Mam nadzieję, że do tej pory przyzwyczaiłeś się do tabelarycznego układu danych i poznałeś sposoby odczytywania i modyfikowania tak zapisanych danych. W tym odcinku poznasz nieco teorii relacyjnych baz danych, w tym algorytmom przekształcania relacji do kolejnych postaci normalnych, czyli dowiesz się, jak zaprojektować podstawowe obiekty baz danych — tabele.

Wprowadzenie

Celem procesu projektowania bazy danych jest utworzenie poprawnego i spełniającego wymagania użytkowników logicznego schematu bazy danych. Ponieważ cały proces jest dosyć skomplikowany, w przypadku rozbudowanych baz danych dzieli się go na kilka etapów:

1. Przygotowanie diagramu związków E/R.
2. Normalizacja projektu.
3. Implementacja zasad wymuszających integralność danych.

Skoro baza danych to zbiór uporządkowanych danych, to praca projektanta sprowadza się do opracowania struktury, w której będą przechowywane dane, oraz wyboru informacji, jakie powinny znaleźć się w bazie danych.

W pierwszej kolejności należy określić schemat (strukturę) bazy danych i dopiero wtedy, dysponując gotowym schematem, wybrać te informacje, jakie będą przechowywane w bazie danych. Tworząc schemat, należy kierować się ogólną regułą, na podstawie której:

1. W otaczającym nas świecie można wyróżnić mniej lub bardziej trwałe, ale będące logicznymi całościami obiekty różnych typów.
2. Obiekty poszczególnych typów mogą być określone za pomocą właściwych im cech (atrybutów, metod i zdarzeń).

Wybór typów obiektów oraz określenie, jakie informacje powinny być przechowywane w bazie, jest podstawą diagramu związku E/R (encja/relacja). Zanim opiszę tworzenie takiego modelu, przedstawię podstawy teorii relacyjnych baz danych.

Model relacyjnych baz danych

Termin relacyjna baza danych oznacza bazę zbudowaną z relacji. Podstawowy obiekt takiej bazy danych, tabela, jest konkretną reprezentacją relacji — technicznego pojęcia matematyki. Wynika z tego, że oba terminy nie są jednoznaczne, a jedna relacja może być odwzorowana za pomocą wielu różnych tabel.

Relacyjny model baz danych został stworzony przez E.F. Codda w 1970 roku i przedstawiony w pracy Relacyjny model danych dla dużych banków danych. Nie używa się tam pojęć tabela, kolumna i wiersz, lecz relacja, atrybut i krotka.

Każda tabela składa się z pewnej liczby wierszy i kolumn. Na przecięciu wiersza z kolumną znajduje się pole. W modelu relacyjnym przyjmuje się, że:

1.

kolejność wierszy i kolumn w tabelach jest nieistotna,
2.

wiersze zawierające takie same dane są identyczne.

Natomiast w tabeli przedstawiającej konkretny przypadek relacji identyczne dane (wartości pól) będą przechowywane w różnych wierszach. Pole zawiera najmniejszą niepodzielną wartość, czyli taką część informacji, która nie może być dalej dzielona ze względu na spójność logiczną. **Podstawowe definicje**

Przed podaniem definicji relacji — podstawowego terminu opisywanej teorii — musimy poznać definicje schematu relacji i dziedziny relacji. **Definicja 1. Schemat relacji**

Schematem relacji nazywamy zbiór R = {A_1, A_2, …, A_n}, gdzie: A_1, A_2, …, A_n są atrybutami, reprezentowanymi w tabeli poprzez nazwy kolumn.

Przykładowym schematem relacji item będzie zbiór: item {item_id, description, cost_price, sell_price}. Każdemu atrybutowi (A_1, A_2, …, A_n) przyporządkowana jest dziedzina (zakres dopuszczalnych wartości atrybutu) reprezentowana przez typ danych. W tym przypadku dla poszczególnych kolumn zdefiniowano następujące typy danych:

item_id

int,

description

varchar(64)

cost_price

numeric(7,2)

sell_price

numeric(7,2)

czyli:
1.

Wartości atrybutu item_id będą liczbami całkowitymi z zakresu od –2 147 483 648 do 2 147 483 647 lub od 0 do 4 294 967 295 (w przypadku liczb bez znaku).
2.

Wartości atrybutu description będą ciągami 64 znaków.
3.

Wartości atrybutu cost_price będą liczbami dziesiętnymi o precyzji 7 i skali 2 (precyzja oznacza liczbę cyfr przed przecinkiem, skala — liczbę miejsc po przecinku).

4.

Wartości atrybutu sell_price będą liczbami dziesiętnymi o precyzji 7 i skali 2.

Definicja 2. Dziedzina relacji

Dziedziną relacji o schemacie $R = \{A_1, A_2, \ldots, A_n\}$ nazywamy sumę dziedzin wszystkich jej atrybutów $Dom(R)=Dom(A_1) \cup Dom(A_2) \cup \ldots \cup Dom(A_n)$.

Przykładową dziedziną relacji item będzie dziedzina Dom(item) = int ∪ varchar (64) ∪ numeric(7,2) ∪ numeric(7,2).

Teraz można już podać definicję podstawowego obiektu relacyjnych baz danych — relacji. **Definicja 3. Relacja**

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

Tak zdefiniowane pojedyncze odwzorowanie nosi nazwę krotki i odpowiada mu pojedynczy wiersz tabeli. Wartością krotki jest suma wartości poszczególnych atrybutów. Na przykład wartością pierwszej krotki opisywanej tabeli jest zbiór: 1∪ ' Wood Puzzle' ∪ 15.23 ∪ 21.95. Krotka może zostać ograniczona do wartości wybranych atrybutów. **Lemat 1. Ograniczenie krotki**

Ograniczeniem krotki t relacji r o schemacie R do zbioru atrybutów X R nazywamy odwzorowanie będące ograniczeniem krotki t do zbioru atrybutów X $t_{|X}$: $X \rightarrow Dom(X)$.

Przykład. Wartością ograniczenia X = { item_id, description } przykładowej krotki będzie zbiór wartości { 1∪ ' Wood Puzzle' }.

Podstawowe zasady implementacji modelu relacyjnych baz danych można podzielić na trzy grupy:

Zasady dotyczące struktury danych.

Zasady dotyczące przetwarzania danych.

Zasady dotyczące integralności danych.

Zasady dotyczące struktury danych

W modelu relacyjnych baz danych informacja o poszczególnych obiektach zapisana jest w tabelach. Jest to model abstrakcyjny, zawsze obsługiwany jednakowo i niezwiązany ze sposobem przechowywania danych. Dzięki temu możliwe jest oddzielenie danych od aplikacji klienckiej (interfejsu użytkownika) i platformy sprzętowej. Serwer bazodanowy zarządza rozmieszczeniem danych w plikach znajdujących się na dysku bądź w pamięci podręcznej oraz metodami dostępu do tych danych.

Cechą charakterystyczną modelu jest wymóg przechowywania w bazie danych tylko konkretnych wartości. W relacyjnych bazach danych nie możemy posługiwać się wskaźnikami do danych.

Twórca relacyjnego modelu baz danych, E.F. Codd, przedstawił zbiór dwunastu postulatów, które powinny być uwzględnione przez projektantów systemów zarządzania relacyjnymi bazami danych. Zamieszczone niżej postulaty dotyczą struktury danych; ich znajomość może okazać się przydatna podczas projektowania baz.

- Postulat informacyjny. Na poziomie logicznym dane reprezentowane są wyłącznie za pomocą tabel wartości.

- Postulat dostępu. Do każdej pojedynczej danej jest dostęp za pomocą nazwy tabeli, kolumn i wartości kluczy głównych.
- Postulat fizycznej niezależności danych. Zmiany w sposobie przechowywania danych i dostępu do nich nie wpływają na aplikację kliencką.
- Postulat logicznej niezależność danych. Zmiany w tabelach, zachowujące informację i dopuszczalne semantycznie, nie mają wpływu na aplikację kliencką.
- Postulat niezależności dystrybucyjnej. System i jego język umożliwiają dostęp do danych zapisanych w różnych miejscach, np. na wielu komputerach w sieci.
- Postulat zabezpieczania przed operacjami na niższym poziomie abstrakcji. Jeśli system zarządzania bazą danych umożliwia bezpośrednie operacje na niższych poziomach abstrakcji, nie mogą one naruszać reguł relacyjnego modelu baz danych, w szczególności nie mogą pomijać ograniczeń określonych przez więzy spójności.

Zasady dotyczące przetwarzania danych

Dane przechowywane w bazie danych nie są niezmienne. Wręcz przeciwnie — aby baza danych była przydatna, przechowywane w niej informacje muszą odpowiadać stanowi faktycznemu, a więc muszą być cały czas aktualizowane. Operacje modyfikowania danych nie mogą jednak naruszać struktury danych. Wynika z tego, że przekształcenie danych dokonywane bądź to w celu ich modyfikacji, bądź pobrania danych, musi przebiegać z zachowaniem wewnętrznej logiki (struktury) powiązań istniejących pomiędzy danymi.

Pierwszy typ przekształceń, które można wykonywać na wartościach atrybutów, wynika z definicji relacji jako zbioru. Operatorom algebry zbiorów, czyli $\cup$ $\cap$ —, odpowiadają operatory algebry relacji UNION (suma relacji), INTERSECT (część wspólna relacji, przecięcie, przekrój relacji) i EXCEPT (dopełnienie relacji, różnica relacji).

MySQL pozwala na łączenie wyników dwóch zapytań jedynie za pomocą operatora UNION. Aby obliczyć część wspólną lub dopełnienie wyników zapytań, należy posłużyć się podzapytaniem powiązanym.

Na sumę dwóch relacji r $\cup$ s składają się wszystkie elementy relacji r i wszystkie elementy relacji s (rysunek 9.1).

Rysunek 9.1. Suma relacji

Wynikiem przecięcia relacji r $\cap$ s jest relacja składająca się z elementów wspólnych relacji r i s (rysunek 9.2).

Rysunek 9.2. Przecięcie relacji

Wynikiem odjęcia relacji r - s jest relacja powstała na skutek usunięcia z relacji r wszystkich elementów wchodzących w skład relacji s (rysunek 9.3).

Rysunek 9.3. Dopełnienie relacji

Oprócz operatorów teoriomnogościowych dla operacji na relacjach zdefiniowane są następujące operatory: selekcji, projekcji, iloczynu kartezjańskiego i złączenia naturalnego. Dwa pierwsze dotyczą wyników zapytania odwołującego się do pojedynczej relacji, dwa kolejne — wyników odwołania się do wielu relacji. **Definicja 4. Selekcja relacji**

Selekcją relacji r o schemacie R = {A_1, A_2, …, A_n} nazywamy:

1.

Zbiór tych krotek relacji r, w których wartość atrybutu A wynosi a, przy założeniu, że $A \in R$ i $a \in Dom(A)$, co zapisujemy $\sigma_{A=a}(r) = \{t \in r \mid t(A) = a\}$.

2.

Dla dowolnego warunku logicznego F — zbiór krotek relacji r spełniających warunek F, co zapisujemy $\sigma_F(r) = \{t \in r \mid t \text{ spełnia warunek F}\}$.

Przykład:

$\sigma_{item\_id=1}(item) = \{1 \cup 'Wood\ Puzzle' \cup 15.23 \cup 21.95\}$

Dla operatora selekcji prawdziwe są następujące twierdzenia: **Twierdzenie 1. Selekcja sumy**

σ_F (r ∪ X Y — projekcja wyniku szerszej projekcji równa jest wynikowi bardziej restrykcyjnej projekcji.
Twierdzenie 5. Projekcja sumy

$\pi_X(r \cup s) = \pi_X(r) \cup \pi_X(s)$ — projekcja sumy relacji równa jest sumie projekcji. **Twierdzenie 6. Selekcja projekcji**

$\sigma_F(\pi_X(r)) = \pi_X(\sigma_F(r))$ — selekcja projekcji równa jest projekcji selekcji.

Na podstawie twierdzeń 4. – 6. możemy w dowolny sposób zmieniać kolejność operacji sumy, selekcji i projekcji. Zmiana kolejności operacji nie wpłynie na ich wynik. **Definicja 6. Iloczyn kartezjański**

Iloczynem kartezjańskim dwóch relacji: r o schemacie $R = \{A_1, A_2, \ldots, A_n\}$ i s o schemacie $S = \{B_1, B_2, \ldots, B_m\}$, przy założeniu, że $R \cap S = \emptyset$ nazywamy relację $q = r \otimes s$ o schemacie $Q = \{A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m\}$, składającą się ze wszystkich krotek $t \in q$, dla których istnieją krotki $u \in r$ i $v \in s$ takie, że: $t(A_i) = u(A_i)$ dla $1 <= i <= n$ oraz $t(B_i) = v(B_i)$ dla $1 <= i <= m$.

Iloczyn kartezjański dwóch relacji jest zbiorem wszystkich możliwych połączeń krotek obu relacji. Jeżeli schematy relacji nie są rozłączne (istnieje niepusty zbiór powstały w wyniku przecięcia obu relacji), najpierw zmienia się nazwy atrybutów jednej z relacji, a następnie stosuje się powyższą definicję.

Dla operatora iloczynu kartezjańskiego prawdziwe są następujące twierdzenia: **Twierdzenie 7. Iloczyn sumy**

$(r \cup s) \otimes q = (r \otimes q) \cup (s \otimes q)$ — iloczyn kartezjański sumy równy jest sumie iloczynów kartezjańskich.
Twierdzenie 8. Selekcja iloczynu

$\sigma_{F \wedge G}(r \otimes s) = \sigma_F(r) \otimes \sigma_G(s)$ przy założeniu, że F(R), G(S) — selekcja iloczynu kartezjańskiego równa jest iloczynowi kartezjańskiemu selekcji. **Twierdzenie 9. Projekcja iloczynu**

$\pi_{X \cup Y}(r \otimes s) = \pi_X(r) \otimes \pi_Y(s)$ przy założeniu, że X R i Y S — projekcja iloczynu kartezjańskiego równa jest iloczynowi kartezjańskiemu projekcji.

Jak już miałeś okazję się przekonać, wynikiem iloczynu kartezjańskiego jest zbiór wielu krotek, z których nie wszystkie zawierają sensowne dane. Dlatego w produkcyjnych bazach danych bardzo rzadko wykorzystuje się tę metodę łączenia relacji. **Definicja 7. Złączenie naturalne**

Złączeniem naturalnym relacji r o schemacie $R = \{A_1, A_2, \ldots, A_n\}$ i s o schemacie $S = \{B_1, B_2, \ldots, B_m\}$ jest relacja q o schemacie $Q = R \cup S$ taka, że $q = \{t: \exists u \in r, i v \in s$ takie, że $u_{|R} = t$ i $v_{|S} = t\}$. Tak więc złączeniem naturalnym relacji jest zbiór wszystkich możliwych połączeń krotek relacji, przy których ich wspólne atrybuty mają takie same wartości.

Wynikiem złączenia naturalnego dwóch relacji jest zbiór powiązanych ze sobą krotek obu relacji.

Dla operatora złączenia naturalnego prawdziwe są następujące twierdzenia: **Twierdzenie 10. Złączenie relacji z samą sobą**

$q \oplus q = q$ — wynikiem złączenia relacji z samą sobą jest dana relacja. **Twierdzenie 11. Naprzemienność złączenia**

$q \oplus r = r \oplus q$ — zmiana kolejności łączonych relacji nie wpływa na wynik złączenia. **Twierdzenie 12. Przechodniość złączenia**

$(q \oplus r) \oplus s = q \oplus (r \oplus s)$ — zmiana kolejności operacji złączenia naturalnego kilku relacji nie wpływa na wynik złączenia. **Twierdzenie 13. Projekcja złączenia**

$\pi_R(r \oplus s) \subseteq r$ oraz π