduje zapisanie wyłącznie wprowadzonych danych.

- Trzecia część: definiuje znak-wypełniacz, który informuje użytkownika o liczbie znaków w obszarze wprowadzania. W wielu maskach wprowadzania wykorzystuje się w tej roli znaki krzyżyka (#) lub gwiazdki (*).

W definicji masek wprowadzania można używać następujących znaków:

- 0: obowiązkowa cyfra, nie można użyć znaku plus (+) ani minus (-).
- 9: opcjonalna cyfra, nie można użyć znaku plus (+) ani minus (-).
- #: opcjonalna cyfra lub spacja. Spacje są usuwane podczas zapisywania danych w tabeli. Znaki plus i minus są dozwolone.
- L: obowiązkowa litera A-Z.
- ?: opcjonalna litera A-Z.
- A: obowiązkowy znak lub cyfra.
- a: opcjonalny znak lub cyfra.
- &: obowiązkowy dowolny znak lub spacja.
- C: opcjonalny dowolny znak lub spacja.
- , (przecinek): przecinek dziesiętny.
- (spacja): separator tysięcy.
- : (dwukropek): separator dat i godzin.
- ; (średnik): znak separatora.
- - (myślnik): znak separatora.
- / (prawy ukośnik): znak separatora.
- < (mniejszy niż): zamienia wszystkie następne znaki na małe litery.
- > (większy niż): zamienia wszystkie następne znaki na wielkie litery.
- ! (wykrzyknik): powoduje wyświetlanie maski od prawej do lewej. Znaki wypełniają maskę od prawej do lewej.
- \ (lewy ukośnik): wyświetla kolejny znak dosłownie.

Tych samych specyfikatorów można używać w arkuszu właściwości pola w kwerendzie lub formularzu.

Maski wprowadzania są ignorowane podczas importowania danych lub dodawania danych do tabeli za pomocą kwerendy funkcjonalnej.

Właściwość Format zdefiniowana dla pola przesłania właściwość Maska wprowadzania. W takim przypadku maska wprowadzania ma zastosowanie podczas wprowadzania danych. Po zakończeniu wprowadzania format zmienia się zgodnie ze specyfikatorem formatu.

REGUŁA SPRAWDZANIA POPRAWNOŚCI

Właściwość Reguła sprawdzania poprawności definiuje wymagania wprowadzania danych do pola. Reguła sprawdzania poprawności, nad której przestrzeganiem czuwa mechanizm bazy danych Jet, zapewnia zgodność danych wprowadzanych do tabeli z wymaganiami aplikacji.

Właściwość ta nie ma zastosowania do pól wyboru, przycisków opcji bądź przełączników w obrębie grupy opcji na formularzu. Grupa opcji posiada własną właściwość Reguła spr. poprawności, która dotyczy wszystkich formantów w obrębie grupy.

Wartość właściwości Reguła spr. poprawności to ciąg znaków zawierający wyrażenie używane do sprawdzania danych wprowadzanych przez użytkowników. Wyrażenie spełniające rolę właściwości Reguła spr. poprawności nie może zawierać funkcji zdefiniowanych przez użytkownika ani żadnych funkcji domeny Accessa bądź funkcji agregacji. Reguła sprawdzania poprawności dla pola nie może zawierać odwołań do formularzy, kwerend lub innych tabel w aplikacji (ograniczenia te nie dotyczą reguł sprawdzania poprawności stosowanych w odniesieniu do formantów formularzy). Reguły sprawdzania poprawności dla pól nie mogą zawierać odwołań do innych pól w tabeli, ale reguła sprawdzania poprawności zastosowana do rekordu w tabeli może odwoływać się do pól w tej samej tabeli (regułę sprawdzania poprawności poziomu rekordu ustawia się w arkuszu właściwości tabeli, a nie dla indywidualnych pól).

POJĘCIE KLUCZY

Kiedy tworzymy tabele, powinniśmy dla każdej z nich zdefiniować klucz główny - jedno lub kilka pól, których zawartość jest unikalna dla każdego rekordu. Zapewni to, że w tabeli nie będą nigdy istniały rekordy zawierające identyczne wartości, czyli można, na przykład, mieć kilku klientów o nazwisku Michał Kowalski, którzy nawet mogą mieszkać pod tym samym adresem (na przykład ojciec i syn). W takim przypadku trzeba zdecydować, w jaki sposób utworzyć w tabeli tblKlienci rekordy, aby odróżnić od siebie poszczególnych Michałów Kowalskich.

Unikatowa identyfikacja każdego rekordu w tabeli - właśnie to ma zapewnić klucz główny. Na przykład w bazie danych Komis samochodowy pole IDKlienta (unikalny numer, który przypisujesz każdemu klientowi zgłaszającemu się do biura firmy) jest kluczem głównym tabeli tblKlienci – każdy rekord w tabeli ma w tym polu inną wartość (nie istnieją dwa rekordy o takim samym identyfikatorze). Jest to ważne z kilku powodów:

- Nie chcemy mieć w tabeli dwóch rekordów dla tego samego klienta, ponieważ w takim przypadku uaktualnienie informacji dotyczących klienta będzie praktycznie niemożliwe.
- Chcemy mieć pewność, że każdy rekord w tabeli zawiera precyzyjne dane po to, aby informacje pobrane z tabeli także były precyzyjne.
- Nie chcemy, aby tabela (wraz z jej rekordami) była większa, niż jest to konieczne.

Dzięki możliwości przypisania pojedynczej, unikalnej wartości każdemu rekordowi, tabela staje się przejrzysta i czytelna. W terminologii baz danych określa się to mianem integralności jednostkowej. Mając różne wartości klucza głównego w każdym rekordzie (na przykład w polu IDKlienta w tabeli tblKlienci), można zawsze rozróżnić dwa rekordy (w tym przypadku klientów), nawet jeśli pozostałe pola w rekordach są identyczne. Jest to ważne, ponieważ w tabeli łatwo może się znaleźć dwóch klientów o popularnym nazwisku, na przykład Andrzej Kowalski.

Teoretycznie do identyfikacji klienta można by wykorzystać jego nazwisko oraz adres, ale dwóch Andrzejów Kowalskich może mieszkać w tym samym województwie, mieście, a nawet pod tym samym adresem (na przykład ojciec i syn). Celem stosowania kluczy głównych do tworzenia unikalnych rekordów w tabelach jest całkowita gwarancja ich niepowtarzalności.

Jeżeli w trakcie tworzenia tabel w Accessie nie utworzy się kluczy głównych, Access zapyta, czy ma sam to zrobić. Jeżeli odpowiemy twierdząco, Access utworzy klucz główny w postaci pola typu Autonumerowanie. Typ ten powoduje, że do każdego nowego rekordu w polu klucza głównego automatycznie jest wpisywana kolejna liczba oraz nie można zmodyfikować wartości w przypadku, gdy raz zostanie ustawiona. Co więcej, po wprowadzeniu wartości typu Autonumerowanie do tabeli nigdy nie można jej wykorzystać po raz kolejny, nawet wtedy, gdy rekord zawierający tę wartość zostanie usunięty i wartości tej już nie będzie w tabeli.

OKREŚLANIE KLUCZA GŁÓWNEGO

Jak opisano powyżej, tabela zwykle ma unikalne pole (lub kombinację pól), czyli klucz główny, który zapewnia unikalność każdego rekordu. Często jest to pole identyfikatora typu Tekst lub Autonumerowanie. W celu określenia wartości dla pola identyfikatora można zdefiniować metodę generowania unikatowej wartości dla pola. Może to być najprostszą metodą, czyli pozwolenie Accessowi na automatyczne konstruowanie unikalnych wartości, ale można też samemu tworzyć identyfikatory, na przykład jako sklejenie pierwszej litery rzeczywistej wartości z kolejną liczbą (takie jak A001, A002, A003, B001, B002 itd.). Metoda generowania unikalnych wartości może także bazować na losowych kombinacjach liter i liczb (pod warunkiem, że będą one niepowtarzalne dla każdego rekordu) lub opierać się na złożonych obliczeniach wykorzystujących kilka wartości z innych pól tabeli.

Poza pełnieniem funkcji „łącznika” pomiędzy tabelami klucz główny w Accessie zapewnia następujące korzyści:

- Pola kluczy głównych zawsze są poindeksowane, co znacznie przyspiesza wykonywanie kwerend, wyszukiwanie oraz sortowanie.
- Każdorazowo podczas dodawania do tabeli nowych rekordów Access wymusza wprowadzanie w nich wartości (lub automatycznie sam wprowadza wartość w przypadku pól typu Autonumerowanie). Dzięki temu uzyskujemy gwarancję, że baza danych będzie spełniała wymagania węzłów integralności.

- Podczas dodawania nowych rekordów do tabeli ze zdefiniowanym kluczem głównym Access sprawdza, czy dane w kolumnie klucza głównego nie powtarzają się, i zapobiega wprowadzaniu duplikatów. W ten sposób zapewnia integralność danych.
- Domyślnie Access wyświetla dane w kolejności wynikającej z wartości klucza głównego.
-

KWERENDY

Kwerenda w programie Microsoft Access jest pytaniem dotyczącym informacji zapisanych w tabelach Accessa. Kwerendy tworzy się za pomocą odpowiednich narzędzi dostępnych w Accessie, a następnie zapisuje jako oddzielne obiekty w bazie danych. Kwerenda może być prosta, pobierająca dane umieszczone w pojedynczej tabeli lub może mieć formę złożonego zapytania dotyczącego informacji przechowywanych w kilku tabelach. Po zadaniu pytania Microsoft Access zwraca jedynie żądane informacje.

TYPY KWEREND

Access oferuje wiele różnych typów kwerend. Można je pogrupować w sześć podstawowych kategorii:

- **Wybierające:** są to najbardziej popularne typy kwerend. Jak wynika z nazwy, służą do wybierania informacji z jednej lub kilku tabel i tworzenia wynikowego zbioru rekordów. Ogólnie rzecz biorąc, dane zwracane przez kwerendy wybierające można aktualizować. Często wykorzystuje się je jako źródła danych dla formularzy i raportów.
- **Sumujące:** jest to specjalna odmiana kwerend wybierających. Umożliwiają one zwracanie przez kwerendy sum lub innych obliczeń (na przykład liczby wierszy) na podstawie rekordów zwracanych przez kwerendę wybierającą. Wybranie tego typu kwerendy powoduje, że Access dodaje do obszaru QBE (od ang. Query by Example - zapytanie przez przykład) dodatkowy wiersz o nazwie Podsumowanie.
- **Funkcyjne:** kwerendy tego typu umożliwiają tworzenie nowych tabel oraz modyfikowanie danych (usuwanie, aktualizację i dołączanie) istniejących w tabelach. Kwerendy funkcyjne umożliwiają wprowadzanie zmian w wielu rekordach za pomocą pojedynczej operacji.
- **Krzyżowe:** takie kwerendy mogą wyświetlać podsumowania danych w postaci tabelarycznego formularza przypominającego arkusz kalkulacyjny, którego nagłówki wierszy i kolumn odpowiadają polom w tabeli. Poszczególne elementy zestawu rekordów wyliczane są na podstawie danych w tabelach.
- **SQL:** istnieją trzy typy kwerend SQL - Składająca, Przekazująca oraz Definicja danych. Są one wykorzystywane do zaawansowanych operacji na bazach danych (na przykład w bazach SQL typu klient-serwer takich, jak SQL Server lub Oracle). Kwerendy tego typu tworzy się poprzez zapisanie określonych instrukcji SQL.
- **Początkowe(n):** kwerendy typu Początkowe(n) umożliwiają wprowadzenie liczby lub procenta rekordów, które mają się znaleźć w wyniku kwerendy dowolnego typu (wybierającej, podsumowującej, itp.).

MOŻLIWOŚCI KWEREND

Kwerendy są bardzo elastyczne. Umożliwiają przeglądanie danych w praktycznie każdy sposób, jaki można sobie wyobrazić. Większość systemów baz danych ciągle się rozwija i zmienia. Bardzo często zmianie ulega nawet pierwotne przeznaczenie, dla którego powstał system.

Za pomocą kwerend Accessa można osiągnąć:

- Wybieranie tabel: można pobierać informacje zarówno z jednej tabeli, jak i z kilku powiązanych ze sobą za pomocą określonych wspólnych danych.
- Wybieranie pól: można ustalić, które pola z każdej tabeli mają się pojawić w zbiorze wynikowym.
- Wybieranie rekordów: definiując kryteria, można wyświetlić w zbiorze wynikowym tylko wybrane rekordy.
- Sortowanie rekordów: rekordy można sortować w określonym porządku. Na przykład rekordy klientów można posortować według nazwisk i imion.
- Wykonywanie obliczeń: kwerendy można wykorzystywać także do wykonywania różnych obliczeń na danych. Mogą to być, na przykład, obliczanie średniej, sumy oraz zliczanie rekordów.
- Tworzenie tabel: tworzenie nowej tabeli na podstawie danych zwracanych przez zapytanie.
- Tworzenie formularzy i raportów bazujących na kwerendach: zestaw rekordów tworzony na podstawie zapytania może zawierać pola i dane nadające się do wykorzystania w raporcie bądź formularzu.
Tworzenie raportu lub formularza bazującego na kwerendzie oznacza, że przy każdym drukowaniu raportu czy otwarciu formularza kwerenda pobiera najpierw z tabel najbardziej aktualne informacje.
- Tworzenie wykresów w oparciu o kwerendy: na podstawie danych zwracanych przez kwerendy można tworzyć wykresy.
- Używanie kwerend jako źródeł danych dla innych kwerend (podzapytań): tworzenie dodatkowych kwerend bazujących na zbiorze rekordów zwróconych przez poprzednią. Jest to bardzo wygodny sposób wykonywania kwerend ad hoc, gdzie może występować konieczność wielokrotnego wprowadzania niewielkich zmian w kryteriach. W takich sytuacjach drugą kwerendę wykorzystuje się do modyfikowania kryteriów, podczas gdy pierwsza wraz ze zwracanymi przez nią danymi pozostaje nienaruszona.
- Wprowadzanie zmian w tabelach: kwerendy Accessa mogą pobierać dane z szerokiego zakresu źródeł. Można pobierać dane zapisane w systemach dBASE, Paradox, Btrieve czy Microsoft SQL Server, a także arkuszy Excela, plików tekstowych i innych źródeł.

RAPORTY

Występuje pięć podstawowych typów raportów:

- Raport tabelaryczny. Ten raport drukuje dane w wierszach i kolumnach, grupując je i obliczając sumy. Odmianami tego typu raportu są raporty sumaryczny i grupująco-sumujący.
- Raport kolumnowy. Ten raport drukuje dane w postaci formularza. Może zawierać sumy i wykresy.
- Raport korespondencji seryjnej. Służy do tworzenia szablonowych listów.
- Etykiety pocztowe. Ten raport służy do tworzenia wielokolumnowych arkuszy etykiet pocztowych albo do prezentowania danych w wielu kolumnach.
- Wykresy. Wizualna reprezentacja danych, na przykład w postaci wykresu słupkowego lub kołowego.

RAPORTY TABELARYCZNE

Typowy raport typu tabelarycznego w oknie podglądu wydruku pokazano na rysunku 10. Raporty tabelaryczne (zwane też raportami grupująco-sumującymi) przypominają tabelę, w której dane wyświetlają się w wierszach i kolumnach. Raporty tabelaryczne – w przeciwieństwie do formularzy i arkuszy danych – zwykle grupują dane według wartości jednego lub wielu pól. Często raporty tabelaryczne obliczają oraz wyświetlają sumy częściowe albo informacje statystyczne dotyczące pól liczbowych w każdej grupie. Niektóre raporty wyświetlają także sumy stron i sumy łączne. Mogą nawet zawierać zawinięte kolumny, dzięki którym można tworzyć katalogi (takie jak książka telefoniczna). Raporty tego typu mogą zawierać numery stron, daty utworzenia raportu oraz linie i ramki oddzielające informacje. Mogą zawierać kolory i cieniowanie oraz wyświetlać obrazy, wykresy biznesowe i pola typu Nota. Specjalny typ raportu tabelarycznego – raport sumaryczny – ma wszystkie cechy raportu tabelarycznego, ale nie drukuje poszczególnych rekordów.

Produkty - podsumowanie
Komis samochodowy

Kategoria	ID Produktu	Opis	Ilość w magazynie	Koszt	Cena detaliczna	Cena umowna	Zysk
Terenowe							
	35	2002 Olds SUV	3	35 900,00 zł	39 900,00 zł	38 900,00 zł	3 000,00 zł
	36	1995 GMY JIMMIE SLE	1	5 000,00 zł	7 500,00 zł	6 990,00 zł	1 990,00 zł
	37	1995 Bjeep Laredo Red	1	6 000,00 zł	8 900,00 zł	8 650,00 zł	2 650,00 zł
	38	1998 Range Rover	1	12 500,00 zł	14 000,00 zł	13 200,00 zł	700,00 zł
	39	1998 Fordman Explorer XLP	1	9 000,00 zł	12 500,00 zł	11 950,00 zł	2 950,00 zł
	40	1998 Isuzz Rodeo	1	10 000,00 zł	15 100,00 zł	13 999,00 zł	3 999,00 zł
	41	1998 Fordman Expedition XLT	1	16 500,00 zł	19 500,00 zł	18 990,00 zł	2 490,00 zł
	42	1998 Fordman Explorer XLT	1	16 000,00 zł	18 500,00 zł	17 999,00 zł	1 999,00 zł
	43	2002 Red bjeep	1	23 895,00 zł	25 900,00 zł	24 900,00 zł	1 005,00 zł
	Kategoria Terenowe Razem:		11	134 795,00 zł	161 800,00 zł	155 578,00 zł	20 783,00 zł
Wozy kempingowe							
	34	1973 Rare Popup Hard sided Indian	1	1 200,00 zł	1 750,00 zł	1 400,00 zł	200,00 zł
	Kategoria Wozy kempingowe Razem:		1	1 200,00 zł	1 750,00 zł	1 400,00 zł	200,00 zł
	Razem:		12	135 995,00 zł	163 550,00 zł	156 978,00 zł	20 983,00 zł

Strona: 1 z 1 Bez filtru

Rys. 10. Raport tabelaryczny.

RAPORTY KOLUMNOWE

Raporty kolumnowe zwykle wyświetlają jeden lub kilka rekordów na stronie w układzie pionowym. Raporty kolumnowe wyświetlają dane niemal tak samo jak formularze, ale służą wyłącznie do oglądania, a nie do modyfikowania danych. Fragment raportu kolumnowego w widoku podglądu wydruku zamieszczono na rysunku 11.

PRODUKTY

Komis samochodowy

Informacje o produkcji

Opis: 2001 Sportster

Kategoria: Osobowe

Ilość w magazynie: 1

Koszt: 32 656,00 zł

Cena detaliczna: 38 995,00 zł

Cena umowna: 38 999,00 zł

Opodatkowany:

Cechy:

Seledynowy. Odnosił sukcesy w lokalnych rajdach w 2002 r. Nowy silnik i przekładnia.

Zdjęcie:



ID Produktu: 1

Aukcja:

Data zakończenia aukcji: 2003-03-19

Typ aukcji: W firmie Internetow

Sprzedawca: Celina Wrońska
Auto-Celina-Shop
Ul. 11-go Listopada 6
Pińczów, SWT 12301
(003) 555-68-87

Cena wywoławcza:

Cena minimalna:

Zakończona:

Rys. 11. Raport kolumnowy.

Inny typ raportu kolumnowego wyświetla jeden rekord na głównej stronie (tak jak formularz biznesowy), ale może pokazywać wiele rekordów na osadzonych podformularzach. Typowym przykładem takiego raportu jest faktura. Raporty tego typu mają sekcje, które wyświetlają tylko jeden rekord, oraz sekcje, które wyświetlają kilka rekordów ze strony „wiele” relacji „Jeden do wielu”, mogą też zawierać sumy.

Raport faktury z bazy danych Komis samochodowy w oknie podglądu wydruku pokazano na rysunku 12.

Komis samochodowy
Przygoda dla wszystkich!
 123 Main Street
 Clearwater, FL 33764
 Phone: (727) 555-1234
 Fax: (727) 555-6789

FAKTURA
4

Data faktury:
 2007-02-03
 Data sprzedaży:
 2007-01-15

Strona 1 z 1

NABYWCA: Jan Jankowski

Nopa Auto Części
 Pl Wolności 5
 Oława, DLN 32535
 (013) 555-32-16

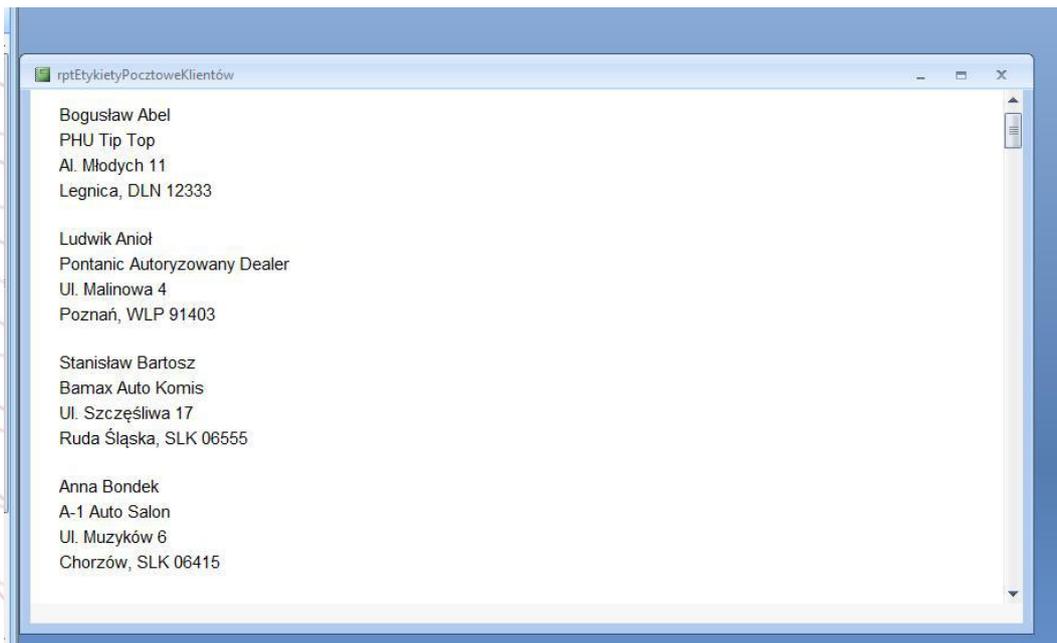
METODA PŁATNOŚCI	SPRZEDAWCA	LOKALIZACJA POJAZDU
Czek	Jan Jankowski	Bobrowniki, DLN

Produkt	Ilość	Opis	Cena	Rabat%	Podatek?	Kwota
	59	1 2003 Ford T-150 Regular Cab	20 400,00 zł	10,00%	<input checked="" type="checkbox"/>	18 360,00
	45	1 1960 Chevy Pickup	6 500,00 zł	5,00%	<input checked="" type="checkbox"/>	6 175,00
	2	1 1957 Chevy Sedan	21 900,00 zł	0,00%	<input checked="" type="checkbox"/>	21 900,00
	4	1 1950 Ford Coupe	19 000,00 zł	0,00%	<input checked="" type="checkbox"/>	19 000,00
	34	1 2002 Olds SUV	39 900,00 zł	0,00%	<input checked="" type="checkbox"/>	39 900,00

Rys. 12. Raport faktury.

ETYKIETY POCZTOWE

Etykiety pocztowe również są typem raportu. Etykiety te pokazano na rysunku 13. Można je łatwo utworzyć za pomocą Kreatora etykiet pocztowych dostępnego w Accessie. Kreator pozwala wybrać rodzaj etykiety z długiej listy formatów firmy Avery (i innych producentów), po czym tworzy projekt raportu na podstawie określonych danych. Po utworzeniu etykiet, można otworzyć raport w trybie projektu i dostosować go do własnych potrzeb.



Rys. 13. Typowy raport etykiet pocztowych.

RÓŻNICE MIĘDZY RAPORTAMI A FORMULARZAMI

Główną różnicą między raportami a formularzami jest przeznaczenie wyświetlanych danych. Formularze służą przede wszystkim do wprowadzania danych i komunikacji z użytkownikami, natomiast raporty - do przeglądania danych (na ekranie lub w postaci drukowanej). Na formularzach używa się pól obliczeniowych, które obliczają pewne wartości na podstawie innych pól. W raportach wykonuje się obliczenia na pewnych grupach rekordów, stronach rekordów albo wszystkich rekordach przetwarzanych podczas tworzenia raportu. Wszystko, co można zrobić na formularzu – z wyjątkiem wprowadzania danych - można również powielić w raporcie. W rzeczywistości można zapisać formularz jako raport, a następnie dostosować formanty w oknie projektu raportu.

PROCEDURA TWORZENIA RAPORTU

Planowanie raportu zaczyna się na długo przed przystąpieniem do tworzenia jego projektu. Punktem wyjścia jest chęć obejrzenia danych w tabeli, ale w sposób, który różni się od widoku arkusza danych. Zaczyna się od projektu takiego widoku; Access rozpoczyna od surowych danych. Celem raportu jest przetworzenie surowych danych w sensowny zbiór informacji. Procedura tworzenia raportu składa się z kilku etapów:

- definiowanie układu raportu,
- gromadzenie danych,
- tworzenie projektu w oknie projektu raportu Accessa,
- drukowanie lub wyświetlanie raportu.

DEFINIOWANIE UKŁADU RAPORTU

Należy zacząć od pomysłu na ogólny układ raportu. Można zdefiniować układ w myślach, na papierze lub interaktywnie, korzystając z okna projektu raportu Accessa. Dobry raport należy najpierw rozplanować na papierze, wskazując potrzebne pola i ich rozmieszczenie. Bardzo często od raportu Accessa oczekujemy powielenia istniejącego raportu papierowego wykorzystywanego przez użytkowników aplikacji.

GROMADZENIE DANYCH

Kiedy już mamy ogólne wyobrażenie co do tego, jaki powinien być układ raportu, powinniśmy zebrać dane potrzebne do jego utworzenia. W raportach Accessa dane pochodzą z dwóch zasadniczych źródeł: pojedynczej tabeli bazy danych lub zestawu rekordów utworzonego w wyniku wykonania kwerendy. W kwerendzie można złączyć wiele tabel i wykorzystać jej zestaw rekordów jako źródło danych raportu. W raporcie Accessa zestaw rekordów kwerendy jest interpretowany tak, jakby był jedną tabelą.

Jak wiadomo, w kwerendzie można wybierać pola, rekordy i kolejność sortowania rekordów. Access traktuje ten zestaw rekordów jak jedną tabelę (do celów przetwarzania) na arkuszach danych, formularzach i raportach. Zestaw rekordów staje się źródłem danych raportu, a Access przetwarza każdy rekord, aby utworzyć raport. Kiedy raport jest wykonywany, Access dopasowuje dane z zestawu rekordów lub tabeli do pól użytych w raporcie i wykorzystuje dane dostępne w określonym momencie do utworzenia raportu.

W Accessie dostępne są następujące sekcje raportu:

- Nagłówek raportu. Drukowany tylko na początku raportu; używany jako strona tytułowa.
- Nagłówek strony. Drukowany na górze każdej strony.
- Nagłówek grupy. Drukowany przed przetworzeniem pierwszego rekordu należącego do grupy.
- Szczegóły. Drukuje każdy rekord z tabeli lub zestawu rekordów kwerendy.
- Stopka grupy. Drukowana po przetworzeniu ostatniego rekordu należącego do grupy.
- Stopka strony. Drukowana na dole każdej strony.
- Stopka raportu. Drukowana na końcu raportu po przetworzeniu wszystkich rekordów.

Literatura

1. <http://office.microsoft.com/pl-pl/access-help/>
2. Feddema H., Microsoft Access. Podręcznik administratora, Wydawnictwo Helion.
3. Groh M.R., *Access 2007 PL. Biblia*, Wydawnictwo Helion.
4. Murray K., Microsoft Office 2010 PL. Praktyczne podejście, Wydawnictwo Helion.