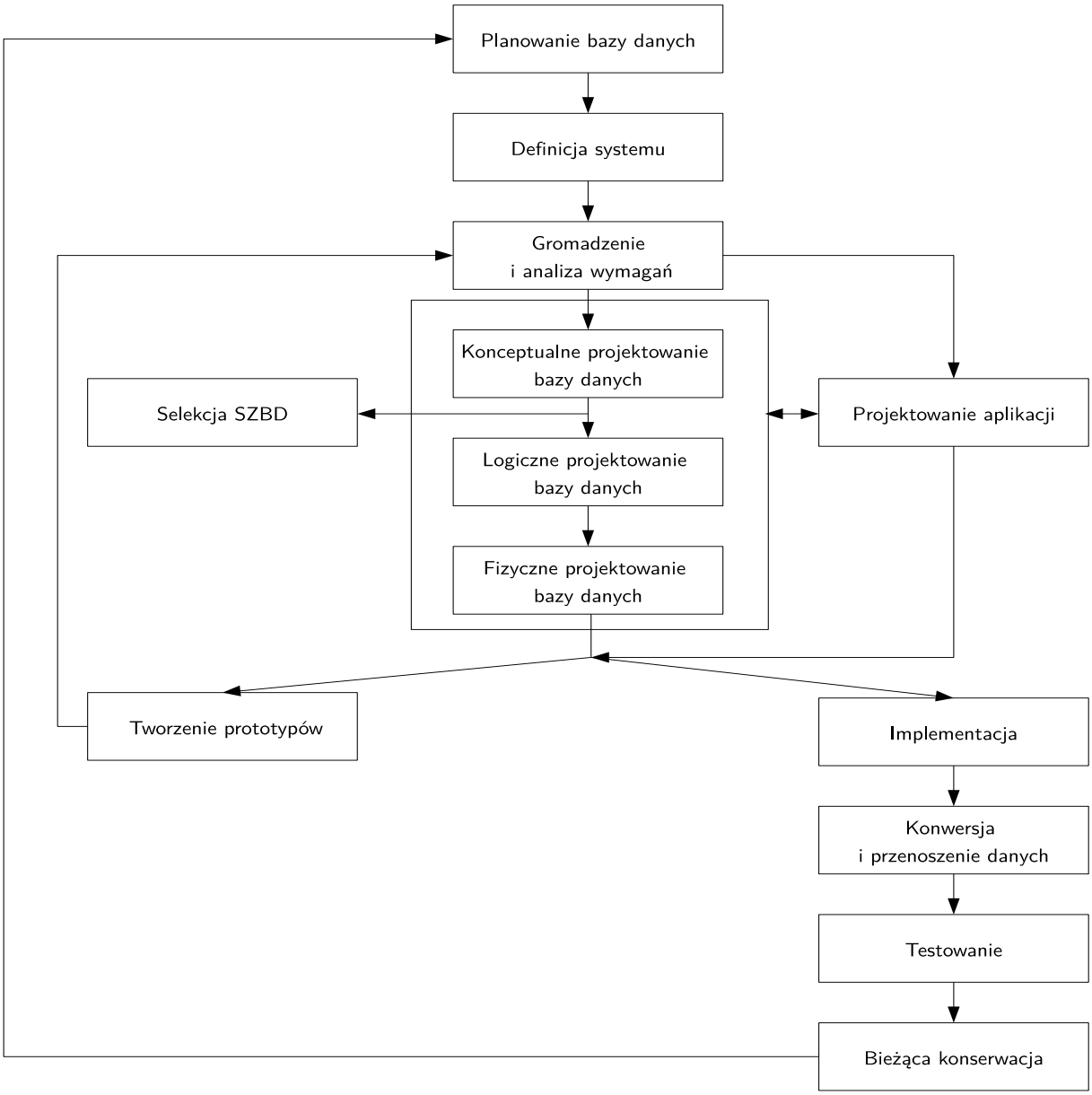
1 Cykl życia aplikacji bazy danych



Źródło:
Thomas Connolly, Carolyn Begg, Systemy baz danych, Wydawnictwo RM, 2004

Zestawienie głównych czynności związanych z każdą fazą cyklu życia aplikacji bazy danych [3]:

- Planowanie bazy danych - planowanie najbardziej skutecznych metod realizacji faz cyklu życia.
- Definicja systemu - określenie zakresu i granic stosowania danej aplikacji bazy danych.
- Gromadzenie i analiza wymagań - zbieranie i analiza wymagań pochodzących od użytkowników i wynikających z obszarów zastosowań.
- Projektowanie bazy danych - projektowanie konceptualne, logiczne i fizyczne bazy danych.

- Selekcja SZBD - wybór SZBD odpowiedniego dla aplikacji bazy danych.
- Projektowanie aplikacji - projektowanie interfejsów użytkownika i programów użytkowych, które będę przetwarzać bazę danych.
- Tworzenie prototypów (opcjonalne) - budowanie działającego modelu aplikacji bazy danych, który pozwala projektantom i użytkownikom zobrazować i ocenić sposób działania i wygląd końcowego systemu.
- Implementacja - tworzenie zewnętrznych, konceptualnych i wewnętrznych definicji bazy danych i programów użytkowych.
- Konwersja i przenoszenie danych - przenoszenie danych ze starego systemu do nowego.
- Testowanie - testowanie i usuwanie błędów z aplikacji bazy danych oraz sprawdzanie zgodności z wymaganiami użytkowników.
- Bieżąca konserwacja - aplikacja bazy danych jest w pełni zaimplementowana. System jest na bieżąco monitorowany i konserwowany. W razie potrzeby do aplikacji baz danych są wprowadzane nowe wymagania poprzez ponowne przejście przez powyższe fazy.

Metodologia projektowania - Kompleksowe podejście wykorzystujące procedury, techniki, narzędzia oraz metody tworzenia dokumentacji służącej realizacji i uproszczeniu procesu projektowania.

Konceptualne projektowanie bazy danych [3] - proces tworzenia modelu dla informacji wykorzystywanych w przedsiębiorstwie, niezależnego od wszelkich szczegółów reprezentacji fizycznej.

2 Model Związków-Encji

Opracowano na podstawie: Thomas Connolly, Carolyn Begg, Systemy baz danych, Wydawnictwo RM, 2004

Model Związków-Encji lub Encja-Związek (ang. Entity-Relationship Model) [3] - Modelowanie związków encji reprezentuje podejście zstępujące do projektowania baz danych. Rozpoczyna się mianowicie od wyodrębnienia istotnych danych, nazywanych *encjami*, i tych *związków* pomiędzy nimi, które powinny być reprezentowane w modelu. Następnie dodaje się do modelu coraz więcej szczegółów, takich jak informacje, które chcemy przechowywać o encjach i związkach, nazywane *atrybutami*, oraz *więzy* (warunki) odnoszące się do encji, związków i atrybutów.

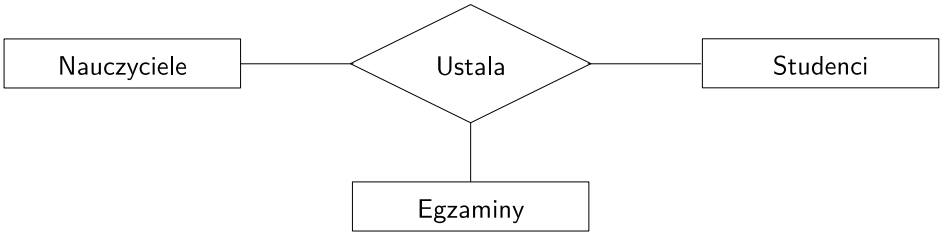
Do zapisu modelu związków encji zastosowany został zunifikowany język modelowania - UML (ang. Unified Modeling Language).

Definicje [3]:

- Zbiór encji - to grupa obiektów o tych samych właściwościach, które w ramach przedsiębiorstwa zostały uznane za niezależne byty.
- Silny zbiór encji - to zbiór encji, którego istnienie nie jest zależne od innych zbiorów encji.
- Słaby zbiór encji - to zbiór encji, którego istnienie zależy od innych zbiorów encji. Cechą charakterystyczną słabego zbioru encji jest brak jednoznacznej identyfikacji każdego wystąpienia encji za pomocą atrybutów przypisanych wyłącznie temu zbiorowi encji.
- Wystąpienie encji - to unikalny i rozpoznawalny obiekt ze zbioru encji.
- Związek - to zbiór znaczących powiązań pomiędzy zbiorami encji.
- Związek rekurencyjny - to związek, w którym ten sam zbiór encji występuje więcej niż jeden raz w różnych rolach np. dla zbioru encji Pracownicy może wystąpić związek rekurencyjny „Ma przełożonego".
- Wystąpienie związku - to unikalne i identyfikowalne powiązanie zachodzące pomiędzy pojedynczymi wystąpieniami encji z uczestniczących w związku zbiorów encji.

Studenci	Przypisani do ▷	◁ Zawiera	**Grupy**
studentID	12..32	1..1	groupID

– Stopień związku - to liczba uczestniczących w nim zbiorów encji. W notacji UML używa się symbolu rombu do przedstawienia związku o stopniu większym niż dwa np. związek opisujący ustalanie przez nauczyciela terminu egzamin ze studentami.



– Atrybut - to konkretna cecha encji lub związku np. nazwisko.
– Atrybut pochodny - to atrybut reprezentujący wartość, która jest wyliczana z innego atrybutu lub ze zbioru atrybutów, niekoniecznie pochodzących z tego samego zbioru encji np. wiek lub liczba studentów w grupie.
– Dziedzina atrybutu - to zbiór dopuszczalnych wartości dla przynajmniej jednego atrybutu.

Studenci
studentID
imie
nazwisko

– Klucz kandydujący - to najmniejszy zbiór atrybutów, który jednoznacznie identyfikuje każde wystąpienie encji w zbiorze encji.
– Klucz główny PK (ang. Primary Key) - to klucz kandydujący, który został wybrany do jednoznacznej identyfikacji każdego z wystąpień encji w zbiorze encji.
– Klucz alternatywny - to klucz kandydujący, który nie został wybrany jako klucz główny.
– Klucz prosty - to klucz, który składa się z jednego atrybutu.
– Klucz złożony - to klucz, który składa się z co najmniej dwóch atrybutów.
– Krotność - to liczba lub zakres możliwych wystąpień encji z jednego zbioru, które mogą być w danym związku z pojedynczym wystąpieniem innej powiązanej encji.

Związki binarne można podzielić na:

– "jeden do jednego" (1:1),
– "jeden do wielu" (1:*),
– "wiele do wielu" (*:*).

Krotności [3]:

– 0..1 - zero lub jedno,
– 1..1 - dokładnie jedno,
– 0..* - zero lub więcej,
– 1..* - jedno lub więcej,
– n..m - minimalnie n, a maksymalnie m
– n, m, o..p, q - n lub m lub od o do p lub q wystąpień encji.

3 Konceptualne projektowanie bazy danych

Konceptualne projektowanie bazy danych [3] - proces tworzenia modelu dla informacji wykorzystywanych w przedsiębiorstwie, niezależnego od wszelkich szczegółów reprezentacji fizycznej.

Krok 1

Stwórz lokalny konceptualny model danych dla każdej perspektywy

Perspektywa oznacza tutaj obszar zastosowań, punkt widzenia innego użytkownika lub grupy użytkowników.

Krok 1.1

Określ występujące zbiory encji

Może zostać to zrealizowane na podstawie specyfikacji wymagań użytkownika (bazując na występujących rzeczownikach).

Po tym kroku powinniśmy uzyskać opis zbiorów encji, z których każdy powinien zawierać: nazwę zbioru encji, opis, aliasy (synonimy) oraz własności.

Nazwa zbioru encji	Opis	Aliasy	Własności
Studenci	Pojęcie ogólne opisujące wszystkich studentów uczących się na uczelni	Uczniowie	Każdy student może być przypisany tylko do jednego kierunku
...	...	...	...

Krok 1.2

Ustal typy występujących związków

Związki mogą zostać zidentyfikowane na podstawie specyfikacji wymagań użytkownika (bazując na występujących czasownikach).

Należy zwrócić uwagę na występowanie:

- stopień związku (np. binarny lub wyższego stopnia),
- występowanie związków rekurencyjnych,
- krotności związków albo ich dolne lub górne ograniczenia,
- sprawdzenie czy związki są reprezentacją rzeczywistych zależności,
- pułapki wachlarzowe i szczelinowe,
- występowanie każdego ze zbiorów encji w co najmniej w jednym związku.

Po tym kroku powinniśmy uzyskać opis związków, z których każdy powinien składać się z nazwy związku, nazw zbiorów encji oraz krotności. Zazwyczaj wygodniej jest reprezentować zbiory encji oraz związków w postaci graficznej za pomocą diagramów związków encji niż w postaci tekstowej. Do zapisania diagramów można wykorzystać język UML.

Nazwa zbioru ecnji	Krotność	Związek	Krotność	Nazwa zbioru encji
Studenci	1:*	PrzypisaniDo	1:*	Grupy
...	...	...	...	

Pułapka wachlarzowa [3] występuje w sytuacji, gdy model przedstawia związek pomiędzy pewnymi zbiorami encji, ale wynikające z tego ścieżki pomiędzy wystąpieniami encji nie są jednoznaczne.

Pułapka szczelinowa [3] występuje, gdy model sugeruje istnienie związku pomiędzy zbiorami encji, ale nie istnieją ścieżki łączące pewne wystąpienia tych encji.

Krok 1.3

Określ atrybuty odpowiadające poszczególnym zbiorom encji i związkom

Może zostać to zrealizowane na podstawie specyfikacji wymagań użytkownika (bazując na występujących rzeczownikach, które określają właściwości poszczególnych wystąpień encji lub związków).

Należy zwrócić uwagę na występowanie:

- atrybutów prostych i złożonych,
- atrybutów jednowartościowych i wielowartościowych,
- atrybutów pochodnych,
- potencjalnych problemów.

Po tym kroku powinniśmy uzyskać opis atrybutów zawierający nazwę zbioru encji, do którego powinien być przyporządkowany, nazwę, opis, typ danych i długość, aliasy (jeżeli występują), wartość domyślną (jeśli posiada), jeżeli atrybut jest złożony to atrybuty proste, z których się składa, informację czy jest to atrybut wielowartościowy, czy jest to atrybut pochodny, a jeśli tak to sposób wyznaczania jego wartości.

Nazwa zbioru encji	Nazwa atrybiutu	Typ, długość	Złożony	Wielo-warto-ściowy	...
Studenci	studentID	Liczba, 6 cyfr	Nie	Nie	...
	imię	tekst, 20 znaków	Nie	Nie	...
	nazwisko	tekst, 20 znaków	Nie	Nie	...
	telefon	Liczba, 9 cyfr	Nie	Tak	...
	adres	tekst, 40 znaków	Tak	Nie	...
...	...	...	...	...	...

Krok 1.4

Określ dziedziny poszczególnych atrybutów

Po tym kroku powinniśmy uzyskać opis dziedzin, dopuszczalny zbiór wartości atrybutów, dopuszczalny zakres długości i format atrybutów oraz opcjonalnie zbiór dopuszczalnych operacji na atrybutach, wykaz atrybutów, które mogą być ze sobą porównywane lub używane łącznie.

Przykłady:

- dziedzina atrybutu kod pocztowego: zbiór ciąg znaków zgodny z formatem: liczba z zakresu 00 do 99, znak '-', liczba z zakresu 000-999,
- dziedzina atrybutu tablicaRejestracyjnaPojazdu: ???

Krok 1.5

Ustal klucze kandydujące oraz klucze główne

Przy wyborze klucza głównego spośród kluczy kandydujących należy zwrócić uwagę na:

- klucz o najmniejszej liczbie atrybutów,
- klucz, którego wartości najrzadziej ulegają zmianom,
- klucz kandydujący o najmniejszej liczbie znaków lub o najmniejszej wartości maksymalnej,
- klucz, z którego najłatwiej będzie korzystać użytkownikowi.

Można także dodać nowy atrybut jednoznacznie identyfikującego każde wystąpienie encji np. *nauczycielNr*.

Po tym kroku powinniśmy uzyskać informacje o kluczach głównych oraz kandydujących. Nie dla każdego zbioru encji możemy na tym etapie wyznaczyć klucz główny. Dla słabych zbiorów encji ustalenie kluczy głównych jest możliwe dopiero po kroku 2.2.

Krok 1.6

Rozważ możliwość zastosowania zaawansowanych metod modelowania (krok opcjonalny)

Rozszerzony model związków encji (EER ang. Enhanced Entity-Relationship model) [3] - jest to model ER rozszerzony o pojęcia, takie jak m.in. specjalizacja-generalizacja, agregacja i kompozycja.

Nadklasa [3] - „To zbiór encji zawierający przynajmniej jedną podgrupę złożoną ze swoich wystąpień, która powinna być oddzielnie reprezentowana w modelu danych."

Podklasa [3] - „To wyróżniona podgrupa wystąpień ze zbioru encji, która powinna być oddzielnie reprezentowana w modelu danych."

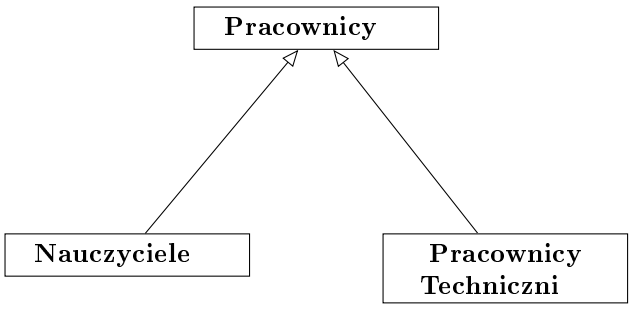
Przykład: pracownik (nadklasa) - nauczyciel (podklasa)

Specjalizacja [3] - „To proces maksymalizacji różnic pomiędzy encjami, realizowany poprzez identyfikację wyróżniających ich charakterystyk."

Przykład: Pracownik → Nauczyciel, PracownikTechniczny

Generalizacja [3] - „To proces minimalizacji różnic pomiędzy encjami, realizowany poprzez wyznaczane ich wspólnych cech."

Przykład: Nauczyciel, PracownikTechniczny → Pracownik



Agregacja [3] - „Reprezentuje związki typu 'jest częścią' lub 'ma' pomiędzy zbiorami encji, w których jeden uczestnik związku jest 'całością', a drugi 'częścią'."

Kompozycja [3] - „To specjalna forma agregacji przedstawiająca powiązanie pomiędzy encjami, w którym występuje silny związek 'posiadania' części przez całość oraz zgodności okresów ich istnienia."

Krok 1.7

Zweryfikuj utworzony model pod kątem występowania redundancji

W tym kroku należy:

- ponownie sprawdzić związki wzajemnie jednoznaczne (1:1), jeżeli istnieją dwa zbiory encji reprezentujące te same obiekty ze świata rzeczywistego, to jeden z nich powinien zostać usunięty, jeżeli mają one różne klucze główne, to należy wybrać jeden z nich, a drugi potraktować jako klucz alternatywny,
- usunięcie związków redundantnych (nadmiarowych), czyli związków dostarczających informacji, które mogą zostać uzyskane w inny sposób.

Redundancja (łac.) [Encyklopedia PWN], nadwyżka informacji zawartej w wiadomości w stosunku do tej ilości informacji, która jest niezbędna, aby wiadomość została przekazana bez uszczuplenia treści.

Krok 1.8

Zweryfikuj możliwość realizacji transakcji w lokalnym modelu konceptualnym

Pierwsza metoda polega na sporządzeniu opisu wymagań poszczególnych transakcji i sprawdzeniu czy wszystkie informacje niezbędne do ich wykonania, takie jak encje, związki i atrybuty znajdują się w modelu.

W drugiej metodzie na diagramie encji należy nanieść ścieżki reprezentujące transakcje. Umożliwia to określenie, które elementy są wykorzystywane w poszczególnych transakcjach oraz znalezienie takich, które nie są nigdy wykorzystywane.

Najczęstszym powodem, dla którego nie można zrealizować wszystkich transakcji jest pominięcie w modelu jakiegoś zbioru encji, związku lub atrybutu.

Jakie elementy są wymagane do zrealizowania następującej transakcji: dodanie studenta do grupy laboratoryjnej?

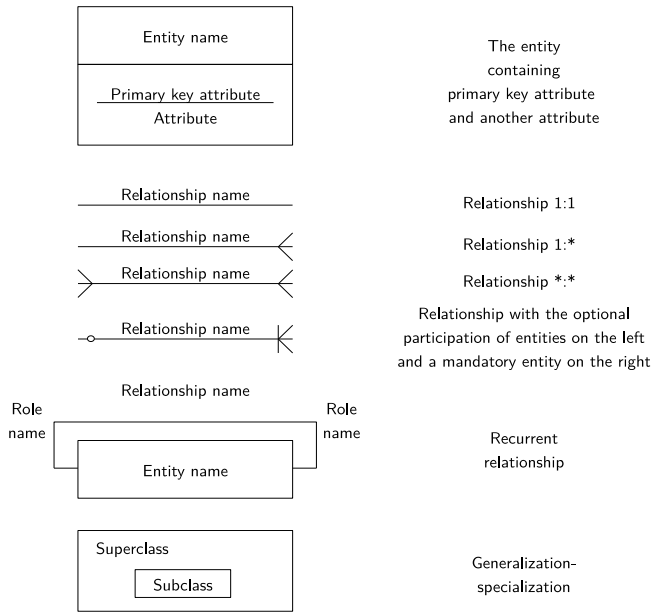
Studenci	Przypisani do ▷	◁ Zawiera	**Grupy**
studentID	12..32	1..1	groupID

Krok 1.9

Omów lokalny konceptualny model danych z użytkownikiem

Użytkownik powinien zweryfikować poprawność wykonania lokalnego konceptualnego modelu danych, na który składa się diagram ER oraz dokumentacja.

4 Notacja "kurzej łapki"



Notacja "kurzej łapki" . Thomas Connolly, Carolyn Begg, Systemy baz danych, Wydawnictwo RM, 2004

Literatura

[1] Jeffrey D. Ullman, Jennifer Widom, "Podstawowy wykład z systemów baz danych", WNT, 2001

[2] Jeffrey D. Ullman, Jennifer Widom, "Systemy baz danych, Pełny wykład", WNT, 2006

[3] Thomas Connolly, Carolyn Begg, "Systemy baz danych", Wydawnictwo RM, 2004

[4] Thomas Connolly, Carolyn Begg, "Database systems", Peason, 2015

[5] Kevin Loney, Bob Bryla, Oracle Database 10g, Podręcznik administratora baz danych. Helion 2008

[6] Scott Urman, Ron Hardman, Michael McLaughlin, "Oracle Database 10g, Programowanie w języku PL/SQL", Helion, 2008

[7] Price J, "Oracle Database 11g i SQL Programowanie", Helion, 2009

[8] Ramez Elmasri, Shamkant B. Navathe, "Wprowadzeni do systemów baz danych", Helion, 2005

[9] Sharon Allen, "Modelowanie danych", Helion, 2006

[10] Jose A. Ramalho, Oracle 8i, Mikom, 2001

[11] Dokumentacja do bazy danych Oracle dostępna na stronie www.oracle.com

[12] Sadalage P. J., Fowler M., "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley Professional, 2012

[13] Michael Stonebraker, "New Sql: An Alternative to Nosql and Old Sql For New Oltp Apps", June 16, 2011

[14] Oracle Database PL/SQL Language Reference 11g Release 1 (11.1) B28370-02 (B28370.pdf)

Materiały do przedmiotu dostępne są na stronie: http://weaii-moodle.tu.kielce.pl/