

Projektowanie hurtowni danych

Wspomaganie zarządzania relacjami z klientami

UMIĘJĘTNIE ZARZĄDZAJ INFORMACJAMI!

- Czym są hurtownie danych?
- Jak je zaprojektować, aby wykorzystać maksimum ich możliwości?
- Jak wydajnie zarządzać relacjami z klientem?

Chris Todman

Prentice
Hall



» Idź do

- Spis treści
- Przykładowy rozdział
- Skorowidz

» Katalog książek

- Katalog online
- Zamów drukowany katalog

» Twój koszyk

- Dodaj do koszyka

» Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

» Czytelnia

- Fragmenty książek online

» Kontakt

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 32 230 98 63
e-mail: helion@helion.pl
© Helion 1991–2011

Projektowanie hurtowni danych Wspomaganie zarządzania relacjami z klientami

Autor: Chris Todman

Tłumaczenie: Paweł Gonera

ISBN: 978-83-246-3225-1

Tytuł oryginału: [Designing A Data Warehouse:
Supporting Customer Relationship Management](#)

Format: 172×245, stron: 296



Umiejętnie zarządzaj informacjami!

- Czym są hurtownie danych?
- Jak je zaprojektować, aby wykorzystać maksimum ich możliwości?
- Jak wydajnie zarządzać relacjami z klientem?

Zastanawiasz się, jak zmaksymalizować zyski czerpane z właściwego zarządzania relacjami z klientem? Jest to pytanie, które spędza sen z oczu każdemu przedsiębiorcy. Jedną z możliwości jest wykorzystanie odpowiednio zaprojektowanej hurtowni danych. Projekt takiej bazy danych, uwzględniającej potrzeby klienta, wymaga nowych technik i metodologii. Jakich?

Na to pytanie odpowiada Chris Todman w książce, którą trzymasz w rękach. Wiodący konsultant w dziedzinie hurtowni danych przedstawi Ci kompletną metodologię, pozwalającą na zaprojektowanie, wytworzenie i wdrożenie hurtowni danych pod kątem zarządzania relacjami z klientem. Najpierw przeczytasz o kilku zagadnieniach teoretycznych, związanych z relacjami z klientem oraz hurtowniami danych. Potem zapoznasz się z typowymi problemami, aby w rozdziale piątym przejść do omówienia modelu koncepcyjnego. Dowiesz się, jak obsługiwać okoliczności, identyfikować zmiany w danych oraz modelować metodą kropki. Kolejne omawiane zagadnienia to model logiczny i sposoby rozwiązywania problemów wydajnościowych. W rozdziale poświęconym implementacji fizycznej zobaczysz, jak kontrolować poprawność danych, zarządzać kopiami zapasowymi oraz aplikacjami CRM. Ponadto nauczysz się zarządzać projektem, przejrzysz możliwości dostępnego oprogramowania oraz zdobędziesz wiedzę o perspektywach rozwoju. To wszystko znajdziesz w tej długo oczekiwanej książce, poświęconej hurtowniom danych.

- Zarządzanie relacjami z klientem
- Wartość marki
- Zarządzanie kampaniami i marketing personalizowany
- Budowanie hurtowni danych – komponent ekstrakcji oraz integracji danych
- Baza danych hurtowni – opis encji
- Problemy przy wykorzystywaniu relacyjnych baz danych
- Problemy z obróbką czasu w hurtowniach danych
- Wybór modelu koncepcyjnego
- Modelowanie metodą kropki
- Modelowanie logiczne – schemat, wybór rozwiązania, ograniczenia
- Implementacja fizyczna
- Zarządzanie kopiami zapasowymi
- Aplikacje CRM
- Zarządzanie projektem
- Oprogramowanie
- Przyszłość hurtowni danych

Sprawdź, jak zmaksymalizować zyski poprzez właściwe zarządzanie relacjami z klientem!

Spis treści

Wprowadzenie	7
Podziękowania	11
1. Zarządzanie relacjami z klientem	13
Wymiar biznesowy	13
Cele biznesowe	15
Strategia biznesowa	16
Wartość marki	17
Zarządzanie relacjami z klientem	18
Podsumowanie	30
2. Wprowadzenie do hurtowni danych	31
Wprowadzenie	31
Czym jest hurtownia danych?	37
Analiza wymiarów	38
Budowanie hurtowni danych	41
Problemy związane z zastosowaniem relacyjnych baz danych	58
Podsumowanie	59
3. Problemy projektowe do rozwiązania	61
Wielowymiarowe modele danych	61
Jak wspierać CRM?	69
Podsumowanie	79
4. Problemy z obróbką czasu w hurtowniach danych	81
Rola czasu	81
Problemy związane z czasem	84
Rejestrowanie zmian	90
Rozwiązania problemów z czasem w hurtowniach danych pierwszej generacji	94
Odmienne podejścia	107
Konkluzje z przeglądu metod pierwszej generacji	109

5. Model koncepcyjny	111
Wymagania modelu koncepcyjnego	111
Identyfikowanie zmian w danych	118
Modelowanie metodą kropki	119
Warsztaty modelowania metodą kropki	124
Podsumowanie	147
6. Model logiczny	149
Modelowanie logiczne	150
Implementacja retrospekcji	150
Zastosowanie wymiaru czasu	157
Schemat logiczny	161
Problemy wydajnościowe	163
Wybór rozwiązania	165
Częstotliwość przechwytywania zmian danych	166
Ograniczenia	167
Weryfikacja i podsumowanie modelu logicznego	171
7. Implementacja fizyczna	173
Architektura hurtowni danych	173
Aplikacje CRM	193
Kopia zapasowa danych	193
Archiwizacja	194
Ekstrakcja i ładowanie	194
Podsumowanie	197
8. Uzasadnienie biznesowe	199
Podejście przyrostowe	199
Wdrożenie	205
Podsumowanie	207
9. Zarządzanie projektem	209
Wprowadzenie	209
Co jest dostarczane?	212
Jakie należy zdefiniować założenia i ryzyka?	213
Jakiego zespołu potrzebujemy?	215
Podsumowanie	226
10. Oprogramowanie	227
Ekstrakcja, przekształcanie i ładowanie	228
OLAP	230
Narzędzia zapytań	232
Wydobywanie danych	234
Zarządzanie kampaniami	238
Personalizacja	240
Narzędzia metadanych	241
Sortowanie	242
11. Przyszłość	245
Temporalne bazy danych (rozszerzenia temporalne)	246
Rozszerzenia OLAP dla SQL	246
Aktywne wsparcie decyzji	247

Dane zewnętrzne	247
Dane niestrukturalizowane	248
Agenci wyszukiwania	248
Aplikacje obsługujące DSS	249
A. Temporalna klasyfikacja danych Klubu Winiarza	251
B. Model kropki dla Klubu Winiarza	255
C. Model logiczny dla Klubu Winiarza	269
D. Atrybuty klienta	275
Atrybuty finansowe	281
Atrybuty zatrudnienia	282
Atrybuty zainteresowań i hobby	283
E. Odnośniki	287
Skorowidz	289

Zarządzanie projektem

Żaden projekt nie może być ukończony bez przygotowania odpowiedniego planu! W tym rozdziale przedstawię różnice pomiędzy projektami tworzenia hurtowni danych a tradycyjnymi projektami dostarczania oprogramowania. Tłumaczy to, dlaczego organizacjom nie udaje się doprowadzić procesu do końca, jeżeli próbują korzystać wyłącznie z tradycyjnych metod, znanych z „twardych systemów”. Proces ten musi zostać zmodyfikowany, aby pasował do „miękkich systemów”. Dodatkowo zaprezentuję pełną strukturę podziału zadań (WBS) dla projektowania i tworzenia przyrostów hurtowni danych. WBS zawiera ponad 100 elementów, które muszą być zrealizowane w każdym przyroście.

Wprowadzenie

Odpowiednie zarządzanie jest często kluczem do sukcesu projektu. Każdy projekt potrzebuje kierownika oraz zastosowania pewnej metodologii zarządzania. Nie ma znaczenia, w jakiego rodzaju projekt jesteśmy zaangażowani. Projekty dużych fizycznych konstrukcji, jak mosty, statki czy tunele, również muszą być zarządzane. Choć czasami kończą się spektakularną kląpą, to jednak mamy znacznie większe doświadczenie w budowaniu tego typu struktur niż w budowaniu dużych systemów informatycznych. Budowanie fizycznych struktur ma jedną dużą zaletę w porównaniu z budowaniem oprogramowania: na wszystkich etapach pracy można zobaczyć, co zostało zrobione. Nawet niewykszolony obserwator może spojrzeć na most i określić, w jakiej części jest on wykonany. To, czy budowane są fundamenty, podpory, czy przęsło, jest dosyć oczywiste i każdy z nas może zaryzykować określenie ilości pozostałych zadań. W przypadku oprogramowania tak nie jest. Nawet wykwalifikowany obserwator będzie miał problem z określeniem, jaka część projektu została zakończona.

Innym problemem jest poziom złożoności systemów komputerowych, który jest znacznie większy niż w przypadku innych projektów. Jako przykład weźmy odrzutowiec. Projektowanie i budowanie takiej niesamowitej maszyny jest bardzo wymagającym zadaniem. Jednak najbardziej złożoną częścią nie są skrzydła, silnik czy też kadłub, ale oprogramowanie sterujące systemami samolotu oraz oprogramowanie, które jest używane do zarządzania tymi systemami. Systemy pozwalają na kierowanie samolotem i zarządzanie jego uzbrojeniem oraz dostarczają pilotowi informacji, co się dzieje wewnątrz i na zewnątrz maszyny. Ponadto skrzydła, silnik

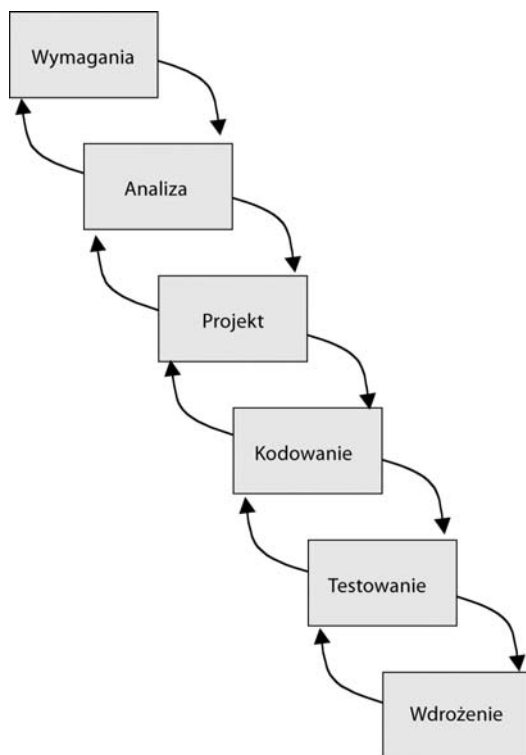
i kadłub zostały zaprojektowane z użyciem oprogramowania. Nie możemy zobaczyć żadnego z tych elementów, więc nasze podejście do zarządzania takim projektem musi być bardziej złożone niż w przypadku konwencjonalnych przedsięwzięć.

Jednym ze sposobów na rozwiązanie tego problemu jest połączenie metodologii zarządzania projektem z metodologią tworzenia oprogramowania użytą w tym projekcie. Istnieje wiele różnych metodologii tworzenia oprogramowania, ale ostatecznie dzielą się one na dwie główne kategorie:

- metodę wodospadu,
- metodę szybkiego tworzenia aplikacji (RAD).

Zagadnienie to było omówione w rozdziale 1. Możemy oczekiwać, że większość Czytelników tej książki zna te metody, więc wyjaśnienie będzie krótkie.

Istnieje wiele odmian metody wodospadu, które różnią się wyborem głównego komponentu. Przyjmuje się, że powinny być zastosowane fazy zbierania wymagań, analizy, projektowania, kodowania, testowania oraz wdrażania, które są często przedstawiane na diagramie pokazanym na rysunku 9.1.



Rysunek 9.1. *Klasyczna metoda wodospadu*

Zasadą metody wodospadu jest konieczność zakończenia tworzenia każdego głównego komponentu przed rozpoczęciem pracy nad następnym. Dlatego przed analizą muszą być zebrane wszystkie wymagania, a przed przystąpieniem do projektowania niezbędne jest zakończenie fazy analizy. Jest to często nazywane metodą od startu do mety.

Innym ważnym założeniem metody wodospadu jest to, że zwykle nie cofamy się więcej niż o jeden krok. Jeżeli więc okaże się w czasie testowania, że należy zmienić kod, to wykonuje się iteracje pomiędzy kodowaniem i testowaniem aż do momentu, gdy kod przejdzie wszystkie testy. W świecie rzeczywistym często się zdarza, że testowanie wykrywa złe zrozumienie części analizy dotyczącej jednego z wymagań.

Częściowe ponowne opracowywanie wymagań, a następnie modyfikowanie projektu, kodu i planu testów jest czasochłonne, demoralizujące oraz oczywiście bardzo kosztowne — dlatego właśnie metoda wodospadu nie jest już polecana. Pomimo tego nadal istnieje ona w różnych formach jako jedna z głównych metod produkcji oprogramowania. Jest szczególnie często wykorzystywana przy tworzeniu hurtowni danych, gdzie w niektórych przypadkach jest nadal sensowna.

Metody RAD działają w oparciu o nieco inne zasady i mogą być efektywne dla aplikacji, które mają następującą charakterystykę:

- *Interaktywność* — funkcje aplikacji są jasno widoczne w interfejsie użytkownika. RAD bazuje na przyrostowym prototypowaniu aplikacji w ścisłej współpracy z użytkownikami. Muszą oni być w stanie w łatwy sposób ocenić funkcje, gdy zostaną im zademonstrowane działające prototypy.
- *Jasno zdefiniowana grupa użytkowników*. Jeżeli grupa użytkowników nie jest jasno zdefiniowana, może zaistnieć niebezpieczeństwo skierowania tworzenia aplikacji w złym kierunku lub (co gorsza) nastąpić całkowite zignorowanie ważnych elementów aplikacji.
- *Brak złożoności obliczeniowej*. Poziom złożoności obliczeniowej aplikacji jest trudny do określenia i może się zmieniać w różnych projektach. Jeżeli aplikacja wymaga na przykład złożonego modelowania statystycznego, projekt może zakładać dwa podejścia do programowania — dopasowanie istniejącego, dobrze przetestowanego komponentu modelowania statystycznego lub opracowanie modelu od początku. Pierwsze będzie akceptowalne w projekcie RAD. Drugie może być uznane za ryzykowne, o ile nie istnieje możliwość podziału komponentu na mniejsze lub zapewnienie jego przezroczystości dla interfejsu użytkownika.
- *Jeżeli są duże, mają możliwość podziału na mniejsze komponenty funkcjonalne*. Jeżeli system jest duży, powinien dawać możliwość podziału na mniejsze, łatwiejsze do zarządzania fragmenty realizujące jasno zdefiniowane funkcje. Elementy te mogą być dostarczane etapami lub równolegle. Jednak część funkcji może być realizowana z użyciem standardowej metody wodospadu.
- *Ograniczenie czasowe*. Musi zostać określona data zakończenia projektu. Jeżeli nie jest ona podana, dosyć często zdarzają się poślizgi i podstawowa zaleta metody RAD będzie niewykorzystana.

Choć jest jasna różnica pomiędzy metodą wodospadu a metodą RAD, istnieje jedno podstawowe podobieństwo — obie dotyczą tworzenia tradycyjnych aplikacji. Aplikacje zazwyczaj realizują zbiór funkcji biznesowych. Oznacza to, że dużo mówi się o funkcjach oraz procesach. Jeżeli pomyślimy o hurtowni danych, to gdzie wyróżnimy funkcję, a gdzie procesy? Oczywiście istnieją pewne procesy, na przykład przetwarzanie VIM, ale hurtownia danych nie realizuje procesów biznesowych — jej zadaniem jest wspieranie innych aplikacji, na przykład wspomaganie decyzji lub zarządzania kampaniami. W tym sensie hurtownia danych nie jest aplikacją biznesową. Trochę to dziwne, ale już wcześniej wspominałem, że hurtownie danych są inne. Musimy więc rozważyć, w jaki sposób będziemy je tworzyć.

Co jest dostarczane?

Jednym z elementów, które muszą koniecznie być zrealizowane prawidłowo w każdym projekcie, jest upewnienie się, że klient otrzyma to, czego oczekiwał. W rozdziale 1. przedstawiłem podejście z użyciem celów biznesowych do ustalenia oczekiwań, ale nadal mogą wystąpić problemy w czasie odbioru. Kierownik projektu musi uzyskać akceptację klienta, zanim projekt będzie mógł być oficjalnie zamknięty. To bardzo ważne, ponieważ firma konsultingowa właśnie wtedy otrzymuje płatność. Nawet w wewnętrznych projektach odbiór jest istotny, gdyż wskazuje on moment, w którym system został zaakceptowany. Ten moment jest zawsze przyjmowany z dużą ulgą przez osoby zaangażowane w projekt i zwykle jest to okazja do świętowania. Dodatkowo pozwala to przydzielić członków zespołu do następnych projektów. Dla niektórych być może stanowi to początek następnego etapu budowy hurtowni danych.

W jaki sposób uzyskać końcowy podpis? Większość wykwalifikowanych kierowników projektu uzgadnia listę warunków akceptacyjnych, jeszcze zanim przejmą odpowiedzialność za projekt. Jest to jeden z tych trudnych problemów, z którymi musimy sobie poradzić. Czasami zdarza się, że jeżeli hurtownia danych została zbudowana na solidnych podstawach i właściwie obsługuje cele biznesowe, to wystarczający jest fakt, że hurtownia fizycznie materializuje się na końcu procesu. Zwykle jednak klient wymaga więcej.

Dzieje się tak, ponieważ hurtownie danych są inne. Jeżeli nie byłaby to hurtownia danych, ale na przykład system przetwarzania zamówień, moglibyśmy opracować z klientem zbiór kryteriów akceptacyjnych i wykonać testy akceptacyjne na zestawie danych testowych. Następnie możemy przetworzyć zamówienie, upewnić się, że konto klienta jest prawidłowo aktualizowane, że dokument dostawy powoduje wygenerowanie faktury itp. Jest to możliwe, ponieważ większość systemów jest ukierunkowana na proces i jesteśmy w stanie sprawdzić, czy przebiega on w taki sposób, że używane przez niego dane zostają prawidłowo przetworzone. W przypadku tworzenia hurtowni danych procesy dotyczą tylko pozyskiwania danych z systemów źródłowych i ich ładowania do hurtowni danych w odpowiedniej postaci. Możemy i powinniśmy zrealizować kontrolę poprawności w celu upewnienia się, że wszystkie rekordy, które zostały pobrane z systemu źródłowego, mogą być dodane do bazy danych, ale nie wystarczy to do wytworzenia testów akceptacyjnych. To, co jest najważniejsze, to zawartość hurtowni. Niestety, każdy przypadek będzie prawdopodobnie inny, ale poniżej przedstawię kilka sugestii.

- *Powiązanie hurtowni danych z jedną z aplikacji.* Przykładem może być system zarządzania kampaniami. Możemy uzgodnić zbiór kryteriów segmentacji przy wyborze klientów, do których ma trafić kampania, i zbudować aplikację, za której pośrednictwem system zarządzania kampaniami otrzyma listę odpowiednich klientów. Kryterium akceptacji mogą być parametry wyboru tych klientów. Nasi użytkownicy będą w stanie sprawdzić otrzymaną listę.
- *Wybranie kilku pytań, które zostały postawione na warsztatach strategii informacyjnej.* Jeżeli będziemy w stanie wykonać kilka z tych pytań, udowodni to, że hurtownia danych może dostarczać organizacji wartościowych informacji. Należy uważnie wybierać pytania, aby mogły być zrealizowane przez mechanizmy dostarczane w pierwszych etapach. Nie ma sensu wykonywać zapytań związanych ze zmieniającymi się okolicznościami klientów, jeżeli dostarczyliśmy część dotyczącą zachowań przy kupnie wina.

Należy również zachować ostrożność w przypadku kryteriów akceptacji odnoszących się do wydajności systemu. Często nasi klienci żądają zapewnienia sztamkowych metryk wydajności, takich jak:

Wszystkie zapytania będą realizowane w czasie poniżej 10 sekund.

Może to być ogromny problem, ponieważ tego typu kryteria akceptacji najprawdopodobniej uniemożliwią odbiór końcowy. W żadnym razie nie wolno zgadzać się na tego typu warunki. Należy wyjaśnić klientowi, że hurtownie danych różnią się od konwencjonalnych aplikacji. Zazwyczaj rozsądne jest żądanie, aby aplikacje OLTP odpowiadały niemal natychmiast. W większości przypadków realizowane przez nie transakcje przebiegają w następujący sposób:

- (1) wyszukanie jednego rekordu w bazie danych (z użyciem klucza unikatowego),
- (2) zmiana pewnej wartości,
- (3) zapisanie rekordu.

Jak wszyscy wiedzą, tego typu operacje mogą być realizowane w mgnieniu oka. Zapytanie hurtowni danych zwykle musi:

- (1) odczytać miliony rekordów,
- (2) dopasować je to tysięcy innych rekordów z użyciem złączenia,
- (3) wyliczyć sumy częściowe,
- (4) zaprezentować wyniki w czytelnym formacie.

Każdy, kto ma pewne doświadczenie z bazami danych, zaświadczy, że może to zająć sporo czasu. Jeżeli jest to możliwe, należy unikać kryteriów akceptacyjnych bazujących na wydajności. Nie oznacza to, że możemy ignorować problemy wydajnościowe, tylko dlatego, że system nie będzie sprawdzany na ich okoliczność.

Powinniśmy prezentować pogląd, że odpowiedzi na niektóre pytania są niemożliwe lub niemal niemożliwe do uzyskania. Jeżeli jesteśmy w stanie otrzymać odpowiedzi na wszystkie pytania, wskazuje to, że system odniesie sukces. Strojenie wydajności jest czymś, co można zawsze przeprowadzić w później, stosując standardowe metody optymalizacji relacyjnych baz danych oraz techniki specyficzne dla hurtowni danych, takie jak tabele podsumowań.

Jakie należy zdefiniować założenia i ryzyka?

Każdy plan projektu zawiera lub powinien zawierać założenia. Na początku projektu nigdy nie wiemy wszystkiego, więc musimy czasami zgadywać. Nazywa się to założeniami. Różne projekty wymagają różnych założeń, ponieważ są realizowane dla różnych klientów. Różnica pomiędzy założeniem i ryzykiem nie jest jasna w kontekście planu projektu, ponieważ możemy ograniczać ryzyko przez przyjmowanie założeń. Na przykład możemy uznać za ryzyko możliwość, że nasz główny architekt rozwiązania zostanie potrącony przez ciężarówkę, więc ograniczamy je, zakładając, że tak się nie stanie. Jeżeli wolisz prowadzić rejestr ryzyk, to i tak ten podrozdział nadal jest istotny.

Wymienię teraz założenia, jakie powinniśmy przyjąć, planując projekt hurtowni danych:

- *Jakość danych.* Jeżeli projekt zawiera zadanie związane z analizą jakości danych, to jest to prawidłowe postępowanie. Jeśli tego zadania nie ma, to musimy polegać tylko na twierdzeniach klienta, że dane są odpowiedniej jakości. Doświadczenie pokazuje, że klienci często przesadzają z oceną jakości swoich danych. Zwykle nie jest to próba oszukania nas, ale po prostu nie wiedzą oni, jak niskiej jakości są ich dane. Przykładem niskiej jakości danych są dane z brakami, dane powielone, błędy odwołań itp. Często zdarza się, że istnieje kilka baz danych klientów (najgorszym przypadkiem, z jakim się spotkałem,

były 23 takie bazy), a każda z nich rozwijała się w czasie i każda była stosowana do innych celów. Każda z tych baz danych zawiera małą część informacji, których potrzebujemy w hurtowni. Niestety, nigdy nie są one spójne. Nieraz w różnych bazach danych stosuje się różne systemy kodowania, a identyfikatory klientów w jednym systemie są zupełnie inne niż w pozostałych. Może nam się wydawać, że jeżeli każdy system zawiera fragment układanki, wystarczy wykonać ogromne złączenie tabel i mamy wszystkie potrzebne informacje. Czasami się to udaje, a czasami nie. Często okazuje się, że konieczna będzie tabela odwzorowań, taka jak pokazana poniżej, która pozwoli utrzymać spójność identyfikatorów klientów w hurtowni danych.

Odwzorowanie identyfikatorów klientów

ID klienta hurtowni char(8)
 ID klienta przetwarzania zamówień number(6)
 ID klienta sprzedaży char(6)
 ID klienta księgowości char(6)

Podejście to działa, jeżeli będzie istniała relacja „jeden do jednego” pomiędzy systemami, ale często tak nie jest. Baza sprzedawców może definiować klientów na różnych poziomach, przez co firmy córki mogą być interpretowane jako jednostkowi klienci, natomiast system księgowy będzie zainteresowany tylko klientami, którzy opłacają faktury. Kolejnym problemem jest brak spójności danych pomiędzy poszczególnymi źródłami. Adresy mogą być różne, informacje mogą być bardziej aktualne w jednym systemach niż w innych itp. Różnice te składają się na ogólny problem związany z jakością danych. Nasz klient rzadko kiedy rozpoznaje te nieścisłości. Jeżeli nie zarezerwujemy czasu w projekcie na analizę danych, to trzeba przyjąć założenia na temat jakości danych. Niska jakość danych może w znacznym stopniu wstrzymać realizację projektu hurtowni danych.

- *Dostępność danych.* Musimy być w stanie pobierać wszystkie potrzebne nam dane, a czasami może być to trudne. Przedstawiłem już problem pobrania zmienionych danych przy okazji opisu problemów temporalnych w rozdziale 4., ale często podobne kłopoty mogą być z danymi zachowań. Zdarza się, że dane zachowań są dostępne w pewnym momencie w czasie conocnego przetwarzania wsadowego. W jednym z etapów cyklu przetwarzania wsadowego odpowiednie dane są umieszczane w pliku, dzięki czemu mogą być przekazane do hurtowni danych w celu przetworzenia. Jeżeli proces wsadowy nie dostarczy danych z jakiegoś powodu, to hurtownia nie może być zaktualizowana. Choć wydaje się to problemem operacyjnym, a nie programistycznym, to jednak nasz klient może nie zaakceptować systemu, który nie zapewnia odpowiedniej dostępności danych. Upewnienie się, że tego typu problemy nie wystąpią, jest bardzo trudne. Lepiej dodać założenie do planu, w którym odpowiedzialność za terminowe dostarczenie danych spoczywa na kliencie.
- *Nocne okno przetwarzania.* Jest to problem podobny do poprzedniego. Jesteśmy odpowiedzialni za udostępnianie hurtowni danych przez określoną część dnia, na przykład od 8.00 do 20.00. Potrzebujemy pewnej liczby cykli przetwarzania w postaci conocnego okna czasowego, w którym wykonywane jest ładowanie danych i przeprowadzane są czynności konserwacyjne. Nie jest to problem, ale czasami zdarzają się przypadki, w których okno to staje się bardzo małe. Na przykład na koniec miesiąca, kwartału, a szczególnie roku firma musi wykonać dodatkowe przetwarzanie danych. Wtedy musimy ustąpić miejsca. Ponadto gdy pojawią się problemy w „krytycznych dla firmy” operacjach, jeżeli pakiety oprogramowania muszą być odtworzone, co jest poprzedzone długim procesem odtwarzania

bazy danych itp., to conocne przetwarzanie wsadowe nachodzi na nasze okno czasowe i znów musimy zmieścić się w krótszym czasie. Tak jak wcześniej, niezwykle trudno jest zawrzeć w kodzie zabezpieczenia przed taką sytuacją i najlepsze, co możemy zrobić, to dopisać założenie, że nasze conocne okno przetwarzania musi pozostać otwarte.

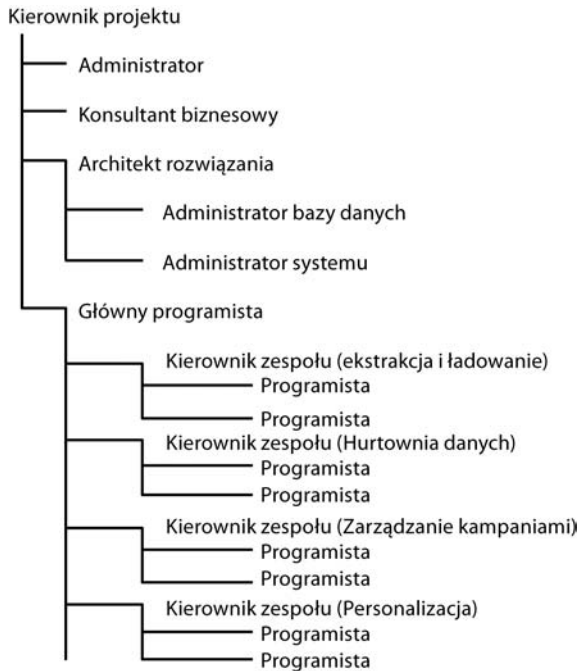
- *Sponsor biznesu.* Niezwykle ważne jest, aby w firmie klienta był wysoko postawiony sponsor. Ta osoba musi „być właścicielem” projektu po stronie klienta. Nie można nie doceniać wagi tej roli w końcowym sukcesie projektu. Osoba ta musi być entuzjastycznie nastawiona do projektu oraz mieć odpowiednie uprawnienia do samodzielnego podejmowania decyzji. Sponsor musi być właścicielem projektu przez cały czas jego trwania. Jeżeli opuści on projekt z jakiegś powodu, stanowi to poważne zagrożenie dla sukcesu przedsięwzięcia. Nowi sponsorzy, którzy przejmują projekt w połowie jego trwania, rzadko kiedy rozumieją go tak dobrze, jak osoba zaangażowana w projekt od początku. Warto więc dodać założenie, że sponsor projektu po stronie klienta pozostanie ten sam przez cały czas trwania projektu.
- *Wiedza na temat systemów źródłowych.* Jest to kolejny poważny problem, szczególnie niebezpieczny, gdy systemy źródłowe starzeją się oraz gdy były opracowywane własnymi siłami. Większość projektantów i programistów jest już niedostępna. System może być rozszerzany i dostosowywany przez lata działania, a dokumentacja, o ile istnieje, nie jest tak rygorystycznie aktualizowana. Krótko mówiąc, nikt nie wie zbyt dużo na ten temat. Istnieją pewne pliki lub miejsca w cyklu przetwarzania, w których można uzyskać potrzebne dane, ale nie ma nikogo, kto dostarczy pełnego opisu pól danych w rekordach. Czasami zdarza się to również w całkiem nowych systemach, szczególnie w przypadku dużych pakietów oprogramowania. Dowiedzenie się, jak działa interesujący nas system, często jest koszmarem. Dobrym pomysłem jest więc założenie, że klient może wskazać osoby, które w pełni rozumieją dane systemy.

Jakiego zespołu potrzebujemy?

Aby prawidłowo zaplanować projekt, jego kierownik musi wiedzieć, kiedy i jakich zasobów będzie potrzebował. Zaczniemy od nakreślenia struktury zespołu produkcji oprogramowania — rysunek 9.2.

Osoby pełniące funkcje wymienione na rysunku 9.2 powinny mieć następującą charakterystykę:

- *Kierownik projektu.* Musi on dobrze radzić sobie z dużym poziomem niejednoznaczności, niepewności i zmienności. Projekty hurtowni danych różnią się od normalnego procesu produkcji oprogramowania, ponieważ muszą być bardziej elastyczne i dostosowywać się do zmiennych warunków biznesowych; nie są tak dokładnie zdefiniowane, jak byśmy tego chcieli. Doświadczony kierownik projektu potrafi dostosować się do płynnych wymagań projektu hurtowni danych. Kierownicy projektów, którzy przyzwyczaili się do pracy na podstawie stałych założeń, na przykład dokładnej specyfikacji systemu, uznają te projekty za bardzo trudne.
- *Konsultant biznesowy.* Jest to funkcja krytyczna dla sukcesu projektu. Absolutnie *nie* mamy tu na myśli konsultanta *zarządzającego* — konsultant biznesowy to osoba, która pomoże klientowi zrozumieć zalety CRM i potrafi wyjaśnić, w jaki sposób wszystkie komponenty współpracują ze sobą w celu wspomaganie wdrażania strategii CRM. W idealnej sytuacji konsultant biznesowy powinien dobrze znać środowisko biznesowe klienta. Osoba ta jest odpowiedzialna za zbieranie wymagań biznesowych oraz pomoc przy tworzeniu modelu koncepcyjnego.



Rysunek 9.2. Przykładowa struktura zespołu projektowego

- *Architekt rozwiązania.* Osoba ta jest również kluczowym graczem w zapewnieniu końcowego sukcesu projektu. Architekt rozwiązania specyfikuje komponenty w hurtowni, upewniając się, że wszystkie dobrze do siebie pasują. Funkcja architekta rozwiązania jest najważniejszą funkcją techniczną w projekcie. Jest on odpowiedzialny za zrozumienie wymagań i przekształcenie ich w rozwiązanie. Ważne jest, aby osoba na tym stanowisku miała dużą wiedzę nie tylko na temat integracji systemów, ale również na temat hurtowni danych. W projektach CRM tradycyjne techniki hurtowni danych muszą być zmodyfikowane w sposób opisany w tej książce.
- *Główny programista.* W wielu aspektach funkcja ta może być uznawana za dosyć podobną do funkcji kierownika produkcji oprogramowania w tradycyjnych projektach. W hurtowni danych istnieje wiele niewielkich podprojektów, które działają w niej jednocześnie. Każdy z tych podprojektów jest realizowany przez mały zespół złożony z głównego programisty oraz podlegających mu programistów. Zespoły te równoległe tworzą proces ekstrakcji i ładowania, samą hurtownię danych, jak również aplikacje, na przykład zarządzanie kampaniami, jak pokazano na rysunku 9.2. Liczba potrzebnych zespołów jest różna w każdym projekcie, ale dopóki jest stosowane podejście przyrostowe, nie powinniśmy potrzebować więcej niż trzy lub cztery małe zespoły. Gdy skończy się dany etap, możemy przydzielić zespołom kolejne zadania.
- *Administrator bazy danych.* Większość pracy, jaką wykonujemy przy tworzeniu hurtowni danych, wymaga korzystania z bazy danych. Z tego powodu w naszym projekcie potrzebujemy dobrego DBA, który będzie z nami współpracował. Jeżeli jest to możliwe, powinien to być administrator z doświadczeniem w zakresie hurtowni danych. Projekt hurtowni danych będzie wykonywany przez jeden z zespołów projektowych, ale powinien on ściśle współpracować z DBA, którego dobra znajomość hurtowni danych jest bardzo pożądana.

Podobnie zespół ekstrakcji i ładowania może skorzystać ze współpracy z DBA, który rozumie problemy związane z ładowaniem danych występujące w hurtowniach danych.

- *Administrator systemu.* Jest to ekspert w dziedzinie systemów operacyjnych i infrastruktury. Osoba ta przydziela odpowiednie prawa dostępu dla komputerów programistów i dba o środowisko pracy zespołu. Dobrze jest jednak, jeżeli możemy polegać na kimś, kto może pomóc przy bardziej technicznych elementach. Konieczne jest skonfigurowanie użycia procesora oraz pamięci w optymalny sposób, a dodatkowo mirrorowania dysków, kontrolerów cache itp. Te elementy muszą być wykonane w pewnym momencie, gdy przejdziemy do dostarczania wydajności, więc dużą zaletą jest dysponowanie w zespole kimś, kto potrafi się tym zająć.

Podział struktury zadań w hurtowni danych

W tym podrozdziale przedstawię ogólną strukturę podziału danych (WBS) dla projektów hurtowni danych. Nasz WBS ma sporo powyżej 100 elementów i zawiera również zależności pomiędzy elementami. Jest dosyć wyczerpujący, ale jeżeli Czytelnik uważa, że powinien coś do niego dodać lub go zmodyfikować, proszę bardzo!

Przedstawiłem WBS w logicznych częściach, przez co niektóre elementy mogą nie być natychmiast oczywiste i będą wymagać wyjaśnienia. Zwróćmy uwagę na kolumnę po lewej stronie w tabeli 9.1, „Etap projektu”. Ten WBS został zaprojektowany tak, aby mógł obsłużyć podejście przyrostowe. Musimy wypełnić WBS dla każdego przyrostu naszej hurtowni danych.

Pierwszą część stanowi model koncepcyjny. W modelu koncepcyjnym dodaliśmy wszystkie komponenty, które zidentyfikowaliśmy w rozdziale na temat modelowania koncepcyjnego. Zwróćmy uwagę, że nie ma potrzeby korzystania z techniki modelowania kropki. Ten WBS nie jest zależny od żadnej metodologii, choć dosyć dobrze pasuje do metodologii kropki.

Tabela 9.1. Zadania modelowania koncepcyjnego

Numer etapu projektu	Numer WBS	Opis zadania	Zależności	Osobodni
Model koncepcyjny				
	CM010	Zbieranie wymagań biznesowych		
	CM020	Tworzenie koncepcyjnego modelu wielowymiarowego	CM010	
	CM030	Określenie wymagań retrospekcji	CM020	
	CM040	Określenie źródeł danych	CM020	
	CM050	Określenie przekształceń	CM040	
	CM060	Określenie zależności w zmienionych danych	CM040	
	CM070	Określenie częstotliwości i opóźnień czasowych	CM040	
	CM080	Tworzenie metadanych	CM040	

Retrospekcja, zależności zmienionych danych oraz częstotliwości i opóźnienia czasowe powinny być już znanymi terminami.

Następnym krokiem jest konwersja modelu koncepcyjnego na model logiczny (tabela 9.2). Jest to opisane w rozdziale 6.

Tabela 9.2. Zadania modelowania logicznego

Numer etapu projektu	Numer WBS	Opis zadania	Zależności	Osobodni
Model logiczny				
	LM010	Przekształcanie modelu koncepcyjnego w logiczny	CM030	
	LM020	Projektowanie modelu bezpieczeństwa	CM020	
	LM030	Tworzenie logicznego modelu danych	LM010	

Tworzenie modelu fizycznego jest opisane szczegółowo w rozdziale 7. Wynik z fazy modelu fizycznego jest zbiorem instrukcji języka definicji danych (DDL) wymaganych do utworzenia wszystkich obiektów hurtowni danych (tabel, indeksów itp.) (tabela 9.3).

Tabela 9.3. Zadania modelowania fizycznego

Numer etapu projektu	Numer WBS	Opis zadania	Zależności	Osobodni
Model fizyczny				
	PM010	Przekształcanie modelu logicznego w fizyczny	LM030	
	PM020	Tworzenie modelu bezpieczeństwa	LM020	
	PM030	Projektowanie modelu fizycznego	PM010	
	PM040	Tworzenie DDL	PM030	

Większość z kolejnych części projektu może być realizowana równolegle z modelowaniem danych. Modelowanie logiczne i fizyczne będzie realizowane przez zespół hurtowni danych, a przechwytywanie danych z systemów źródłowych przez zespół ekstrakcji. Występują tu pewne zależności od modelu koncepcyjnego w zakresie metadanych, które będą kierowały zespół ekstrakcji i ładowania do odpowiednich systemów źródłowych i skojarzonych z nimi elementów danych. W tabeli 9.4 wymienione są zadania wymagane w *początkowym* ładowaniu wielowymiarowych modeli zachowań. Należy pamiętać, że ładowanie to jest wykonywane tylko raz. W przypadku tego typu elementów systemu można zastosować mniej rygorystyczne podejście niż w odniesieniu do przyrostowych operacji ekstrakcji i ładowania.

Zwróćmy uwagę, że procesy od ID010 do ID120 są powtarzane dla każdej encji.

Następny zbiór procesów wydaje się bardzo podobny i w pewnym sensie taki jest. Jednak istnieje znacząca różnica, ponieważ te procesy będą wykorzystywane jako część gotowego, dostarczonego systemu i muszą być przygotowane z jakością przemysłową. Każdy proces powinien rejestrować swoje działanie w dzienniku systemu. Informacje zapisywane w dzienniku powinny być oznaczone jako:

- *Informacja*. Komunikaty informacyjne zawierają datę i czas rozpoczęcia oraz zakończenia. Czas realizacji jest kolejną użyteczną informacją, która pozwala administratorowi systemu na określenie najbardziej czasochłonnych procesów systemu. Bardzo przydatną informacją jest

Tabela 9.4. Początkowe zadania przechwytywania i ładowania danych

Numer etapu projektu	Numer WBS	Opis zadania	Zależności	Osobodni
Początkowe przechwytywanie i ładowanie danych				
Dane zachowań				
IB010		Proces projektowania ekstrakcji danych	CM040	
IB020		Proces tworzenia ekstrakcji danych	IB010	
IB030		Proces testowania ekstrakcji danych	IB020	
IB040		Projektowanie procesu VIM	CM050	
IB050		Tworzenie procesu VIM	IB040	
IB060		Testowanie procesu VIM	IB050	
IB070		Proces projektowania ładowania danych	PM030	
IB080		Proces tworzenia ładowania danych	IB070	
IB090		Proces testowania ładowania danych	IB080	
IB100		Wykonanie ładowania danych faktów	IB090	
IB110		Weryfikacja dokładności i spójności danych	IB100	
IB120		Tworzenie metadanych	CM080	
Dane okoliczności i wymiarów				
ID010		Proces projektowania ekstrakcji danych	CM040	
ID020		Proces tworzenia ekstrakcji danych	ID010	
ID030		Proces testowania ekstrakcji danych	ID020	
ID040		Projektowanie procesu VIM	CM050	
ID050		Tworzenie procesu VIM	ID040	
ID060		Testowanie procesu VIM	ID050	
ID070		Proces projektowania ładowania danych	PM030	
ID080		Proces tworzenia ładowania danych	ID070	
ID090		Proces testowania ładowania danych	ID080	
ID100		Wykonanie ładowania wymiarów	ID090	
ID110		Weryfikacja dokładności i spójności danych	ID100	
ID120		Tworzenie metadanych	CM080	

również liczba przetworzonych rekordów, a jeżeli jest to możliwe, także wartość z przetworzonych rekordów. Pozwala to utworzyć pewnego rodzaju zapis śladu przepływu danych z systemu źródłowego do hurtowni i jest nieocenione przy śledzeniu błędów. Innym dobrym pomysłem jest rejestrowanie wpisów do dziennika w czasie działania procesu, a nie na jego końcu. Jeżeli proces rutynowo obsługuje miliony rekordów, to przydatne jest zapisywanie danych do dziennika co 100 000 rekordów. Jednym ze sposobów na zapewnienie bardziej dynamicznego działania jest utworzenie argumentu wykonania definiującego częstotliwość zapisu danych do dziennika.

- *Ostrzeżenie.* Proces powinien dodawać ostrzeżenie do pliku dziennika, gdy zostanie wykryte coś podejrzanego, ale nie ma powodu, aby natychmiast na to reagować. Wygenerowanie ostrzeżenia może wystąpić wskutek odrzucenia rekordu. Czasami zapis w dzienniku może być informacyjny, na przykład gdy system jest w 10 procentach pełny, w 20 procentach pełny itd. Te komunikaty informacyjne mogą być „wypromowane” do ostrzeżeń, jeżeli system plików jest zajęty w ponad 60 procentach. W systemie proaktywnym może to spowodować wysłanie komunikatu do operatora, aby mógł on podjąć działania, zanim sytuacja stanie się krytyczna.
- *Błąd.* Gdy wystąpi błąd, proces nie może dalej działać i musi zostać zatrzymany. Gdy na przykład zostanie zapełniony system plików, to pomimo że proces może uzyskać dostęp do innego systemu, i tak nie może kontynuować pracy i musi się zatrzymać. Czasami błędy są wykrywane od razu, na początku, na przykład jeżeli proces ustali, że otrzymał plik danych, które już wcześniej przetworzył. Błędy zwykle powodują zatrzymanie systemu i aby mógł dalej pracować, wymagana jest interwencja.

Gdy wystąpi błąd, istnieje naturalna pokusa „spróbowania ponownie”. Rozumiemy przez to ponowne uruchomienie procesu. W odniesieniu do relacyjnych baz danych zwykle polega to na wycofaniu wszystkich operacji, które zostały wykonane prawidłowo, a następnie, po rozwiązaniu problemu, na ponownym realizowaniu zadań. Niestety, w przypadku hurtowni danych będziemy musieli przetworzyć wiele milionów transakcji. Nierzadko samo wycofanie zmian zajmuje dużo czasu. Zwykle wycofywanie wszystkich transakcji jest niepotrzebne, szczególnie jeżeli po prostu zabrakło nam miejsca — co jest bardzo częstym problemem w hurtowniach. Jeżeli przetworzymy połowę danych, to w przypadku konieczności wycofania zmian i ponownego przetworzenia danych od początku zadanie potrwa dwa razy dłużej, niż początkowo oczekiwaliśmy. Gdy działamy w ciasnym serwisowym oknie czasowym, to w efekcie może to spowodować, że nie będziemy mogli rano otworzyć hurtowni danych. Należy pamiętać, że część tych procesów może nawet w normalnych okolicznościach zająć kilka godzin. Warto zastanowić się, czy nie można pozwolić na kontynuowanie procesu. Oznacza to, że choć proces został zatrzymany po wykryciu błędu, to może być wznowiony po rozwiązaniu problemu.

Podobnie jak w przypadku początkowego ładowania danych, zadania związane z ciągłym ładowaniem wymiarów, od SD010 do SD1100 w tabeli 9.5, muszą być powtórzone dla każdej encji.

Następny zbiór zadań jest związany z aktywowaniem użytkowników. Operacji tu wymienionych nie trzeba objaśniać. Dodatkowo po zdefiniowaniu ról klienta zadania te muszą być wykonywane równolegle z pozostałymi (tabela 9.6).

Następną operacją jest zorganizowanie zadań administratora hurtowni danych (tabela 9.7). Obejmuje to definicje ról użytkowników, schematy użytkowników (perspektywy danych użytkowników), jak również zidentyfikowane tabele podsumowań. Jako naczelną zasadę zalecam, aby wstrzymać tworzenie tabel podsumowań na tym etapie. Powody tego są następujące:

- (1) Ich tworzenie powoduje opóźnienie momentu, w którym można udostępnić system użytkownikom. Dosyć często proces sumowania pochłania znaczną część zadań programowania. Każda odpowiedź, do której sformułowania były użyte tabele na poziomie podsumowań, może być również utworzona na podstawie tabeli szczegółów, ponieważ tabele podsumowań pobierają swoje dane ze szczegółów. Jest to wyłącznie kwestia wydajności. Całkowicie akceptowane jest najpierw udostępnienie systemu użytkownikom, a dopiero później utworzenie mechanizmów poprawiających wydajność.

Tabela 9.5. *Ciągłe zadania przechwytywania i ładowania danych*

Numer etapu projektu	Numer WBS	Opis zadania	Zależności	Osobodni
Ciągłe przechwytywanie i ładowanie danych				
Dane zachowań				
SB010		Identyfikacja punktu przechwycenia w cyklu życia encji	CM040	
SB020		Proces projektowania ekstrakcji danych	CM040	
SB030		Proces tworzenia ekstrakcji danych	SB020	
SB040		Proces testowania ekstrakcji danych	SB030	
SB050		Projektowanie procesu VIM	Powtórzyć dla każdej encji	CM050
SB060		Tworzenie procesu VIM		SB050
SB070		Testowanie procesu VIM		SB060
SB080		Proces projektowania ładowania danych	PM030	
SB090		Proces tworzenia ładowania danych	SB080	
SB100		Proces testowania ładowania danych	SB090	
Dane okoliczności i wymiarów				
SD010		Identyfikowanie zmienionych danych	CM040	
SD020		Tworzenie połączeń zależności	CM060	
SD030		Proces projektowania ekstrakcji danych	CM040	
SD040		Proces tworzenia ekstrakcji danych	SD030	
SD050		Proces testowania ekstrakcji danych	Powtórzyć dla każdej encji	SD040
SD060		Projektowanie procesu VIM		CM050
SD070		Tworzenie procesu VIM		SD060
SD080		Testowanie procesu VIM	SD070	
SD090		Proces projektowania ładowania danych	PM030	
SD100		Proces tworzenia ładowania danych	SD090	
SD110		Proces testowania ładowania danych	SD100	

Tabela 9.6. *Zadania dostępu użytkowników*

Numer etapu projektu	Numer WBS	Opis zadania	Zależności	Osobodni
Dostęp użytkowników				
UA010		Przydział użytkowników do ról	WA010	
UA020		Rejestrowanie użytkowników	UA010	
UA030		Konfigurowanie oprogramowania użytkowników		
UA040		Testowanie dostępu użytkowników	UA030	
UA050		Projektowanie podręcznika użytkownika	UA030	

Tabela 9.6. *Zadania dostępu użytkowników — ciąg dalszy*

Numer etapu projektu	Numer WBS	Opis zadania	Zależności	Osobodni
Dostęp użytkowników				
	UA060	Tworzenie podręcznika użytkownika	UA050	
	UA070	Projektowanie pomocy dla procesów	UA030	
	UA080	Tworzenie pomocy dla procesów	UA070	
	UA090	Testowanie pomocy dla procesów	UA080	

Tabela 9.7. *Zadania administrowania zadaniami przetwarzania w hurtowni danych*

Numer etapu projektu	Numer WBS	Opis zadania	Zależności	Osobodni
Administrowanie hurtownią danych				
	WA010	Definiowanie ról użytkowników	LM020	
	WA020	Odwzorowanie modelu bezpieczeństwa na role	WA010	
	WA030	Projektowanie schematów użytkownika	WA020	
	WA040	Tworzenie schematów użytkownika	WA030	
	WA050	Testowanie schematów użytkownika	WA040	
	WA060	Projektowanie procesu sumowania danych	PM030	
	WA070	Tworzenie procesu sumowania danych	WA060	
	WA080	Testowanie procesu sumowania danych	WA070	
	WA090	Projektowanie procesu sumowania danych nawigacji	WA060	
	WA100	Tworzenie procesu sumowania danych nawigacji	WA090	
	WA110	Testowanie procesu sumowania danych nawigacji	WA100	

- (2) Udostępnienie użytkownikom danych możliwie szybko ma również inne praktyczne zalety. Możemy obserwować ich wykorzystanie i użycie tej wiedzy do określenia, które tabele podsumowań powinniśmy zainstalować. W projektach hurtowni danych często się zdarza, że tabele podsumowań muszą być przeanalizowane dwukrotnie. Bywa, że trafiamy za pierwszym razem, ale czasami się mylimy i niektóre z naszych podsumowań nie są nigdy wykorzystywane przez użytkowników.

Następnym krokiem są zadania automatyzacji (tabela 9.8). Wielu kierowników projektów będzie udowadniać, że jest to największe zagrożenie przy budowaniu hurtowni danych. Nasze podejście przyrostowe przy wszystkich swoich zaletach często pogarsza problem. Powoduje ono, że po pierwszym przyroście użytkowników zaczynają korzystać z danych. Jest to ten moment, kiedy mogą się oni przekonać do rozwiązania i stwierdzić: „Spójrzcie, naprawdę są pewne korzyści ze wszystkich tych danych!”. Gdy to się stanie, nie będą mieli dosyć i będą chcieli więcej — nie tylko modyfikacji tu czy tam — będą chcieli znacznie więcej. Zanim będziemy wiedzieli, gdzie jesteśmy, część zespołów zostanie zaangażowana do drugiego etapu. Jednak

Tabela 9.8. *Zadania automatyzacji*

Numer etapu projektu	Numer WBS	Opis zadania	Zależności	Osobodni
Automatyzacja przetwarzania				
	AP010	Projektowanie procesu automatyzacji		
	AP020	Tworzenie procesu automatyzacji	AP010	
	AP030	Testowanie procesu automatyzacji	AP020	
	AP040	Wykonanie testów integracyjnych	AP030	
	AP050	Projektowanie podręcznika procedur operacyjnych	AP010	
	AP060	Tworzenie podręcznika procedur operacyjnych	AP050	
	AP070	Uzyskanie akceptacji operacyjnej	AP060	

klient nie wie, że pierwszy etap jest daleki od zakończenia. Udało się nam tylko pokazać dane, ale wszyscy będą uważać, że już skończyliśmy. Użytkownicy nie będą widzieli tego, że aby pobrać dane z systemów źródłowych, przetworzyć je i załadować do hurtowni danych, trzeba wykonać żmudną ręczną pracę. Żaden z procesów nie został jeszcze zintegrowany i systemy jeszcze ze sobą nie współpracują. Musimy więc wrócić do pierwszego etapu i zrealizować procesy harmonogramu, sterowania, obsługi błędów itd., które są potrzebne, aby zmienić zwykły system w produkt jakości przemysłowej z mocnymi spawami oraz śrubami i nakrętkami zamiast tych wszystkich sznurków i taśmy klejącej.

Jest to poważny problem w starej sztuce określania oczekiwań. Jeżeli pozwolimy, aby się to wymknęło spod naszej kontroli, możemy nigdy jej nie odzyskać. Działy operacyjne danej firmy nigdy nie zaakceptują systemu, który nie potrafi samodzielnie stać na własnych nogach, szczególnie jeżeli jest tak duży i zasobochłonny jak hurtownia danych. Za każdym razem musimy przypominać naszemu klientowi, że to, co widzi, nie jest ukończonym produktem i że do zakończenia pracy potrzeba jeszcze sporo czasu. Istnieje również pogląd, aby nie dawać użytkownikom dostępu do hurtowni danych do momentu zakończenia pierwszego etapu. Choć niektórzy podzielają to twierdzenie, osobiście nadal podtrzymuję opinię, że użytkownicy powinni mieć dostęp do hurtowni tak szybko, jak jest to możliwe. W końcu są to ich dane, a nie nasze. Silne zarządzanie projektem i równie silne zarządzanie oczekiwaniami są tu kluczem do sukcesu.

Przejdźmy teraz do wsparcia. Wsparcie istnieje na kilku poziomach (tabela 9.9). Każdy użytkownik ma specyficzne wymagania. Występują dwa główne rodzaje wsparcia:

- (1) *Wsparcie użytkowników.* W tym przypadku polega to na prowadzeniu telefonicznej linii pomocy lub działu wsparcia użytkowników, który im pomaga, gdy mają pytania na temat systemu lub usługi albo kiedy mają problemy.
- (2) *Wsparcie systemu.* Oczywiście jest, że jako dostawcy systemu musimy wziąć pewną odpowiedzialność za jego obsługę, co najmniej przez jakiś czas. Po tym czasie musi być zapewnione dalsze wsparcie. Obejmuje to rutynową konserwację systemu, w tym aktualizacje oprogramowania systemowego i aplikacyjnego. Konieczne jest również wsparcie ogólnej natury, na przykład przy rozbudowie przestrzeni dyskowej czy też odtwarzaniu systemu po awarii sprzętowej lub programowej.

Tabela 9.9. *Zadania wsparcia użytkowników*

Numer etapu projektu	Numer WBS	Opis zadania	Zależności	Osobodni
Wsparcie użytkowników				
	US010	Projektowanie procesów wsparcia		
	US020	Tworzenie procesów wsparcia	US010	
	US030	Testowanie procesów wsparcia	US020	

W zależności od klienta zmieniają się wymagane poziomy wsparcia, jak również sposoby, w jakie to wsparcie jest realizowane. Niektóre organizacje mają własne oddziały pomocy technicznej, natomiast inne zlecają te zadania innym firmom. Niezależnie od sytuacji musimy zaprojektować strategię wsparcia, aby była ona zgodna z wymaganiami.

Kolejną pułapką jest niezapewnienie użytkownikom wsparcia od samego początku. Często są to użytkownicy, z którymi warto współpracować, ponieważ są oni w stanie powiedzieć nam precyzyjnie, czego potrzebują. Musimy dostarczyć im tych mechanizmów, zanim przejmą od nas odpowiedzialność za system. W wielu przypadkach czas i pieniądze przeznaczone na projekt są tracone przez to, że nie zwrócono uwagi na wymagania wsparcia. Jest to nieco podobne do problemu z automatyzacją, gdyż jest pozostawiane nietknięte do końca projektu i zwykle uważane za jedno z najnudniejszych zadań, choć w rzeczywistości powinien mu być nadany strategiczny priorytet.

Problem kopii zapasowej i wycofywania przedstawiłem już w rozdziale 7. Warto powtórzyć, że tworzenie kopii zapasowej w hurtowni danych nie jest prostym zadaniem i w uzyskanie efektywnego rozwiązania trzeba włożyć sporo pracy (tabela 9.10). Występuje również problem wycofania. W jaki sposób możemy wycofać dane z systemu, gdy nie powinny one wcale trafić do hurtowni danych?

Tabela 9.10. *Zadania kopii zapasowej i odtwarzania*

Numer etapu projektu	Numer WBS	Opis zadania	Zależności	Osobodni
Kopie bezpieczeństwa i odtwarzanie				
	BR010	Projektowanie procesu tworzenia kopii zapasowej		
	BR020	Tworzenie procesu tworzenia kopii zapasowej	BR010	
	BR030	Testowanie procesu tworzenia kopii zapasowej	BR020	
	BR040	Projektowanie procesu wycofywania danych		
	BR050	Tworzenie procesu wycofywania danych	BR040	
	BR060	Testowanie procesu wycofywania danych	BR050	
	BR070	Projektowanie procesu ponownego wprowadzania danych		
	BR080	Tworzenie procesu ponownego wprowadzania danych	BR070	
	BR090	Testowanie procesu ponownego wprowadzania danych	BR080	

Czym jest „ponowne wprowadzanie danych”? Jest to przeciwieństwo wycofywania danych. Czasami okazuje się, że do hurtowni danych zostały załadowane nieprawidłowe dane. Zwykle się dowiadujemy o tym, gdy ktoś z zespołu operacyjnego przyjdzie do naszego biura i powie, że jeden z plików, jaki dostaliśmy cztery dni temu, był nieprawidłowy. Musimy wtedy użyć procedury wycofania w celu usunięcia nieprawidłowych danych, a następnie procedury ponownego wprowadzenia danych, aby umieścić w hurtowni prawidłowe dane (nie zapominajmy, że te nowe dane również mogą być później wycofane, jeżeli okażą się nieprawidłowe — to się zdarza).

Aby hurtownia danych mogła normalnie działać, większość rutynowych procesów musi być wykonywana automatycznie. Wykonywanie procesów oraz ustawianie ich w czasie i określanie zależności między nimi zwykle realizuje się przez mechanizm harmonogramowania. Harmonogramy mają bardzo różne zestawy dostępnych funkcji, ale większość z nich jest dosyć zaawansowana. Na przykład może się zdarzyć, że nie chcemy, aby proces ekstrakcji danych był uruchamiany wcześniej, niż zakończy się proces tworzenia pliku potrzebnego mu do pracy. Choć plik ten może być dostępny, założymy, o siódmej po południu, to jednak nie ma gwarancji, że się tak stanie. Jeżeli o umówionej godzinie plik nie będzie dostępny, nie możemy uruchomić procesu ekstrakcji. Harmonogramy mogą być konfigurowane tak, aby reagowały na istnienie lub brak plików, jak również by rozumiały kody zwracane przez inne procesy, informujące o ich udanym wykonaniu lub błędzie.

Zespół operacyjny nie przejmie odpowiedzialności za działanie systemu, jeżeli nie będzie wiedział, jak go obsługiwać. Jest to problem podobny do aspektów związanych ze wsparciem i automatyzacją; to kolejna znana pułapka, nie tylko w projektach hurtowni danych, ale we wszystkich projektach IT. Tak jak w przypadku zespołu wsparcia, zespół operacyjny będzie miał własny zestaw wymagań, często spisanych, które muszą być spełnione przed przekazaniem mu systemu.

Kolejnymi elementami, jakie musimy dostarczyć, są monitorowanie i dostrajanie wydajności oraz planowanie pojemności. Należy zarezerwować nieco czasu na tego rodzaju działania. Nie ma sensu, aby dostarczać system, który jednego dnia działa doskonale, a w następnym zaczyna mu brakować pamięci, cykli procesora, przestrzeni dyskowych lub pasma sieciowego (tabela 9.11).

Tabela 9.11. *Zadania zarządzania systemem*

Numer etapu projektu	Numer WBS	Opis zadania	Zależności	Osobodni
Zarządzanie systemem				
	SM010	Planowanie		
	SM020	Szkolenie zespołów operacyjnych		
	SM030	Monitorowanie wydajności		
	SM040	Planowanie pojemności		

W tabeli 9.12 wymieniono elementy, które czasami są zapominane lub są tyle razy odkładane na później, że stają się opóźnione. Inną pułapką jest niedostarczenie użytkownikom możliwie dokładnej specyfikacji. Nie ma znaczenia, jak dobrze jest skonfigurowany serwer ani jak dużo ma dostępnej przestrzeni. Wiele świetnie zaprojektowanych, dobrze skonfigurowanych i doskonale działających hurtowni danych zostało wyłączonych przez negatywne przyjęcie, spowodowane przez próby wcześniejszego wciśnięcia komponentów użytkownika końcowego do przestarzałych komputerów klienckich.

Tabela 9.12. Zadania instalacji i wdrożenia

Numer etapu projektu	Numer WBS	Opis zadania	Zależności	Osobodni
Instalacja i wdrożenie				
	IR010	Przeprowadzenie szkolenia użytkowników		
	IR020	Instalowanie komponentów użytkownika		
	IR030	Testowanie konfiguracji oprogramowania użytkowników	UA030	

Tabela 9.13 zawiera kilka zadań, jakie należy wykonać, zanim zespoły programistyczne rozpoczną swoją pracę.

Tabela 9.13. Zadania konfiguracji początkowej

Początkowe wymagania systemowe
Pozyskanie i zainstalowanie oprogramowania DBMS
Pozyskanie i zainstalowanie oprogramowania dostępu użytkowników
Pozyskanie i zainstalowanie oprogramowania repozytorium metadanych
Pozyskanie i zainstalowanie oprogramowania nawigacji zagregowanej
Pozyskanie i zainstalowanie oprogramowania szybkiego sortowania
Pozyskanie i zainstalowanie systemów kopii zapasowej
Pozyskanie i zainstalowanie serwerów WWW i oprogramowania klienckiego
Pozyskanie i zainstalowanie oprogramowania planowania pojemności

Podsumowanie

Musimy kontrolować postęp projektu — w tym rozdziale przedstawiłem dokładną strukturę podziału zadań wraz z zależnościami, które mogą być przystosowane do potrzeb dowolnych projektów.

Omówiłem również zagadnienie dostarczania oprogramowania. Jest to znany problem w hurtowniach danych, ponieważ ich tworzenie różni się nieco od tworzenia tradycyjnego oprogramowania.

Opisałem też zespół projektowy oraz umiejętności wymagane do udanego zrealizowania projektu hurtowni danych.

Ponadto przeanalizowałem niektóre z pułapek związanych z jakością danych i ich dostępnością, a także z komunikacją z klientem. Innym problemem jest na przykład potrzeba współpracy z zespołem operacyjnym oraz pomocy technicznej po stronie klienta.

Mam nadzieję, że rozdział ten będzie wartościowym przewodnikiem dla kierowników projektów zaangażowanych w dostarczanie hurtowni danych.

Skorowidz

A

administrator systemu, 217, 220
alert, 247
analiza wymiarów, 38, 39, 59
architektura
 EASI, 175, 197, 200
 OLAP, *Patrz* OLAP
architektura systemu, *Patrz* specyfikacja systemu
archiwizacja, *Patrz* dane archiwizacja
atrybut
 finansowy, 281
 gospodarstwa domowego, 275
 identyfikujący, 116
 istnienia, 151, 152, 154, 164, 171, 269
 klienta, 275
 osobisty, 275
 zachowania, 278
 zainteresowań i hobby, 283
 zatrudnienia, 282

C

cele biznesowe, 16
CRM, *Patrz* zarządzanie relacjami z klientem
czas
 opóźnienia, 167
 prawidłowy, 82, 83, 93, 106, 119
 rozdzielczość, 114, 119, 146, 168
 rozwiązania problemów, 94
 transakcji, 82, 83, 93, 106, 119, 167
 wymiar, 90, 157, 164
 znacznik końca, 154
 znakowanie, *Patrz* znakowanie czasem

D

dane
 aktualność, 181
 archiwizacja, 194
 błędne, 179, 181, 220, 229
 cykl życia, 103
 czasu, 81
 ekstrakcja, 194, 228
 integracja, 38, 46, 47, 90
 jakość, 213, 234
 kontrola poprawności, *Patrz* VIM
 kopia okoliczności, 194
 kopia zapasowa, 194
 kopia zmian, 194
 ładowanie, 194, 228, 229
 niestrukturalizowane, 248
 nieulotność, 38
 normalizacja, 62
 odrzućenie, 177, 178, 179
 okoliczności, 83
 operacyjne, 35
 przechwytywanie zmian, 166
 przekształcanie, 228, 229
 pula, 188, 192, 193
 referencyjne, 25, 83
 sortowanie, 242
 spójność, 181
 strategiczne, 35
 strukturalizowane, 248
 udostępnianie, 222
 wartość domyślna, 177, 182
 wycofywanie, 195, 224
 wydobywanie, *Patrz* wydobywanie danych
 wymiarów, 96
 zachowań, 83, 193, 194
 zewnętrzne, 247
 zmiennność w czasie, 38

Database Management System, *Patrz* system zarządzania bazami danych
 DBMS, *Patrz* system zarządzania bazami danych
 DDL, *Patrz* język definicji danych
 Decision Support Systems, *Patrz* DSS
 diagram ERD, 76
 diagram stanów, 44
 domena, 151
 dominacja na rynku produktu, 17, 18
 doskonałość operacyjna, 18
 DSS, 31, 33, 247, 249, *Patrz* ograniczenia, *Patrz* zadania

E

encja, 33, 41, 44, 45, 48, 73, 83, 86, 87, 89, 91, 100, 102, 107, 115, 122, 136, 150, 161, 192, 218
 referencyjna, 92
 zasada integralności, *Patrz* zasada integralności encji
 ETL, 228, 229, 249
 Extensible Markup Language, *Patrz* XML

F

fakt, *Patrz* zdarzenie
 First Manhattan Group, 19

G

GCM, *Patrz* model koncepcyjny

H

hierarchia, 100, 107, 109, 123, 133
 dekompozycja, 164

I

indukcja reguł, 237
 istnienie, 151, 153, 169
 atrybuty, *Patrz* atrybut istnienia
 nieciągłe, 165

J

JAD, *Patrz* warsztat wspólnego tworzenia aplikacji
 język definicji danych, 75, 218, 232

K

kampania, 27, 28
 jednofazowa, 27, 239
 powtarzalna, 28, 239
 wielofazowa, 27, 239
 zarządzanie, 238

Kimball Ralph, 95, 97, 103, 157
 klient
 identyfikacja, 28, 153
 identyfikator, 214
 lojalność, 22
 migracja, 22, 26, 27, 153, 157, 192, 201
 okoliczności, 70, 71, 192
 profil, 24, 202
 segmentacja, 74, *Patrz* segmentacja wartość, 26
 wymiar, 72, 164
 zachowania, 70, 71
 zarządzanie relacjami, *Patrz* zarządzanie relacjami z klientem

klucz
 generalizowany, 96
 główny, 96
 obcy, 87
 produkcyjny, 98
 techniczny, 96
 kluczowe wskaźniki wydajności, 112
 kolumna kluczy obcych, *Patrz* klucz obcy
 retrospekcja, 114
 koncentracja na kliencie, 17
 kontrola poprawności danych, *Patrz* VIM
 kopia okoliczności, *Patrz* dane kopia okoliczności
 kopia zapasowa, 224, *Patrz także* dane kopia zapasowa
 kopia zmian, *Patrz* dane kopia zmian

M

margines błędu, 186
 marketing
 agresywny, 27
 defensywny, 27
 osobisty, 20
 personalizowany, 29
 metadane, 54, 122, 136, 137, 140, 141, 143, 176, 181, 183, 193, 228, 241, 258
 aktywne, 242
 nawigacji, 242
 pasywne, 242
 przekształceń, 241
 metoda
 szybkiego tworzenia aplikacji, 199, 210, 211
 wodospadu, 14, 199, 210
 migracja klientów, *Patrz* klient migracja
 model
 fizyczny, 75, 76, 163
 GCM, 188
 implementacji, *Patrz* model fizyczny
 komponent, 114
 koncepcyjny, 75, 76, 79, 95, 113, 119, 147, 149, 161, 200, 217
 wymagania, 111

korporacyjny, 84
 logiczny, 75, 76, 95, 112, 124, 161, 171, 217
 ograniczenia, 167, 168, 169, 170
 normalizowany, 64
 relacyjny, 42, 48
 trzeciej postaci normalnej, 62
 wielowymiarowy, 61, 64, 70, 79, 112, 123, 171, 230
 wymiarów, 48
 zorientowany na klienta, 72
 modelowanie
 konceptyjne, 111
 metodą kropki, 119, 123, 124, 132, 147, 200,
 202, 255
 modelowanie wymiarów, 42

N

nawigator podsumowań, 54, 55, 56
 numer przetwarzania, 196

O

ograniczenie
 podwójnego zliczania, 167
 spójności odwołań, 169
 usuwania, 170
 okoliczności, 92, 114, 195
 kopia, *Patrz* dane kopia okoliczności
 OLAP, 230, 246

P

partycjonowanie, 193, 194
 personalizacja, 240
 pierwsza postać normalna, 62
 podejście wzrastające, 174, 200
 podręcznik systemu, *Patrz* specyfikacja systemu
 postać normalna Boyce-Codda, 62
 przetwarzanie analityczne na bieżąco, *Patrz* OLAP
 przypadkowość, 118
 pula danych, *Patrz* dane pula

R

RAD, *Patrz* metoda szybkiego tworzenia aplikacji
 relacyjna baza danych, 45
 relacyjny system zarządzania bazą danych, 31, 32,
 45, 81
 Relational Database Management System, *Patrz*
 relacyjny system zarządzania bazą danych
 retrospekcja, 114, 116, 143, 144, 147, 150, 154, 161,
 164, 165, 171, 181, 186, 251
 atrybutu, 116
 encji, 115
 ograniczenia, 169

relacji, 115
 wartość, 145
 ryzyko, 213

S

schemat
 gwiazdy, 40, 48, 50, 67, 68, 72, 101, 121, 231
 kropki, 121
 logiczny, 269
 płatka śniegu, 51, 67, 68, 150, 231
 segmentacja, 24, 27
 według danych skojarzonych, 26
 według zachowań, 25
 sieci neuronowe, 237
 silnik wyszukiwania, 248
 specyfikacja projektowa, *Patrz* specyfikacja systemu
 specyfikacja systemu, 14
 SQL, 32, 45, 49, 54, 58, 59, 88, 104, 232, 246
 rozszerzenia temporalne, *Patrz* TSQL2
 Structured Query Language, *Patrz* SQL
 struktura podziału danych, *Patrz* WSB
 system zarządzania bazami danych, *Patrz* relacyjny
 system zarządzania bazą danych

T

tabela
 faktów, 49, 52, 56, 66, 70, 74, 85, 95, 96, 100, 109,
 164
 podsumowań, 195
 wymiarów, 96, 109, 152
 tematyczna hurtownia danych, 68
 TSQL2, 105

U

użytkownik
 aktywowanie, 220
 wsparcie, 223

V

VIM, 176, 183, 184, 188, 195, 211, 229, 249

W

warstwa
 integracji, 183, *Patrz też* VIM
 kontroli poprawności danych, 176, *Patrz też*
 VIM
 odzworowania, 184, *Patrz też* VIM
 warsztat
 analizy komponentów, 138
 strategii informacyjnej, 125, 126, 127, 138
 wspólnego tworzenia aplikacji, 125

wartość domyślna, *Patrz* dane wartość domyślna
wartość marki, 17, 18
WBS, 209, 217
wdrożenie, 205, 206
wspomaganie podejmowania decyzji, *Patrz* DSS
wydajność systemu, 163, 212, 225
wydobywanie danych, 57, 205, 234, 237, 238
wykres sieciowy, 236
wymiar, 122, 132, 150, 192
 czasu, *Patrz* czas wymiar
 hierarchia, 123, 133
 kształt, 113
 powoli zmieniający się, 95

X

XML, 248

Z

zachowanie, 113, 131, 195, 234
zadania
 strategiczne, 35
zależność przechodnia, 64
założenia, 213
zapytanie SQL, *Patrz* SQL
zarządzanie relacjami z klientem, 18, 19, 21, 23, 29,
 69, 109, 147, 193, 199, 227, 238, 245
zasada
 atrybutów identyfikujących, 116
 integralności encji, 62
 przyczyny i efektu, 71
zdarzenie, 91, 97, 107, 113, 122, 150
zespół produkcji oprogramowania, 215
złączenie teta, 164
zmiana przypadkowa, *Patrz* przypadkowość
znakowanie czasem, 103

PROJEKTOWANIE HURTOWNI DANYCH

Wspomaganie zarządzania relacjami z klientami

KANON INFORMATYKI

Zastanawiasz się, jak zmaksymalizować zyski czerpane z właściwego zarządzania relacjami z klientem? Jest to pytanie, które spędza sen z powiek każdemu przedsiębiorcy. Jedną z możliwości jest wykorzystanie odpowiednio zaprojektowanej hurtowni danych. Projekt takiej bazy danych, uwzględniającej potrzeby klienta, wymaga nowych technik i metodologii. Jakich?

Na to pytanie odpowiada Chris Todman w książce, którą trzymasz w rękach. Wiodący konsultant w dziedzinie hurtowni danych przedstawi Ci kompletną metodologię, pozwalającą na zaprojektowanie, wytworzenie i wdrożenie hurtowni danych pod kątem zarządzania relacjami z klientem. Najpierw przeczytasz o kilku zagadnieniach teoretycznych związanych z relacjami z klientem oraz hurtowniami danych. Potem zapoznasz się z typowymi problemami, aby w rozdziale piątym przejść do omówienia modelu koncepcyjnego. Dowiesz się, jak obsługiwać okoliczności, identyfikować zmiany w danych oraz modelować metodą kroпки. Kolejne omawiane zagadnienia to model logiczny i sposoby rozwiązywania problemów wydajnościowych. W rozdziale poświęconym implementacji fizycznej zobaczysz, jak kontrolować poprawność danych, zarządzać kopiami zapasowymi oraz aplikacjami CRM. Ponadto nauczysz się zarządzać projektem, przejrzysz możliwości dostępnego oprogramowania oraz zdobędziesz wiedzę o perspektywach rozwoju. To wszystko znajdziesz w tej długo oczekiwanej książce, poświęconej hurtowniom danych.

Sprawdź, jak zmaksymalizować zyski poprzez właściwe zarządzanie relacjami z klientem!

- Zarządzanie relacjami z klientem
- Wartość marki
- Zarządzanie kampaniami
- Marketing personalizowany
- Budowanie hurtowni danych — komponent ekstrakcji oraz integracji danych
- Baza danych hurtowni — opis encji
- Problemy przy wykonywaniu relacyjnych baz danych
- Problemy z obróbką czasu w hurtowniach danych
- Wybór modelu koncepcyjnego
- Modelowanie metodą kroпки
- Modelowanie logiczne — schemat, wybór rozwiązania, ograniczenia
- Implementacja fizyczna
- Zarządzanie kopiami zapasowymi
- Aplikacje CRM
- Zarządzanie projektem
- Oprogramowanie
- Przyszłość hurtowni danych



Helion

Sprawdź najnowsze promocje
① <http://helion.pl/promocje>
Książki najchętniej czytane
② <http://helion.pl/bestseller>
Zamów informacje o nowościach
③ <http://helion.pl/nowosci>

Helion SA
ul. Koszowska 1c, 44-100 Gliwice
tel. 32 230 88 63
e-mail: helion@helion.pl
<http://helion.pl>

Nr katalogowy: 6610

Księgarnia internetowa:
<http://helion.pl>

Zamówienia telefoniczne:
0 801 339900
0 601 339900

helion.pl
księgarnia
internetowa

Cena 59,00 zł

ISBN 978-83-246-3225-1



9 788324 632251

Informatyka w najlepszym wydaniu