

Zasady projektowania relacyjnych baz danych



1.1. Wprowadzenie do baz danych

Tradycyjne dane przechowywane są w postaci dokumentów papierowych. Dokumenty takie zawierają opisy przechowywanych obiektów. Obiektami mogą być na przykład samochody, książki lub osoby. Opisywane są także związki zachodzące między obiektami — kto jest właścicielem samochodu, kto wypożyczył książkę, gdzie pracuje dana osoba. Dokumenty opisujące obiekty i związki zachodzące między nimi są gromadzone w postaci kartotek, katalogów lub archiwów i są przechowywane w kopertach, teczkach albo segregatorach. Dostęp do tak opracowanych dokumentów jest trudny i zajmuje dużo czasu. Aby przyspieszyć wyszukiwanie danych, sporządza się różne spisy, wyciągi i katalogi.

Dobrym rozwiązaniem tych problemów jest przeniesienie takich dokumentów do komputera. Po zapisaniu danych w pamięci komputera można obsługiwać tak utworzoną bazę danych, korzystając z dostępnych narzędzi.

Zalety korzystania z komputerowych baz danych to:

- szybkie wyszukiwanie informacji,
- łatwe wykonywanie obliczeń,
- możliwość przechowywania dużej ilości danych na małej powierzchni,
- szybkie porządkowanie danych.

WSKAZÓWKA

Baza danych to uporządkowany zbiór danych z określonej dziedziny tematycznej zorganizowany w sposób ułatwiający do nich dostęp.

WSKAZÓWKA

System zarządzania bazą danych to program zarządzający danymi w bazie i umożliwiający ich przetwarzanie.

WSKAZÓWKA

System bazy danych to baza danych i system zarządzania bazą danych.

U podstaw konstruowania bazy danych leży założenie, że użytkownik, dla którego ta baza jest przeznaczona, nie musi być specjalistą z dziedziny baz danych, może w ogóle ich nie znać. Mimo to powinien bez problemów radzić sobie z obsługą zaprojektowanej bazy danych. Aby to było możliwe, podstawą tworzonej bazy musi być solidny projekt określający potrzeby użytkownika dotyczące gromadzenia, przechowywania i przetwarzania danych oraz definiujący czynności składające się na obsługę bazy danych.

1.2. Modele baz danych

Aby przechowywać dane na komputerze, konieczne jest określenie formy ich przechowywania. Na potrzeby baz danych zostały zdefiniowane klasyczne techniki organizowania informacji, zwane modelami baz danych.

Model danych to abstrakcyjny opis sposobu przedstawiania i wykorzystania danych.

Na model danych składają się:

- *struktura* — opis sposobu przedstawiania obiektów (encji) modelowanego wycinka świata oraz ich związków,
- *ograniczenia* — reguły kontrolujące spójność i poprawność danych,
- *operacje* — zbiór działań, które umożliwiają dostęp do struktur.

Głównymi modelami baz danych są:

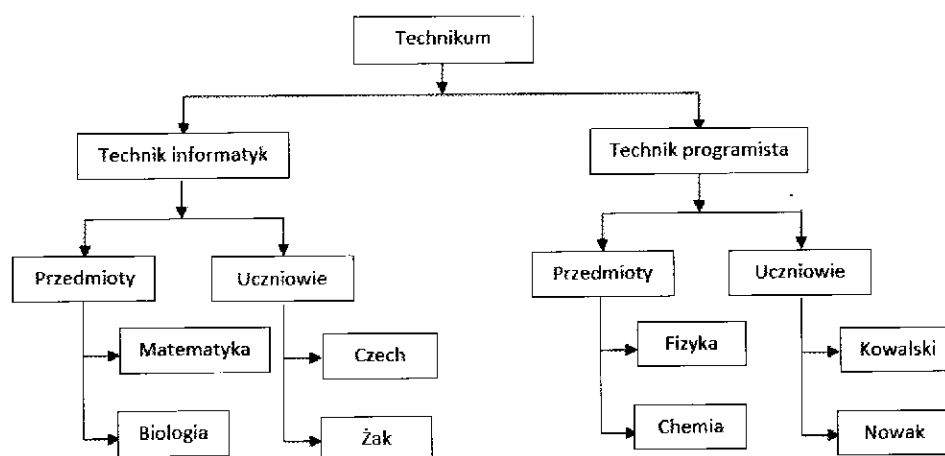
- model hierarchiczny,
- model sieciowy,
- model obiektowy,
- model relacyjny.

1.2.1. Model hierarchiczny

W modelu hierarchicznym przechowywane dane są zorganizowane w postaci odwróconego drzewa (rysunek 1.1). Każdy obiekt (z wyjątkiem obiektu podstawowego) jest połączony z jednym obiektem nadrzędnym.

Tak zbudowana baza danych umożliwia proste wyszukiwanie danych. Rozpoczyna się ono od obiektu podstawowego i poprzez rozgałęzienia oraz kolejne obiekty dochodzi się do obiektu szukanego. Informacja jest zawarta w kolejnych dokumentach oraz w strukturze drzewa (podobnej do drzewa folderów na dysku komputera).

Hierarchiczna baza danych zakłada podstawowe warunki integralności danych.

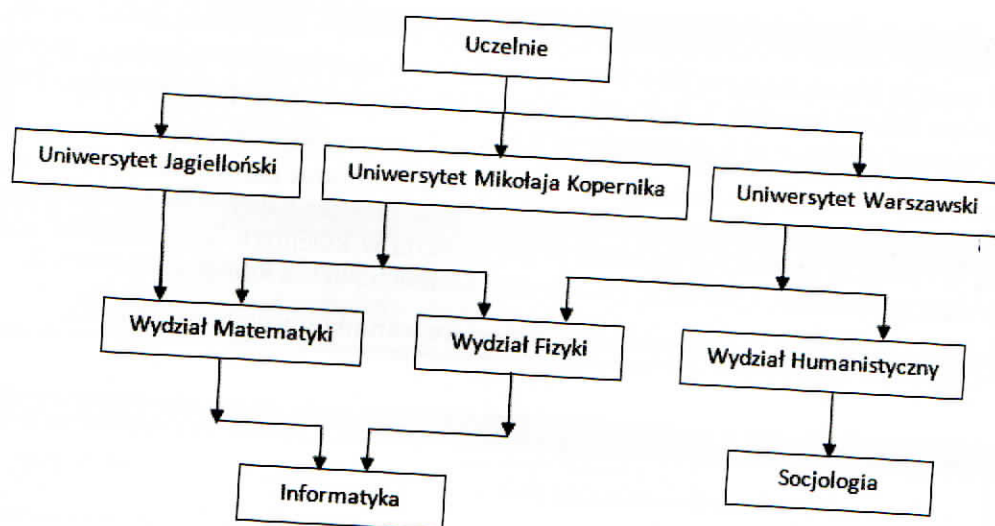


Rysunek 1.1. Model hierarchiczny

Każdy dokument (z wyjątkiem głównego — na górze) jest połączony z dokumentem nadrzędnym. Dokument podrzędny może zostać utworzony dopiero, gdy zostanie powiązany z dokumentem nadrzędnym. Kiedy zostanie usunięty dokument nadrzędny, automatycznie zostaną usunięte wszystkie dokumenty podrzędne.

1.2.2. Model sieciowy

W modelu sieciowym połączenia między dokumentami tworzą sieć (rysunek 1.2). Jest to zmodyfikowana wersja modelu hierarchicznego, w której dozwolone są połączenia na tym samym poziomie drzewa danych. Informacja jest zawarta w dokumentach oraz w przebiegu połączeń sieci.



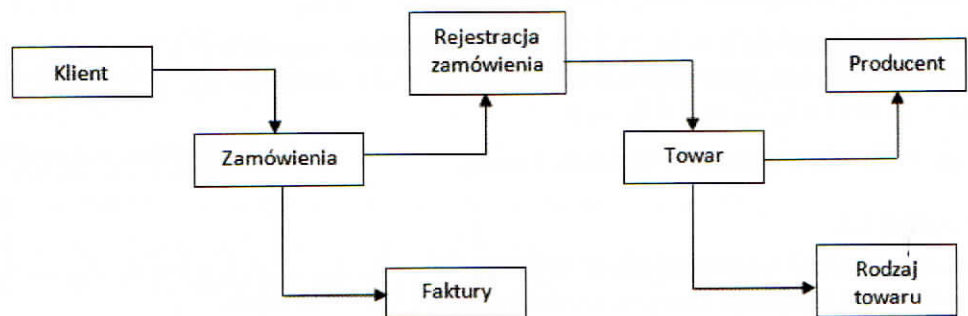
Rysunek 1.2. Model sieciowy

1.2.3. Model obiektowy

Model obiektowy łączy cechy programów komputerowych tworzonych w językach programowania obiektowego z cechami aplikacji bazodanowych. Dane są udostępniane w postaci obiektów. Ich stan i zachowanie są opisywane za pomocą narzędzi dostępnych w programowaniu obiektowym (własności, metody, klasy obiektów). Do przechowywania danych stosowane są obiekty. Obiekty obsługiwane za pomocą tych samych metod i własności są instancjami tej samej klasy.

1.2.4. Model relacyjny

Model relacyjny baz danych został oparty na matematycznym modelu organizacji danych i pojęciu relacji. W tym modelu dane są przedstawiane w postaci relacji reprezentowanych przez tabele. Relacje (tabele) składają się z rekordów (wierszy) o takiej samej strukturze. Tworzą one między sobą powiązania zwane relacjami (rysunek 1.3). Relacje (tabele) grupuje się w schematy bazy danych.



Rysunek 1.3. Model relacyjny

Ze względu na funkcjonalność model relacyjny jest najczęściej wykorzystywany przy projektowaniu baz danych.



1.3. Relacyjny model danych

1.3.1. Model relacyjny według E.F. Codd

Twórcą teorii relacyjnych baz danych jest Edgar Frank Codd. W 1970 roku opublikował on pracę pt. „Relacyjny model danych dla dużych banków danych”, która wprowadzała główne założenia dotyczące modelu relacyjnego baz danych. Później zostały uszczegółowione terminy *algebra relacji* i *rachunek relacyjny*.

Według podanych założeń określenie **relacyjna baza danych oznacza** bazę zbudowaną z relacji. Podstawowy obiekt takiej bazy danych, tabela, jest reprezentacją relacji w ujęciu pojęć matematycznych. Ale terminy *relacja* i *tabela* nie są jednoznaczne, ponieważ jedna relacja może być odwzorowana za pomocą wielu różnych tabel. Nie używa się również pojęć związanych z tabelą: *kolumna* i *wiersz*, lecz *atrybut* i *krotka*.

W modelu relacyjnym przyjmuje się, że:

- kolejność wierszy i kolumn w tabelach jest nieistotna,
- wiersze zawierające takie same dane są identyczne.

Każda tabela składa się z pewnej liczby wierszy i kolumn. Na przecięciu wiersza z kolumną znajduje się pole. Pole zawiera najmniejszą niepodzielną wartość, czyli taką część informacji, która nie może być dalej dzielona ze względu na spójność logiczną.

Definicja relacji według E.F. Codd

Relacja r to dowolny podzbiór iloczynu kartezjańskiego jednego lub więcej zbiorów:

$$r \subset D_1 \times D_2 \times \dots \times D_k$$

$$D_1 \times D_2 \times \dots \times D_k = \{(a_1, a_2, \dots, a_k) : a_i \in D_i, i \in \{1, 2, \dots, k\}\}$$

Schematem R relacji nazywamy zbiór atrybutów $\{A_1, \dots, A_n\}$.

Relacją r o schemacie $R = \{A_1, \dots, A_n\}$ nazywamy skończony zbiór $r = \{t_1, \dots, t_m\}$ odwzorowań $t_i: R \rightarrow D$, gdzie D jest równie sumie dziedzin atrybutów A_1, \dots, A_n takich, że $t_i(A_j) \in D_j$ dla $i = 1, \dots, m, j = 1, \dots, n$.

Każde takie odwzorowanie nazywamy krotką.

Przykład 1.1

Schematem relacji nazywamy zbiór atrybutów $R = \{A_1, A_2, \dots, A_n\}$, gdzie A_1, A_2, \dots, A_n są atrybutami reprezentowanymi w tabeli poprzez nazwy kolumn.

Przykładowym schematem relacji *towar* będzie zbiór:

towar{*Id_towaru*, *nazwa*, *cena*, *ilość*}.

Każdemu atrybutowi (A_1, A_2, \dots, A_n) przyporządkowana jest dziedzina (zakres dopuszczalnych wartości atrybutu) reprezentowana przez typ danych.

W tym przypadku dla poszczególnych kolumn zdefiniowano następujące typy danych:

Id_towaru *int*,

nazwa *varchar*(64),

cena *numeric*(7,2),

ilość *numeric*(6).

Dziedziną relacji o schemacie $R = \{A_1, A_2, \dots, A_n\}$ nazywamy sumę dziedzin wszystkich jej atrybutów:

$$D(R) = D(A_1) \cup D(A_2) \cup \dots \cup D(A_n).$$

Przykładową dziedziną relacji *towar* będzie dziedzina:

$$D(\text{towar}) = \text{int} \cup \text{varchar}(64) \cup \text{numeric}(7,2) \cup \text{numeric}(6).$$

Relacją o schemacie $R = \{A_1, A_2, \dots, A_n\}$ nazywamy skończony zbiór $r = \{t_1, t_2, \dots, t_m\}$ odwzorowań $t_i: R \rightarrow D(R)$ takich, że dla każdego j z zakresu $1 \leq j \leq n$ zachodzi zależność:

$$t_i(A_j) \in D(A_j).$$

Tak zdefiniowane pojedyncze odwzorowanie nosi nazwę krotki i odpowiada mu pojedynczy wiersz tabeli. Wartością krotki jest suma wartości poszczególnych atrybutów.

Przykładową relacją *towar* będzie:

komputer, 2300, 6

drukarka, 350, 13

monitor, 400, 7

klawiatura, 90, 16

Klucz

Kluczem schematu R relacji nazywamy taki zbiór atrybutów K tego schematu, że na podstawie wartości atrybutów z tego zbioru można jednoznacznie zidentyfikować każdą krotkę.

Właściwości klucza:

- Wartość klucza pozwala jednoznacznie identyfikować krotki.
- Dany schemat może posiadać kilka kluczy.
- Każdy nadzbiór klucza jest kluczem.
- Klucz, którego żaden podzbiór właściwy nie jest kluczem, nazywamy kluczem właściwym lub kandydującym.

Wśród kluczy wybiera się jeden, który staje się kluczem głównym (podstawowym).

Integralność danych

Integralność danych (spójność danych) oznacza poprawność struktury bazy danych (czyli zgodność ze schematem bazy danych) oraz poprawność przechowywanych w niej danych. W relacyjnym modelu danych występują następujące rodzaje więzów integralności:

- *Integralność encji* — każda relacja posiada klucz główny i żaden element klucza głównego nie może posiadać wartości pustej (*NULL*).
- *Integralność referencyjna* — każda wartość klucza obcego jest równa wartości klucza głównego określonej krotki lub wynosi *NULL*.
- *Więzy ogólne* — dodatkowe warunki dotyczące poprawności danych określone przez użytkowników lub administratorów baz danych.

Algebra relacji

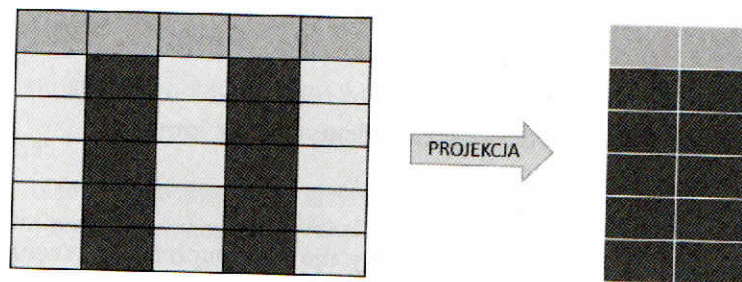
Algebra relacji zawiera zbiór operacji, które pozwalają na tworzenie potrzebnych relacji z relacji dostępnych w bazie danych.

Selekcja to wybór tych krotek relacji, które spełniają określone warunki (rysunek 1.4).



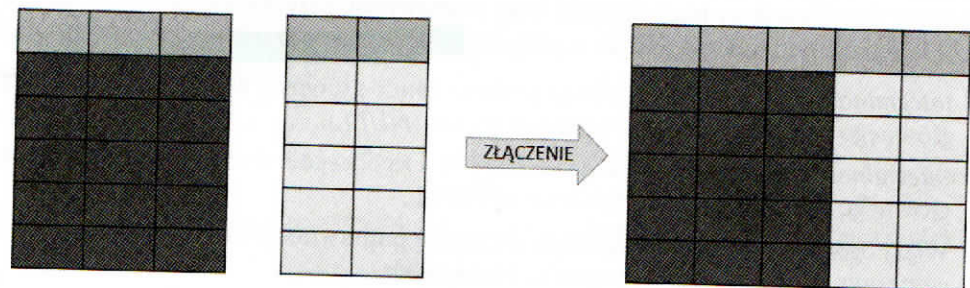
Rysunek 1.4. Wybór krotek z relacji (selekcja)

Projekcja to wybór z relacji określonych atrybutów (rysunek 1.5). Projekcja nazywana jest czasami selekcją pionową lub rzutem.



Rysunek 1.5. Wybór atrybutów z relacji (projekcja)

Złączenie to utworzenie z danych zawartych w dwóch relacjach jednej relacji (rysunek 1.6). Najczęściej łączone są relacje, które mają taki sam atrybut.



Rysunek 1.6. Złączenie dwóch relacji

Iloczynem kartezjańskim dwóch relacji — relacji r o schemacie $R = \{A_1, \dots, A_k\}$ oraz relacji s o schemacie $S = \{B_1, \dots, B_m\}$ takich, że $R \cap S = \emptyset$ — nazywamy relację $q = r \times s$ o schemacie $Q = R \cup S$ taką, że

$$t \in q \Leftrightarrow (\exists u \in r)(t \upharpoonright R = u) \wedge (\exists v \in s)(t \upharpoonright S = v)$$

$$t \in q \Leftrightarrow t \upharpoonright R \in r \wedge t \upharpoonright S \in s.$$

Iloczyn kartezjański dwóch relacji jest zbiorem wszystkich możliwych połączeń krotek obu relacji.

Wartość NULL

Jednym z kluczowych problemów relacyjnego modelu danych było podejście do brakującej informacji (na przykład nieznany numer telefonu lub numer domu). Ostatecznie E.F. Codd wprowadził do modelu relacyjnego dodatkową wartość: *NULL*. Wskutek tego nastąpiło rozszerzenie logiki dwuwartościowej operatorów porównania (*Tak*, *Nie*) do logiki trójwartościowej (na każde pytanie można odpowiedzieć: *Tak*, *Nie*, *Nieznane*).

Wartość *NULL* określa wartość atrybutu, która w danej chwili nie jest znana lub nie może zostać ustalona.

1.3.2. Relacyjny model baz danych

Istnieje wiele różnych podejść do relacyjnego modelu baz danych. Dwa główne to podejście formalne (opis relacyjnego modelu baz danych za pomocą reguł matematycznych) oraz podejście intuicyjne (podejście do relacyjnego modelu baz danych w sposób czysto użytkowy).

Cechy relacyjnego modelu baz danych:

- Podstawową formą przechowywania danych jest tabela (relacja).
- Użycie kombinacji „wartość klucza podstawowego, nazwa tabeli i nazwa kolumny” zapewnia dostęp do dowolnych danych.
- Manipulacje na danych są realizowane za pomocą selekcji, projekcji i złączenia.
- Musi być obsługiwana wartość *NULL* (wartość *NULL* przedstawia brakujące lub bezużyteczne informacje, na przykład nieznany numer telefonu).
- Integralność danych powinna być naturalną cechą projektu bazy danych.

Korzyści płynące z używania modelu relacyjnego:

- efektywność przechowywania danych,
- pewność integralności danych,
- możliwość rozbudowy bazy danych,
- możliwość łatwej zmiany w strukturze bazy danych,
- zwiększenie szybkości dostępu do danych.

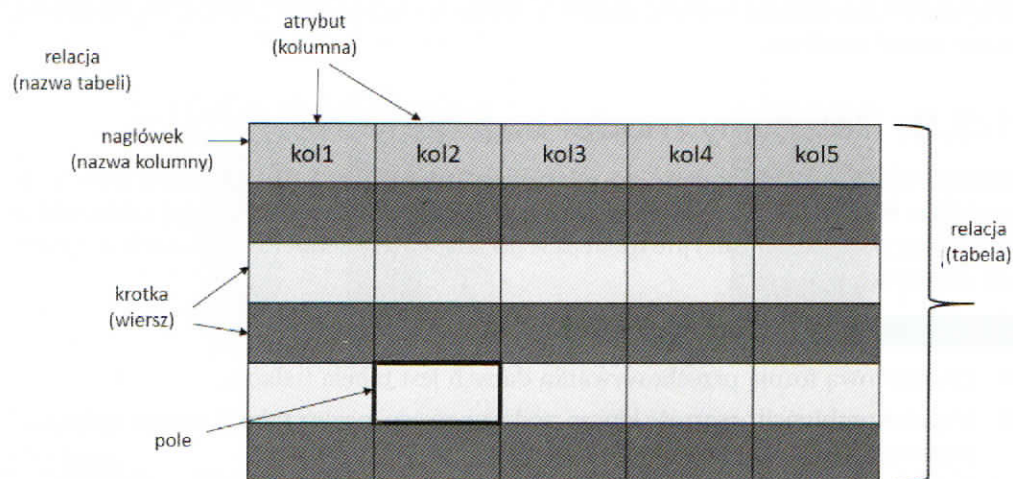
Opisując relacyjny model danych, **musimy określić:**

- struktury danych — czyli w jaki sposób i według jakich zasad zostanie zorganizowane przechowywanie danych oraz według jakich zasad będziemy je projektować,
- język manipulowania danymi — czyli w jaki sposób będą pobierane, zapisywane, modyfikowane i usuwane dane znajdujące się w bazie danych,
- integralność danych — czyli w jaki sposób zostanie zapewniona poprawność przechowywanych danych.

Tabele

W modelu relacyjnym baz danych wszystkie dane są przechowywane w dwuwymiarowych tabelach (relacjach). W skład bazy danych może wchodzić wiele relacji.

Tabela to zbiór powiązanych ze sobą danych. Jest to układ poziomych wierszy, nazywanych rekordami lub krotkami, i pionowych kolumn, nazywanych polami rekordu lub atrybutami. Tabela jest identyfikowana poprzez nazwę (rysunek 1.7).

**Rysunek 1.7.** Struktura tabeli

Przecięcie kolumny i wiersza tworzy pole. Nazwa kolumny jest jednocześnie nazwą pola. Pojęcia *pole* używa się również do określenia kolumny.

W modelu relacyjnym:

- każda tabela musi mieć jednoznaczną nazwę,
- każda kolumna w tabeli musi mieć jednoznaczną nazwę w obrębie tej tabeli,
- wszystkie wartości w kolumnie są tego samego typu,
- w tabeli nie mogą istnieć dwa identyczne wiersze,
- tabela nie może istnieć bez wierszy,
- przechowywane w tabeli dane oparte są na typach prostych (dane elementarne),
- kolejność wierszy i kolejność kolumn w tabeli nie ma żadnego znaczenia.

Klucz podstawowy

Tabela musi mieć zestaw pól, które będą jednoznacznie identyfikować dany rekord. W najlepszym przypadku w tabeli może istnieć jedno pole spełniające to założenie. To klucz podstawowy, nazywany również kluczem głównym lub pierwotnym.

WSKAZÓWKA

Klucz podstawowy to minimalna kombinacja pól identyfikująca każdy rekord w tabeli w sposób jednoznaczny.

Klucz podstawowy pozwala w sposób efektywny przeszukiwać i odczytywać dane w bazie oraz łączyć dane zapisane w różnych tabelach.

Klucz podstawowy nie może zawierać powtarzających się danych oraz nie może być pusty. Oznacza to, że w tabeli musi się znaleźć jedno lub kilka pól, które pozwolą odróżniać dane zapisane w jednym rekordzie od danych zapisanych w innym rekordzie.

Jeżeli tabela zawiera dane osobowe, takim kluczem może być kolumna z nazwiskiem. Każdą osobę zapisaną w tabeli rozpoznamy po nazwisku i odczytamy jej dane z właściwego rekordu. Co jednak zrobić, jeżeli w tabeli zapisaliśmy informacje o kilku osobach, które mają takie samo nazwisko?

W takim wypadku kluczem podstawowym może być kombinacja pól, czyli do pola *Nazwisko* możemy dodać pole *Imię*. Teraz każdą osobę w tabeli znajdziemy, podając jej nazwisko i imię. A co zrobić, jeśli w tabeli są umieszczone informacje o dwóch osobach, które mają identyczne nazwisko i imię? Można do klucza dodać kolejne pole, na przykład *Data urodzenia*.

W celu uniknięcia dodawania kolejnych pól do klucza bardzo często funkcję klucza podstawowego pełni klucz sztuczny. Najprostszym sposobem utworzenia klucza sztucznego jest dodanie do tabeli dodatkowego pola i umieszczenie w nim kolejnych numerów.

WSKAZÓWKA

Klucz sztuczny to pole zawierające unikatowy numer identyfikacyjny nadany w sposób sztuczny każdemu obiektowi umieszczonemu w tabeli.

Jeżeli w tabeli zostało utworzone takie pole, staje się ono automatycznie kluczem podstawowym, ponieważ spełnia wszystkie wymagania dotyczące tego klucza.

Klucz obcy to jedno pole lub więcej pól tabeli (kolumn), które odwołują się do pola lub pól klucza podstawowego w innej tabeli. Klucz obcy pokazuje, w jaki sposób tabele są powiązane. Jest niezbędny do zdefiniowania połączenia między tabelami.

Integralność danych

W modelu relacyjnym baz danych integralność danych zachowana jest poprzez:

- klucz podstawowy,
- klucz obcy,
- zawężenie dziedziny,
- unikatowość wartości,
- możliwość nadawania wartości pustych.

Połączenia

Projektując bazę danych, dzielimy dane na wiele tabel tematycznych, tak aby każda informacja została zapisana tylko raz. Aby zestawić razem dane zapisane w różnych tabelach, tworzy się między nimi połączenia. W wyniku powstaje nowa tabela (relacja) zawierająca połączone wiersze tabel, dla których wartości klucza obcego i klucza

podstawowego są takie same. Definiowanie logicznego połączenia między tabelami bazy danych nazywane jest również relacją.

Typy połączeń (związków)

W rzeczywistości (w życiu codziennym) występują trzy typy połączeń (związków).

Związek „jeden do jednego” (1:1)

W związku „jeden do jednego” każdemu rekordowi z pierwszej tabeli może odpowiadać tylko jeden rekord z drugiej tabeli i każdemu rekordowi z drugiej tabeli może odpowiadać tylko jeden rekord z pierwszej tabeli. Jest to nietypowy rodzaj związku, ponieważ najczęściej informacje powiązane w ten sposób są przechowywane w jednej tabeli. Opisywanego związku używa się w celu odizolowania części tabeli ze względu na bezpieczeństwo danych lub w celu podzielenia danych na podzbiory.

Związek „jeden do wielu” (1:n lub 1:∞)

W związku „jeden do wielu” każdemu rekordowi z pierwszej tabeli może odpowiadać wiele rekordów z drugiej tabeli, a każdemu rekordowi z drugiej tabeli może odpowiadać najwyżej jeden rekord z pierwszej tabeli. Jest to typ związku najczęściej występujący w relacyjnych bazach danych.

Związek „wiele do wielu” (m:n)

W związku „wiele do wielu” każdemu rekordowi z pierwszej tabeli może odpowiadać wiele rekordów z drugiej tabeli i każdemu rekordowi z drugiej tabeli może odpowiadać wiele rekordów z pierwszej tabeli.

Związek „wiele do wielu” zawsze jest realizowany jako dwa związki „jeden do wielu”.

1.4. Projektowanie bazy danych

W bazie danych przechowujemy tylko niektóre informacje o świecie rzeczywistym. Wybór właściwych wycinków rzeczywistości i dotyczących ich danych jest bardzo istotny — od niego zależy prawidłowe działanie bazy. Aby ten wybór był właściwy, należy wskazać informacje, które powinny być przechowywane w bazie danych, oraz określić ich strukturę.

1.4.1. Zasady projektowania bazy danych

Cały proces projektowania bazy danych możemy podzielić na kilka etapów:

- planowanie bazy danych,
- tworzenie modelu koncepcyjnego (diagramy ERD),
- transformacja modelu koncepcyjnego na model relacyjny,
- proces normalizacji bazy danych,
- wybór struktur i określenie zasad dostępu do bazy danych.