

BAZY DANYCH 2020

PROJEKT BAZY DANYCH

28 maja 2020

Elżbieta Jowik

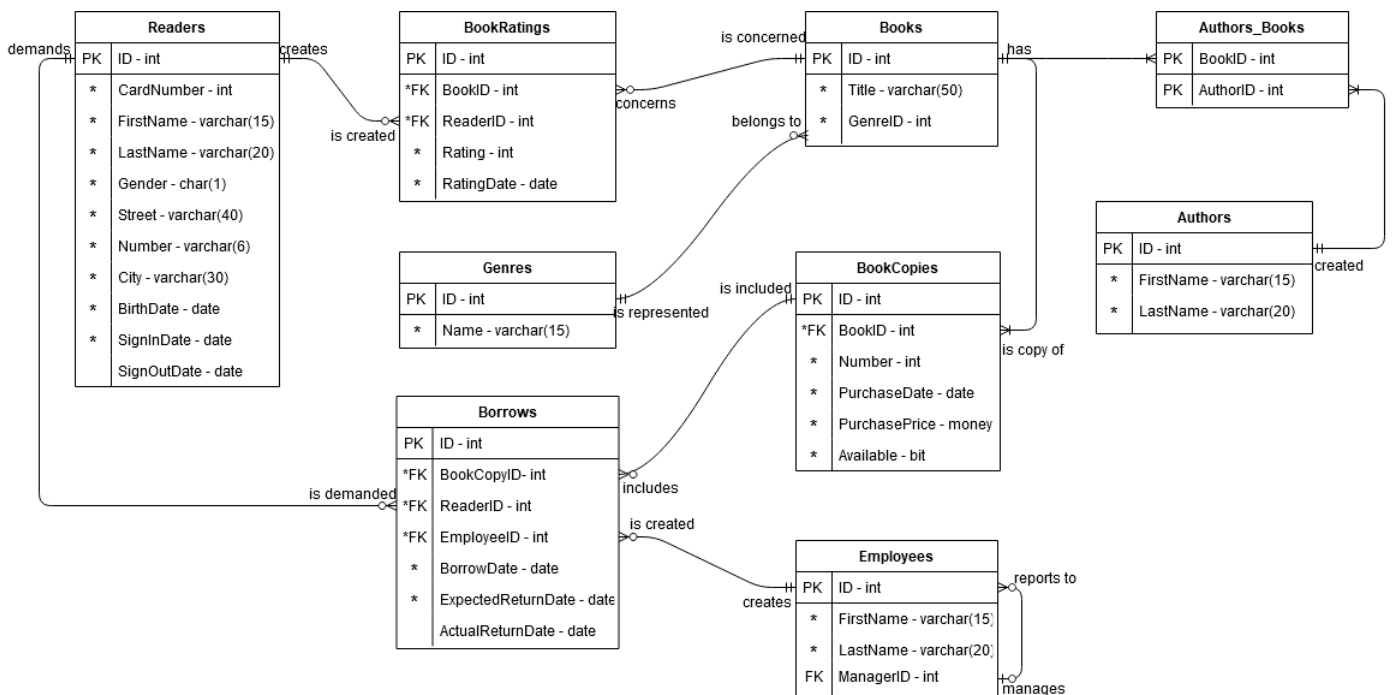
1 Opis problemu i wstępne założenia

Baza *Library* dotyczy obsługi podstawowej funkcjonalności biblioteki książek. Pracownicy biblioteki wypożyczają czytelnikom różne wydania książek, czytelnicy zaś mogą opiniować poszczególne pozycje. Zakładamy, że w tabeli *Books* bazy danych znajdują się tylko takie książki, których co najmniej jeden egzemplarz znajduje się w zasobach rozważanej biblioteki. Analogicznie w tabeli *Authors* znajdują się tylko ci autorzy, których co najmniej jedna książka jest w księgozbiorze. Ponadto dopuszczalna jest sytuacja, w której różne książki mają różne oczekiwane okresy wypożyczeń, a także, że każda książka może posiadać wielu autorów oraz każdy autor może znajdować się na okładkach wielu książek.

2 Opis bazy danych *Library*

2.1 Schemat bazy *Library*

Adnotacja: Do stworzenia wszystkich połączeń wykorzystane zostały identyfikatory poszczególnych obiektów z ograniczeniami kluczy głównych/obcych w zależności od bieżącej tabeli. Decyzja ta jest umotywowana faktem jednoznaczności identyfikacji obiektów względem tych pól.



Rysunek 1: Diagram ER prezentujący strukturę tabel wchodzących w skład bazy danych *Library* oraz relacje pomiędzy nimi. Notacja: *PK* - klucz główny, *FK* - klucz obcy, * - nie pozwala na wprowadzenie wartości *NULL* w kolumnie, próba ustawienia takiej wartości spowoduje wystąpienie błędu.

2.2 Omówienie tabel

W skład bazy danych wchodzi dziewięć tabel, przy czym jedna z nich (*Authors_Books*) jest pomocniczą tabelą łącznikową, pozwalającą zniwelować związek **wiele-do-wielu** pomiędzy tabelami *Authors*, *Books*. We wszystkich tabelach zdefiniowane zostało ograniczenie klucza podstawowego (ang. *Primary Key*, ozn. PK), dzięki czemu w bazie danych zapewniona jest integralność encji¹ i możliwe jest jednoznaczne odniesienie się do każdego z rekordów danych w każdej tabeli. Kluczem głównym może być dowolny klucz potencjalny, jednak w naszej bazie we wszystkich przypadkach, poza tabelą pomocniczą, ograniczenie klucza głównego naniesione jest na kolumnę *ID*. Docelowo będzie to bowiem specjalny atrybut automatycznie pobierany z sekwencji stanowiący jednoznaczny identyfikator każdego rekordu tabeli.

W przypadku wielu praktycznych zastosowań, jeśli z poziomu bieżącej tabeli chcemy odnieść się do rekordów zawartych w innej, to rekordy te przechowywane są w bieżącej tabeli pod kluczem obcym (ang. *Foreign Key*, ozn. FK). O ile w przypadku kolumn będących kluczami głównymi, już samo ograniczenie jest tożsame z brakiem możliwości występowania *NULL* w tej kolumnie, o tyle ograniczenie klucza obcego nie daje takiej gwarancji. Reguła integralności referencyjnej, dotycząca kluczy obcych, mówi, że każda wartość klucza obcego może znajdować się w jednym z dwóch stanów. W ogólności wartość klucza obcego odwołuje się do wartości klucza głównego w innej tabeli w bazie danych, jednak dopuszczalna jest sytuacja, gdy wartość klucza obcego jest reprezentowana przez *NULL*. Wówczas, w zależności od natury problemu, w sposób jawny stwierdzamy, że nie ma związku pomiędzy obiektami albo że ten związek jest nieznany². Jedynym kluczem obcym, na które nie zostało narzucone ograniczenie *NOT NULL* jest pole *ManagerID* w tabeli *Employees*. Jest to charakterystyczny klucz obcy służący stworzeniu połączenia pomiędzy polami znajdującymi się w obrębie jednej encji. Pozostałe klucze obce utworzone na potrzeby zadania w bieżącej tabeli, odwołują się do wartości klucza głównego w innej i służą stworzeniu odpowiednich połączeń pomiędzy różnymi encjami. W celu podkreślenia, że w większości przypadków w naszej bazie, występowanie obiektu *NULL* w kolumnach kluczy obcych jest nieporządane na powyższym diagramie zastosowano oddzielne oznaczenia rozróżniające klucze obce z dopuszczalną wartością *NULL* i pozostałe. Odpowiednio: *FK* i **FK*.

Większość danych, w zilustrowanych na rysunku 1 tabelach, jest wymagana w trakcie tworzenia rekordów (ograniczenie *NOT NULL*, ozn. *). Wyjątki stanowią tutaj pola *SignOutDate* tabeli *Readers* oraz *ActualReturnDate* tabeli *Books*. W pierwszym przypadku pole oznacza datę wypisania się z biblioteki, a w drugim faktyczną datę zwrotu wypożyczonej książki. Wartości *NULL* oznaczają, odpowiednio: że dany użytkownik jest aktywnym czytelnikiem biblioteki oraz że czytelnik jeszcze nie oddał danej książki.

Struktury wszystkich tabel wynikają wprost z treści zadania. Modyfikacją, na którą chciałabym zwrócić uwagę jest typ pola *Available* w tabeli *BookCopies*. Jego sugerowa-

¹Reguła integralności, mówiąca, że każda tabela musi mieć klucz główny oraz że kolumna lub kolumny mające ograniczenie klucza głównego powinny być jednoznaczne i nie powinny zawierać wartości *NULL*.

²Beynon-Davies, 2003

nym typem był bowiem *boolean*, jednak ponieważ środowisko nie zapewnia jawnie takiej możliwości, zdecydowałam się, że kolumna ta będzie miała charakter bitowy taki, że wartość 1 będzie tożsama z *true*, a 0 z *false*.

2.3 Omówienie połączeń i ich motywacji

Zgodnie z zasygnalizowanym wcześniej założeniem przyjmujemy, że każda książka może posiadać wielu autorów oraz każdy autor może znajdować się na okładkach wielu książek. Stąd kilka rekordów z tabeli *Authors* może odpowiadać wielu rekordom z tabeli *Books*. Jest to przykład relacji **wiele-do-wielu**, której stosowanie w praktyce jest wysoce nie wskazane ze względu na ryzyko redundancji danych.

Utworzenie tabeli łącznikowej *Authors_Books*, scalającej obie tabele związkiem **jeden-do-wielu** jest umotywowane chęcią eliminacji tego problemu. W celu uniknięcia sytuacji, w których w bazie znajdują się dane autora, którego ani jedna książka nie znajduje się w zasobach biblioteki albo, że jakaś książka znajdująca się w rejestrze bazy nie ma autora, po obu stronach relacji obecne jest odpowiednie obostrzenie.

Kolejną relacją, która wynika wprost z przyjętych założeń jest związek **jeden-do-jednego lub wielu** pomiędzy tabelami *Books* i *BookCopies*. Pierwsza strona związku, jest efektem oczywistej obserwacji, że dany egzemplarz może być kopią jednej książki. Druga natomiast wynika z tego, że w bazie przechowywane są informacje dotyczące tylko takich książek, których co najmniej jeden egzemplarz jest w posiadaniu wypożyczalni. Takie obostrzenie pozwoli na ograniczenie rozmiaru tabeli *Books* i jej tym samym zajętości pamięciowej. Warto zauważyć, że analogicznego obostrzenia nie przyjęto dla związku tabel *Books-Genres*. Jest to relacja **jeden-do-zera lub wielu**, z której wynika, że w rejestrze bazy, przechowujemy informację o gatunkach, dla których biblioteka nie posiada żadnego przedstawiciela. Niekonsekwencja ta jest celowym zabiegiem, wynikającym z faktu, że zagregowany aglomerat gatunków literackich, to dyskretny zbiór o relatywnie niewielkiej liczności. Ponadto istnieje niewielkie prawdopodobieństwo, że w pełnym księgozborze nie znajdzie się ani jeden reprezentant danego gatunku. Druga strona omawianej relacji, wynika z obserwacji, że dana książka może przynależeć tylko do jednego gatunku.

W opisie problemu zasygnalizowano, że poszczególne książki mogą mieć różne dopuszczalne okresy wypożyczenia. Oznacza to, że o dacie zwrotu konkretnego egzemplarza nie musi decydować data wydania go czytelnikowi, ale inne czynniki, jak na przykład wartość klucza obcego *BookID*. Dlatego pomiędzy tabelami *BookCopies* i *Borrows* zastosowano relację **jeden-do-zera lub wielu**, dzięki czemu każdy rekord w tabeli *Borrows*, odpowiada wypożyczeniu jednego egzemplarza danej książki. W przypadku wypożyczenia wielu pozycji przez tego samego czytelnika, w tabeli tej jest tworzonych dokładnie tyle rekordów, ile książek zostanie wydanych.

Oczywistym spostrzeżeniem jest, że dany czytelnik ma prawo do wielokrotnego wypożyczania, a konkretne wypożyczenie może zostać zarejestrowane względem pojedynczej wartości klucza obcego *ReaderID*. Ponadto każde wypożyczenie musi zostać utworzone przez pracownika biblioteki, a dany pracownik może zarejestrować wiele wypożyczeń lub nie zarejestrować żadnego. Stąd relacje **jeden-do-zera lub wielu** pomiędzy tabelami: *Readers - Borrows* i *Employees - Borrows*. Należy zauważyć, że w obrębie tabeli pra-

owników istnieje relacja **zero lub jeden-do-zera lub wielu**. Związek ten wynika z faktu możliwości hierarchizacji pracowników biblioteki, z założeniem, że jeden pracownik może podlegać jednemu zwierzchnikowi lub nikomu, gdy sam pełni najwyższe stanowisko, natomiast dany menadżer może zarządzać wieloma pracownikami lub nikim, w sytuacji gdy stanowisko jest w pełni samodzielne.

Rozważana biblioteka zapewnia swoim czytelnikom możliwość opiniowania pozycji znajdujących się w księgozbiorze w określonej skali. Relacja **jeden-do-zera lub wielu** pomiędzy tabelami *Readers* i *BookRatings* jest umotywowana tym, że każdy czytelnik może oceniać różne książki lub nie korzystać z tej możliwości, natomiast dana recenzja podlega dokładnie jednej osobie opiniującej (pojedynczej wartości klucza *ReaderID*).

w tabeli *BookRatings* obecne są dwa klucze obce: *ReaderID*, *BookID*. Aby wykorzystać pełen potencjał tej tabeli i umożliwić jej przetwarzanie zarówno względem recenzenta jak i książki została ona połączona relacją **zero lub wiele-do-jednego** z tabelą *Books*. Dzięki takiemu rozwiązaniu każdy rekord w tabeli *BookRatings* odpowiada konkretnej opinii danego recenzenta na temat określonej przez wartość klucza obcego *BookID* pozycji w księgozbiorze. W sytuacji, gdy dany czytelnik opiniuje więcej niż jedną książkę, dla każdej oceny tworzony jest oddzielny rekord w tabeli *BookRatings*.