

Wykorzystanie nowych technologii w dystrybucji usług hotelarskich

Część 1

Literatura:

Jennifer Widom, Jeffrey Ullman Podstawowy kurs systemów baz danych, Wydanie III Wydawnictwo: Helion

Bazy danych

System relacyjnych baz danych charakteryzuje się następującymi właściwościami:

- Wszystkie dane są reprezentowane koncepcyjnie jako zbiór wartości uporządkowanych w formie wierszy i kolumn zwanych relacją
- Wszystkie wartości są skalarne. Oznacza to, że dowolna pozycja relacji na przecięciu kolumny i wiersza zawiera zawsze jedną i tylko jedną wartość
- Wszystkie operacje są wykonywane na całej relacji, a ich wynikiem jest również relacja. Taka właściwość operacji nazywa się domknięciem.

Zależności funkcyjne i wielowartościowe

Postać normalna stanowi sposób klasyfikowania tabel w oparciu o występujące w niej zależności funkcyjne (w skrócie ZF). Zależność funkcyjna oznacza, że znając wartość jednego atrybutu, zawsze możemy określić wartość innego. Symbolem stosowanym w teorii relacji jest strzałka umieszczona pomiędzy dwoma atrybutami, na przykład $A \rightarrow B$, co czyta się następująco: "A określa B". Gdy znamy numer naszego pracownika, możemy określić jego nazwisko; jeśli znamy numer części, możemy określić jej masę i barwę itd.

Zależność wielowartościowa (ZWW) oznacza, że znając wartość jednego atrybutu, możemy zawsze określić wartości zbioru innego atrybutu. W teorii relacji używa się symbolu zależności wielowartościowej w postaci strzałki z dwoma grotami, umieszczonej pomiędzy tymi dwoma atrybutami, na przykład: $A \twoheadrightarrow B$, co można przeczytać jako: "A określa wiele B". Gdy znamy nazwisko nauczycielki, możemy określić listę jej studentów; jeśli znamy numer części, możemy określić numery jej elementów składowych itd.

Pierwsza postać normalna (1PN)

Pierwsza postać normalna (1PN) oznacza, że tabela nie zawiera powtarzających się grup informacji. To znaczy, że każda kolumna jest wartością skalarną (atomową), a nie macierzą lub listą czy też czymkolwiek, co posiada własną strukturę. W przypadku języka SQL, nie jest możliwe, aby tabela nie odpowiadała pierwszej postaci normalnej.

Druga postać normalna (2PN)

Tabela odpowiada drugiej postaci normalnej (2PN), jeśli nie posiada żadnych zależności kluczy częściowych. To znaczy, jeśli X i Y są kolumnami, a X jest kluczem, wtedy dla dowolnego Z, będącego podzbiorem właściwym X, nie występuje sytuacja, w której $Z \rightarrow Y$. Inaczej mówiąc, tabela jest 1PN oraz posiada klucz określający wszystkie niekluczowe atrybuty w tej tabeli.

Trzecia postać normalna (3PN)

Powyższe problemy można zaadresować do innej postaci normalnej. Tabela jest w trzeciej postaci normalnej (3PN), jeśli dla wszystkich $x \rightarrow y$, gdzie X i Y są kolumnami tabeli, X jest kluczem lub Y jest częścią *klucza kandydującego*. (Kluczem kandydującym jest unikalny zbiór kolumn identyfikujący każdy wiersz tabeli; nie można usunąć kolumny z klucza kandydującego bez naruszenia jego unikalności). Implikuje to stwierdzenie, że ta tabela odpowiada 2PN, ponieważ zależność klucza częściowego jest typu przechodniego. Inaczej mówiąc, wszystkie niekluczowe kolumny są określane kluczem, całym kluczem i tylko kluczem. Inaczej mówiąc tabela jest w drugiej postaci normalnej i wszystkie atrybuty niekluczowe są wzajemnie nie zależne.

Pojęcia związane z wartościami

Dane

- Dane są wartościami przechowywanymi w bazie danych

Informacje

- Informacje to dane przetworzone w sposób, który uwidacznia ich znaczenie

Null

O wartości mówimy null, gdy jest nieznana lub jej nie ma (wartość zero w danych numerycznych nie oznacza zera)

Pojęcia związane ze strukturami

- Tabele
- Pole (atrybut)
- Rekord (krotka)
- Perspektywy
- Klucze (podstawowy, obcy)
- Indeksy (jest strukturą usprawniającą przetwarzanie danych i nie mają nic wspólnego logiczną strukturą bazy danych)

Pojęcia związane z relacjami

- Relacje
Powiązanie pomiędzy parą tabel nosi nazwę relacji
- Typy relacji
Jeden do jednego
Jeden do wielu
Wiele do wielu (studenci wykłady) operowanie takimi relacjami jest trudne

Typy uczestnictwa

- Istnienie tabel w relacji
- Obowiązkowy i opcjonalny

Stopień uczestnictwa

- Minimalna i maksymalna liczba rekordów które można połączyć z pojedynczym rekordem drugiej tabeli
- Pośrednik obsługuje przynajmniej jednego klienta, ale nie więcej niż 8 zapisujemy to (1,8)

Pojęcia związane z integralnością

Atrybuty pól

- Każde pole posiada trzy rodzaje atrybutów (ogólne, fizyczne i logiczne)
- Ogólne – opis jego funkcji i nazwa tabeli, której jest częścią
- Fizyczne – jak zbudowane, typ przechowywanych danych, długość, format
- Logiczne jakie wartości mogą się w nim znaleźć

Integralność Danych

To poprawność, spójność, dokładność

- Integralność na poziomie
Tabeli zawartość pola identyfikującego rekord jest unikalna i nie jest pusta
Pól zawartość jest logiczna, i są identycznie zdefiniowane
Relacji poprawnie zdefiniowane i zsynchronizowane
Reguły integralności ograniczenia na sposób użytkowanie

Proces projektowania

- Formułowanie definicji celu i założeń wstępnych
- Analiza istniejącej bazy danych
- Tworzenie struktur danych
- Definiowanie relacji
- Wprowadzenie reguł integralności
- Definiowanie perspektyw
- Kontrola integralności danych

Analiza istniejącej bazy danych

- Poznawanie istniejącej bazy danych
- Papierowe bazy danych

- Spadkowe bazy danych
- Prowadzenie analizy
- Analiza sposobu gromadzenia danych
- Analiza sposobu prezentowania informacji
- Prowadzenie wywiadów
- Przegląd rodzajów i sposobów wykorzystania danych
- Przegląd próbek
- Przegląd wymagań informacyjnych
- Prowadzenie wywiadów kierownictwem
- Przegląd przyszłych wymagań informacyjnych
- Przygotowanie listy pól wyliczalnych

Zasady tworzenia nazw pól

- Nazwy przejrzyste i zrozumiałe dla każdego pracownika organizacji
- Nazwa ma opisywać bez wątpliwości daną cechę
- Nazwy powinny być zwięzłe
- Nie stosować akronimów skrótów
- Nie używać słów gmatwających sens nazwy pola
- Nie używać nazw określających więcej niż jedną cechę np. fax/tel
- Stosować liczbę pojedynczą

Pole doskonale - cechy

- Reprezentuje cechę tematu tabeli
- Zawiera pojedynczą wartość
- Nie można go rozłożyć na części składowe
- Nie zawiera wartości będących wynikiem połączenia lub operacji matematycznych na wartościach innych pól
- Jest unikatowe w zakresie całej struktury bazy danych
- Zachowuje identyczne atrybuty we wszystkich tabelach, w których występuje.

Zwielokrotnione dane i pola

- Zwielokrotnione dane to takie, które w strukturze tabel pojawiają się w bazie więcej niż jeden raz
- Zwielokrotnione pole to takie, które występuje w dwóch lub większej liczbie tabel.

Tabela doskonała – cechy

- Reprezentuje pojedynczy temat, którym może być obiekt lub zdarzenie
- Posiada klucz podstawowy
- Nie zawiera pól segmentowych
- Nie zawiera pól wielowartościowych
- Nie zawiera pól wyliczalnych
- Nie zawiera zbędnych pól zwielokrotnionych

Klucze

- Umożliwiają identyfikację każdego rekordu w tabeli
- Umożliwiają wprowadzenie i egzekwowanie różnych rodzajów integralności danych
- Umożliwiają definicje relacji
- Istnieją trzy typy kluczy: kandydujący, podstawowy, obcy
- Klucz podstawowy stanowi identyfikator danej tabeli w strukturze bazy danych.

Cechy klucza kandydującego

- Musi jednoznacznie identyfikować każdy rekord w tabeli
- Musi zawierać unikatowe wartości
- Nie może zawierać wartości zerowej
- Nie może być polem segmentowym
- Składa się z minimalnej liczby pól niezbędnych do uzyskania niepowtarzalności
- Jego wartość nie może być opcjonalna
- Musi bezpośrednio określać wartość każdego pola w tabeli

- Jego wartość powinno modyfikować się jedynie w wyjątkowych przypadkach.

Cechy klucza podstawowego jak klucza kandydującego

Zasady definiowania klucza podstawowego

- Każda tabela powinna zawierać dokładnie jeden klucz podstawowy
- Wszystkie klucze podstawowe w bazie danych muszą się od siebie różnić

Integralność na poziomie tabel

- W tabeli nie istnieją zwielokrotnione rekordy
- Każdy rekord w tabeli jest identyfikowany przez określoną wartość klucza podstawowego
- Żadna wartość klucza podstawowego nie powtarza się
- Klucz podstawowy nigdy nie przyjmuje wartości zerowej

Integralność na poziomie pól

- Tożsamość i cel każdego pola są oczywiste, każde pole występuje we właściwych tabelach
- Definicje pól są spójne w całej strukturze bazy
- Wartości pól są poprawne i logiczne
- Rodzaje dozwolonych operacji na wartościach danego pola są jasno określone.

Relacje

- Dwie tabele są ze sobą w relacji, jeśli wiążą je wartości klucza podstawowego i klucza obcego lub specjalna tabela łącząca.
- Relacje stanowią sposób na łączenie tabel przechowujących powiązane ze sobą dane
- Pomagają w ulepszeniu struktur tabel i minimalizują ilość nadmiernych danych
- Umożliwiają jednoczesne odczytywanie informacji z wielu różnych tabel
- Poprawnie zdefiniowane wprowadzają do bazy danych integralność na poziomie relacji

Relacja jeden do jednego

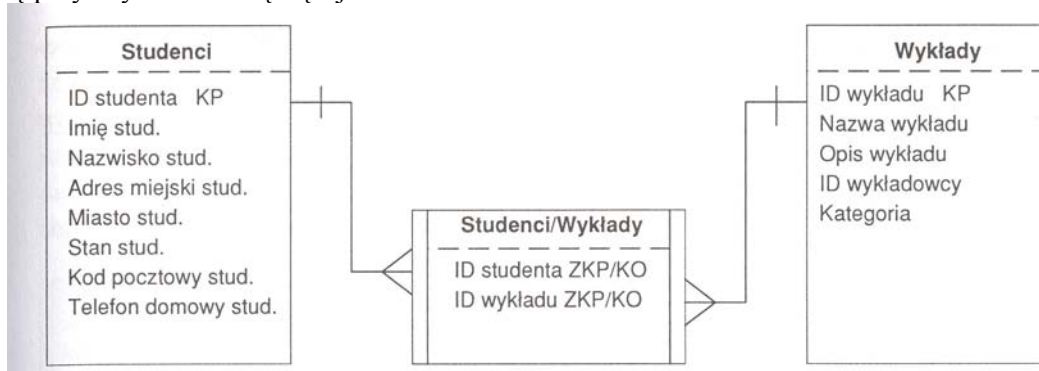
Tabela główna i podporządkowana

Relacja jeden do wielu

Należy dołączyć klucz podstawowy po stronie jeden do tabeli po stronie wiele

Relacja wiele do wielu

Definiuje się przy użyciu tabeli łączącej



Cechy klucza obcego

- Ma tę samą nazwę, co klucz podstawowy tabeli, z której został skopiowany
- Jego atrybuty stanowią replikę specyfikacji pola kluczowego tabeli, z której pochodzi.
- Przybiera tylko istniejące wartości klucza podstawowego, na którym jest oparty

Reguły usuwania

- Restrykcyjna żaden rekord tabeli podstawowej nie zostanie skasowany, jeśli tabela podporządkowana zawiera rekordy
- Kaskadowa kasowany rekord z tabel podstawowej powoduje skasowanie wszystkich skojarzonych z nim rekordów.

Integralność na poziomie relacji

- Efektywne połączenia między każdymi dwiema tabelami uczestniczącymi w relacji
- Umożliwione wprowadzanie nowych rekordów do obu tabel
- Zabezpieczenie przed negatywnymi skutkami kasowania rekordów
- Ustanowiony logiczny limit rekordów, które mogą brać udział w relacji z pojedynczym rekordem.

Tabele walidacji

Ma za zadanie przechowywanie danych zapewniających integralność danych. Dane w takiej tabeli rzadko ulegają zmianie.

Perspektywy

Perspektywa jest tabelą wirtualną składającą się z pól pochodzących z różnych tabel jak i perspektyw.

Elementy języka SQL

Systemy bazodanowe używające SQL

DB2	Firebird
Ingres	Informix
InterBase SQL	MaxDB znana jako SAP DB
Microsoft Access	Microsoft Jet
Microsoft SQL Server	Mimer SQL
MySQL	mSQL
Neteeza	OpenLink Virtuoso
Oracle	Oracle Rdb
PostgreSQL	SQL/DS
SQLite	Sybase (Sybase Adaptive Server Enterprise, Sybase SQL Anywhere, Sybase IQ)
Terawata	

Lp.	Hasło	Opis
1.	SQL	(Structured Query Language – Strukturalny język zapytań), opracowany przez firmę IBM jest rozwinięciem języka SEQUEL (Structured English QUery Language) jest językiem służącym do budowania systemów obsługi relacyjnych baz danych (RDBMS) na dowolnej płaszczyźnie sprzętowej. W chwili obecnej jest on standardowym językiem służącym do wymiany danych pomiędzy różnymi platformami sprzętowymi i programowymi
2.	SELECT	Wybierz. Klauzula ta to podstawowy składnik zapytania. Po tym słowie umieszczamy nazwy kolumn, które chcemy uzyskać w odpowiedzi. Specjalne znaczenie ma tu znak * wstawienie tego znaku spowoduje w odpowiedzi wszystkie możliwe kolumny
3.	INSERT	Wstawia rekord (wiersz w tabeli)
4.	UPDATE	Aktualizacja danych tabeli
5.	DELETE	Kasuje rekord (wiersz tabeli)
6.	FROM	Słowo kluczowe służące do określenia źródła danych, z których pobierane będą informacje
7.	DISTINCT	Zapewnia różnowartościowość odpowiedzi
8.	WHERE	Po tym słowie kluczowym podaje się warunki selekcji
9.	AND	Operator koniunkcji. Stosuje się w specyfikowaniu warunków selekcji
10.	OR	Operator alternatywy. Stosuje się w specyfikowaniu warunków selekcji
11.	NOT	Operator negacji. Stosuje się w specyfikowaniu warunków selekcji
12.	ORDER BY	Klauzula służąca uporządkowaniu odpowiedzi według zadanych pól (porządek może być rosnący ASC lub malejący DSC)
13.	GROUP BY	Słowo grupujące listę odpowiedzi na podzbiory według zadanego pola. Na tak utworzonych podzbiorach można wykonać funkcje agregujące
14.	IN	Pozwala sprawdzić czy wartość danego atrybutu należy do podanego zbioru (zbiorem może także być rezultat klauzuli SELECT)
15.	IS NULL	NULL oznacza pustą wartość komórki

Lp.	Hasło	Opis
16.	AS	Przy pomocy tego słowa można w tabeli z odpowiedzą zmienić domyślne nazwy kolumn
17.	BETWEEN	Za jego pomocą możemy określić przedział wartości w jakim ma się znajdować wartość atrybutu (można użyć > <)
18.	Znaki specjalne	_ znak podkreślenia zastępuje w napisie dokładnie jeden znak % zastępuje dowolny ciąg znaków (także pusty)
19.	LIKE	Operator podobnie jak = Pozwala na porównywanie napisów

SELECT

SELECT kolumny
 FROM tabele
 [WHERE warunki]
 [GROUP BY grupa
 [HAVING warunki grupy]]
 [ORDER BY sort_kolumn]
 [LIMIT ograniczenia]

PRZYKŁADY

select * from wydział;
 z tabeli wydział zostaną wyświetlone wszystkie rekordy
 select nazwisko, id_p from pracownik
 wyświetli nazwiska i ud_ z tabeli pracownik
 select pracownik.nazwisko from pracownik;
 Warunek WHARE
 Select id_p, nazwisko from pracownik where stanowisko='programista';

Operatory

= równy
 != lub <> różny od
 >
 <
 >=
 <=

AND, OR, NOT, XOR

Select count(*) from pracownik;
 Select * from przydzial where id_p=6651 and godziny>8;

DISTINCT

Select distinct stanowisko from pracownik;
 Select count(stanowisko) from pracownik
 Select count(distinct stanowisko) from pracownik;
 select count(*), stanowisko from pracownik group by stanowisko;
 select count(*), stanowisko from pracownik group by stanowisko dsc;
 select count(*), stanowisko from pracownik group by stanowisko having count(*)=1;
 wybranie stanowisk, które w firmie zajmuje tylko jeden pracownik

ORDER

select * from pracownik order by stanowisko asc, nazwisko dsc;
 select * from umyprac limit 5;
 select * from umyprac limit 5,3;
 wyświetla od 5 wiersza trzy kolejne

Łączenie dwóch tabel

select pracownik.nazwisko, wydzial.nazwa from pracownik, wydzial
 where pracownik.idwydz=wydzial.idwydz;

Funkcje agregujące

MIN – do wyznaczania minimum

MAX – do wybrania maksymalnego elementu

SUM - do sumowania wartości danej kolumny

COUNT - do zliczania liczby obiektów spełniających warunek podany w klauzuli WHERE

AVG – do wyznaczania średniej wartości kolumny

Funkcje tekstowe

CONCAT(str1,str2,...) – łączy ciągi znakowe

LCASE(str) – zamienia na małe litery

LCASE() jest odpowiednikiem LOWER().

LEFT(str,n) pozostawia n znaków od lewej

LENGTH(str) – oblicza długość ciągu znakowego

LOCATE(substr,str) , LOCATE(substr,str,pos) określa wystąpienie substr w str

LTRIM(str)

MID(str,pos,len)

MID(str,pos,len) synonim SUBSTRING(str,pos,len), pobiera z ciągu od m, n znaków

ORD(str) zwraca kod znaku

REPLACE(str,z_ciągu,na_ciąg)

RIGHT(str,n) –pozostawia n znaków od prawej strony

SPACE(N) zwraca ciąg z N spacji

UCASE(str) zamienia na duże litery

UCASE() synonim UPPER().

ABS(X) wartość bezwzględna

Funkcje matematyczne

ASIN(X)

ATAN(X)

EXP(X)

FLOOR(X) zwraca liczbę całkowitą z X

LN(X)

LOG(X) , LOG(B,X) logarytm naturalny

ROUND(X) , ROUND(X,D) zaokrągla do D miejsc po przecinku

SQRT(X) pierwiastek kwadratowy

Funkcje daty

CURDATE() YYYY-MM-DD

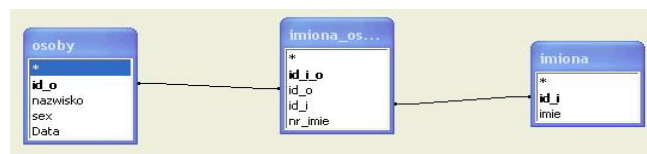
CURTIME()

DATE(expr)

Funkcje bazy danych

LAST_INSERT_ID() id ostatnio dodanego rekordu

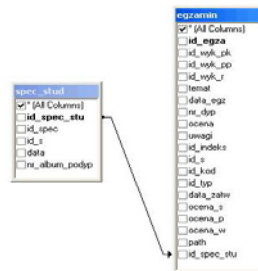
Sprzężenie wewnętrzne



```
SELECT osoby.nazwisko, imiona.imie, imiona_osoby.nr_imie FROM osoby INNER JOIN (imiona INNER JOIN imiona_osoby ON imiona.id_i = imiona_osoby.id_i) ON osoby.id_o = imiona_osoby.id_o WHERE (((imiona_osoby.nr_imie)=1));
```

Sprzężenie Prawe

```
SELECT * FROM egzamin RIGHT JOIN spec_stud ON egzamin.id_spec_stu = spec_stud.id_spec_stu
```

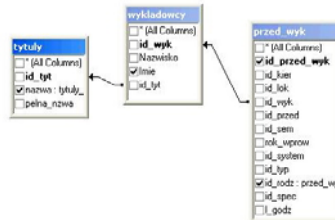


Sprzężenie prawe –wszystkie rekordy z spec_stud i tylko te, które są równew egzamin

Aktualizacja danych w tabeli

```
UPDATE aktualizacja SET aktualizacja.kwota = 1200;
```

Przykład instrukcji select z funkcjami i aliasami



```
SELECT id_przed_wyk, concat(tytuly.nazwa," ",imie," ",Nazwisko) AS prw, Imie, tytuly.nazwa AS  
tytuly_nazwa, przed_wyk.id_rodz AS przed_wyk_id_rodz  
FROM (wykladowcy RIGHT JOIN przed_wyk ON  
przed_wyk.id_wyk = wykladowcy.id_wyk) LEFT JOIN tytuly ON  
wykladowcy.id_tyt = tytuly.id_tyt WHERE przed_wyk.id_lok = s_przed_wyk_id_lok ORDER BY  
przed_wyk.id_przed
```