

Wszystkie materiały udostępnione w opracowaniu przeznaczone są wyłącznie dla uczniów II Liceum Ogólnokształcącego im. Mikołaja Kopernika w Bydgoszczy. Prezentowany materiał został przygotowany na potrzeby zajęć lekcyjnych z informatyki i stanowi jedynie dodatek do treści, które uczeń powinien przyswoić podczas lekcji oraz z podręcznika.

Opracowanie:
Dariusz Bagnucki

BYDGOSKA AGENCJA KOSMICZNA
(oddział terenowy)
85-000 Bydgoszcz, ul. Gwiazdka 100
tel. 52 999 666 999

Bydgoszcz, 1 listopada 2023
Szczegółowa Poni
Krzyszto Borys
ul. Swarzewska 49/30
35-731 Nowy Dwór

temat okresu zatrudnienia oraz podstawy rozwiązania umowy o pracę.

na pracodawcy obowiązkiem informacyjnym przekazujemy Państwu
zatrudnienia pracownika oraz przyczyny rozwiązania umowy o pracę.
PESEL: 33333903630

ZAGROŻENIA W S

Gdy mówimy o zagrożeniu w sieci od fani kodów i zabezpieczeń do naszego, domowego komputera.

Czy tak jest faktycznie?

Otóż prawdziwi hakerzy nie są prywatnych, no chyba że a o dużym zasięgu działania.

Uwagę zawodowych hakerów giełdy, serwery rządowe. Są to specjaliści o ogromnym sławy, albo dla pieniędzy, albo

II Liceum Ogólnokształcące im. Mikołaja Kopernika w Bydgoszczy

RELACYJNE BAZY DANYCH

Cel lekcji:

Celem lekcji jest przedstawienie pojęcia relacyjnych baz danych, pokazanie ich wad oraz zalet oraz zasad ich tworzenia.

Słuchacze zapoznają się z programem do tworzenia schematów blokowych i za jego pomocą przygotowują przykładową strukturę wraz ze schematem powiązań wewnątrz bazy.

Baza danych to uporządkowany zbiór danych z pewnej dziedziny, w którym są one tak zorganizowane, aby można było w łatwy sposób uzyskiwać do nich dostęp oraz otrzymywać na ich podstawie potrzebne informacje.

Bazą danych może być np. zbiór danych o klientach, kartoteka pacjentów przychodni lekarskiej, kolekcja naszych płyt muzycznych lub filmów.

Aby możliwe było jednak otrzymanie poprawnych informacji z bazy, zawarte w niej dane muszą zostać odpowiednio przetworzone. Przetwarzanie danych obejmuje szereg czynności, począwszy od ich wprowadzania przez edytowanie, przeszukiwanie, oraz opracowanie danych wyjściowych np. za pomocą formularzy czy raportów.

Zalety korzystania z baz danych:

- szybkie wyszukiwanie informacji,
- łatwe wykonywanie obliczeń,
- możliwość przechowywania dużej ilości danych na małej powierzchni,
- szybkie porządkowanie danych.

Komputerowe bazy danych możemy podzielić ze względu na sposób dostępu do nich na:

- **bazy jednostanowiskowe** – baza i system zarządzania nią znajdują się na jednym komputerze, dostęp do bazy ma jeden użytkownik,
- **bazy ze współdzieleniem plików** – baza i system zarządzania są dostępne na wielu komputerach,
- **bazy typu klient-serwer** – baza i system zarządzania nią stanowią odrębny proces, często na wydzielonym komputerze – serwerze bazy danych.

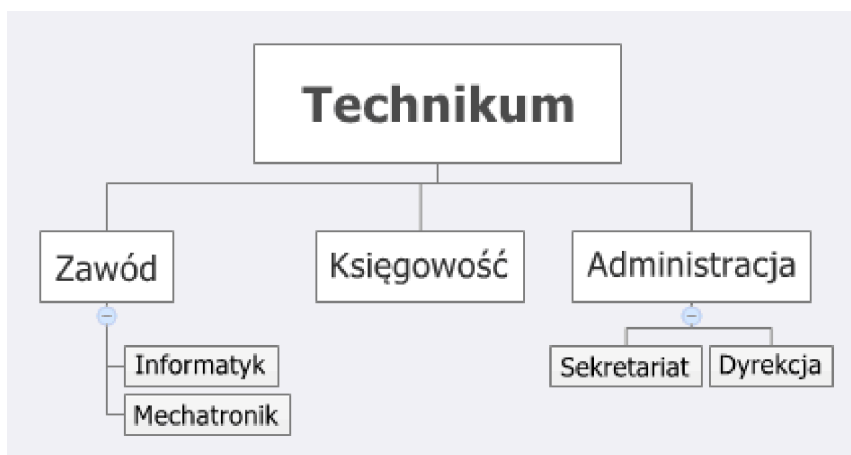
Model bazy danych można określić jako logiczny sposób organizacji danych złożony ze zbioru reguł opisujących dane, wzajemne zależności między nimi, dozwolone operacje na danych oraz zasady dostępu do systemu baz danych.

Wyróżniamy 6 modeli:

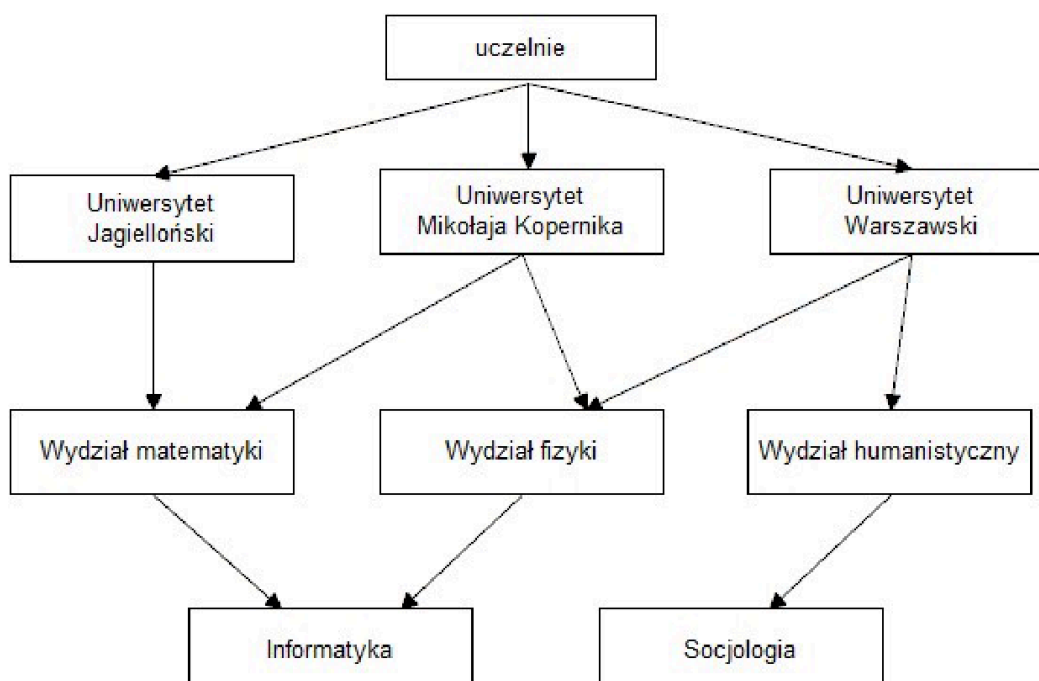
1. Model jednorodny – oparty na jednej tabeli. Może się charakteryzować nadmiarowością danych i utrudnionym wyszukiwaniem informacji:

Przykład: Kłos Marian, Legnica, ul. Senacka 39, tel. 607080901

2. Model hierarchiczny – oparty na strukturze odwróconego drzewa:

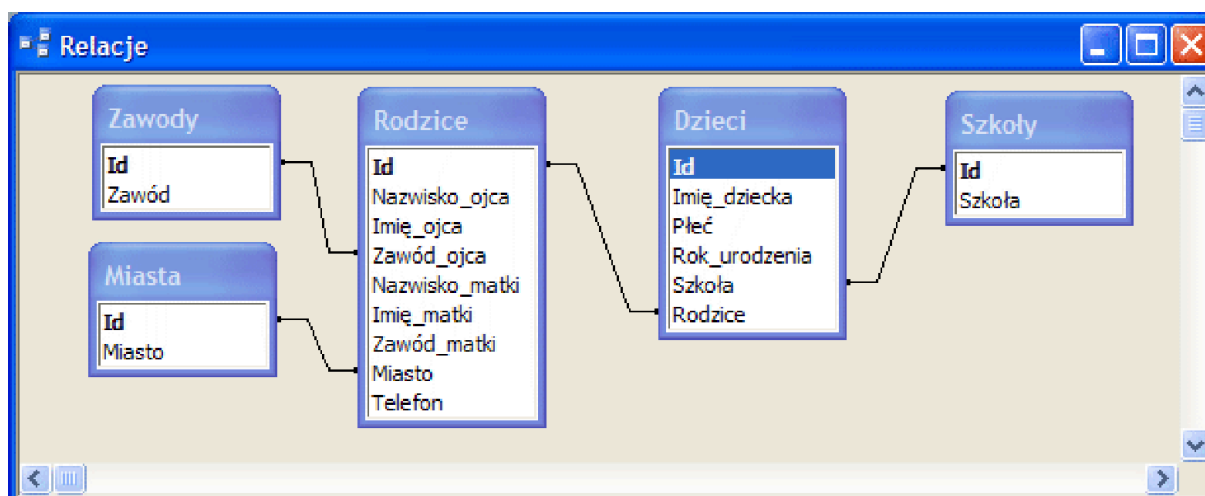


3. Model hierarchiczny (sieciowy) – poszczególne elementy mogą łączyć się z innymi i tworzyć kratownicę.



4. Model obiektowy – łączy cechy programów komputerowych tworzonych w językach programowania obiektowego z cechami aplikacji bazodanowych. Obiekt w bazie reprezentuje obiekt w świecie rzeczywistym.

5. Model relacyjny – tabele tworzą ze sobą powiązania zwane relacjami.



Więcej dowiedzie się o bazach danych z materiałów filmowych, do obejrzenia których zachęcam:

FILM: [Jak bazy danych działają od środka?](#)

FILM: [Relacyjne bazy danych co to i po co?](#)

6. Model postrelacyjny (relacyjno-obiektowy) – w modelu postrelacyjnym bazy danych zakłada się, że są to bazy relacyjne poszerzone na przykład o elementy obiektowości, obsługę XML, rozwiązania analityczne, zapytania historyczne. Model postrelacyjny powstał w wyniku rozszerzenia relacyjnego modelu baz danych o elementy ułatwiające opisanie skomplikowanej rzeczywistości. Są to na przykład złożone struktury danych, zagnieżdżone relacje, atrybuty wirtualne, abstrakcyjne typy danych czy funkcje rozszerzalne.

Podstawowym obiektem w każdej komputerowej bazie danych jest tabela.

Zapisywanie danych w tabelach nie odbywa się przypadkowo, lecz według pewnych reguł. Jeżeli chcemy np. gromadzić w bazie informacje na temat naszej kolekcji płyt muzycznych, to zapisujemy tytuł płyty, imię i nazwisko (nazwę) wykonawcy, nazwę wytwórni i rok wydania albumu. Te same dane będziemy gromadzić o każdej kolejnej płycie w naszej kolekcji.

Zbiór danych opisujący pojedynczy obiekt (płytę) w bazie, jest nazywany rekordem danych.

Pojedynczy zapis w rekordzie (np. tytuł płyty) jest nazywany polem rekordu. Można więc powiedzieć, że dane o każdej płycie w bazie zostaną zapisane w jednym rekordzie.

Przykład:

Scorpions Love At First Sting PolyGram 1984

Rekord przykładowej bazy danych zawiera pola:

- Pole z nazwą wykonawcy,
- Pole z tytułem płyty,
- Pole z nazwą wytwórni,
- Pole z rokiem wydania.

Z kolei zbiór rekordów to tabela, a jeden rekord danych to jeden wiersz tej tabeli.

W tabeli ze wspomnianej wcześniej kolekcji płyt będzie tyle wierszy (rekordów), ile w kolekcji jest płyt.

Tabela dostarcza informacji wszystkim innym obiektom bazy (np. formularzom i raportom).

W zależności od rodzaju baza może opierać się na jednej, kilku lub kilkunastu tabelach.

Dlaczego dane zwykle gromadzi się nie w jednej, ale w kilku tabelach?

Pokażmy to na przykładzie wypożyczalni książek.

W przypadku baz danych płaskich (opartych na jednej tabeli) określone informacje w bazie danych się powtarzają. Jeżeli czytelnik wypożyczył kilka książek, to w każdym rekordzie powtarzają się jego dane. Powtarzanie się pewnych informacji w bazie danych nazywamy redundancjami.

- Gdy wybrany czytelnik zmieni swój adres zamieszkania, będziemy musieli dokonać zmian w każdym rekordzie, w którym on występuje (jest to trudne i czasochłonne przy większej liczbie rekordów dla tego klienta). Problemy związane z taką aktualizacją nazywamy anomaliami aktualizacji bazy danych.

- Gdy czytelnik zwraca książkę, zdarza się że następuje usunięcie odpowiedniego rekordu z bazy. Zwrot wszystkich wypożyczonych książek przez jednego czytelnika w takim przypadku spowoduje usunięcie wszystkich rekordów z bazy, z jego danymi.

Tym samym zostały usunięte informacje o samym czytelniku. Z kolei przy ponownej wizycie w bibliotece musimy od nowa wpisywać jego dane i adres. Problemy tego typu są **anomaliami przy usuwaniu**.

Zanim przejdziemy do omówienia przykładu relacyjnej bazy danych, trzeba powiedzieć o budowie pojedynczego rekordu (wiersza) bazy danych.

Mamy dwa podstawowe modele.

Pierwszy z nich oparty jest na rekordach o stałej długości.

Pamiętamy listy pracowników z naszych ćwiczeń w Excelu. Jedna linia zawierająca wszystkie dane o jednym pracowniku to właśnie jeden rekord.

Każdy rekord ma jednak swoje pola, jak imię, nazwisko, adres, data zatrudnienia.

W pierwszym typie baz danych każde pole rekordu ma z góry zadeklarowaną ilość znaków.

Drugi model oparty jest na polach o zmiennej długości.

Skutkuje to tym, że każdy rekord może mieć zupełnie inną wielkość.

Oba modele mają swoje wady i zalety, jednak przy obecnym rozwoju technologii, od ładnych kilku lat model drugi wiedzie prym w rozwiązaniach bazodanowych.

Model pierwszy zakładał dużą nadmiarowość zbędnych znaków, co prowadziło do gigantycznego wzrostu wielkości plików, w których były zapisywane tabele. Przy ogromnych bazach danych powodowało to bardzo duże problemy.

Jednak przy mniejszych bazach rozwiązanie to potrafiło być szybsze. Zwiększenie szybkości pracy komputerów spowodowało, że oba rozwiązania są równie wydajne.

Największą wadą pierwszego rozwiązania poza wielkością plików była konieczność konwersji danych w przypadku, gdy autor przewidział zbyt mało znaków przykładowo na nazwisko czy adres.

Konwersja taka przy dużych bazach zajmowała nieraz kilka dni, skutecznie wyłączając całą infrastrukturę informatyczną przedsiębiorstw czy urzędów.

Model relacyjnej bazy danych na przykładzie bazy wypożyczonych książek.

Tabela Wypożyczenia			
IDCzytelnika	IDKsiążki	Data wypożyczenia	Data zwrotu
1	LIT1	23.04.2023	15.05.2023
2	LIT2	23.03.2023	14.05.2023
1	LIT3	23.02.2021	16.05.2023
3	LIT4	22.09.2022	17.05.2023
2	LIT5	23.01.2023	19.05.2023
4	LIT6	21.02.2023	

Tabela Książki		
IDKsiążki	Autor	Tytuł
LIT1	Dymna Anna	Zaniechanie
LIT2	Diuk Mariola	Zaćmienie
LIT3	Tuch Marcin	Przygody Kleksa
LIT4	Łoś Wanda	Brzdąc
LIT5	Stec Zenon	Maria
LIT6	Mleko Adam	Klaus

		Tabela Czytelnicy	
IDCzytelnika	Nazwisko i Imię	Kod i Miejscowość	Ulica i nr domu
1	Wrona Marian	30-500 Katowice	Kolejowa 5
2	Kulma Anna	30-450 Zabrze	Motylowa 8
3	Kaszuba Zenon	30-470 Sosnowiec	Warszawska 3
4	Zawada Robert	30-480 Gliwice	Wróbla 60

Baza ta składa się z trzech tabel zapisanych w oddzielnych plikach.

Tabela nadrzędna (tabela książki) korzysta z usług dwóch tabel podległych. Tabeli czytelników oraz tabeli wypożyczeń.

Dzięki takiemu rozwiązaniu tabela książki nie musi przechowywać informacji ani o czytelnikach, ani o wypożyczeniach.

Jest ona powiązana z bazami podległymi jedynie numerami (kluczami) zarówno czytelników jak i książek.

Tabelę wypożyczenia można dowolnie modyfikować, usuwać rekordy bez utraty istotnych informacji zarówno o księgozbiorze jak i o czytelnikach.

Można również w miarę potrzeb modyfikować dane czytelników oraz informacje o książkach. Nasza tabela wypożyczenia będzie dysponowała cały czas aktualnymi danymi adresowymi czytelników.

Każda taka zmiana będzie odbywała się wyłącznie jeden raz i będzie zapisywana wyłącznie w jednym rekordzie.

Raz zmienione dane o czytelniku np. adres, nazwisko w przypadku zawarcia związku małżeńskiego czy numer dowodu osobistego będą zawsze widoczne w całej bazie danych bez konieczności ich zmiany we wszystkich rekordach każdej tabeli.

UWAGA: NIE ZAWSZE MOŻEMY DOKONYWAĆ TAKICH ZMIAN.

Rozwiązanie polegające na modyfikacji danych w tabeli czytelników jest w pełni dopuszczalne np. w przypadku biblioteki. Nie będziemy potrzebowali starego adresu zamieszkania czytelnika czy numeru nieaktualnego dowodu osobistego.

Są jednak dziedziny życia (na przykład rejestry sprzedaży przeznaczone do celów skarbowych), w których możemy modyfikować tylko część danych, pozostałe informacje muszą pozostać niezmienione nawet w przypadku faktycznych zmian dotyczących adresów, nazwisk itp. Jest to istotne ze względu na zachowanie prawdy historycznej dotyczącej transakcji.

Ciekawostka:

Klucze relacyjne nazywane czasami indeksami towarowymi o których wspominaliśmy spotkaliście wielokrotnie w sklepach. Takim kluczem jest kod paskowy, którym oznaczane są wszystkie dopuszczone do obrotu towary.

Dla producenta produkującego np. mleko, kod kreskowy umieszczony na kartoniku jest jego kluczem **WŁASNYM**.

Ten sam kod w chwili nabijania towarów na kasie, dla sklepu stanowi indeks **OBCY**.

Jak się dzieje, że w sklepie wiele towarów nie posiada identycznych kodów paskowych ... np. producent mleka mógłby zastosować taki sam kod jak producent szynki w puszcze?

Otóż istnieje centralny rejestr, który pilnuje nadawania kodów paskowych. Każdy produkt ma w nim swoją reprezentację, która nie może się powtórzyć.

Legalność zakupionego produktu można sprawdzić pod adresem:

<https://www.gs1.org/services/verified-by-gs1>

Baza ta obejmuje swoim zasięgiem wszystkie produkowane, dopuszczone do obrotu handlowego produkty oznaczone kodem paskowym. Nietrudno sobie wyobrazić ile zawiera ona pozycji i jak bardzo skomplikowany jest proces jej nadzorowania, aktualizacji czy zabezpieczenia przed uszkodzeniem.

W bazach danych możemy spotkać 5 rodzajów kluczy:

- **klucz prosty** – klucz jednoelementowy (jedna kolumna, np. numer PESEL),
- **klucz złożony** – klucz kilkuelementowy (kilka kolumn, np. imię i nazwisko),
- **klucz sztuczny** – dodatkowa kolumna identyfikująca każdy rekord utworzona w sposób sztuczny; z klucza sztucznego korzysta się wtedy, gdy z cech opisujących obiekt nie można wyodrębnić takiej, która spełnia warunki dla klucza głównego (np. id_klient),
- **klucz obcy** – klucz główny jednej tabeli (przykładowo tabeli jednego z dostawców towarów) wykorzystywany do tworzenia związków pomiędzy tabelami w bazie danych zarządzanej przez odbiorcę. Klucz obcy nie może być kluczem głównym w bazie danych gospodarza,
- **klucz kandydujący** – kolumna brana pod uwagę jako przyszły klucz główny w danej tabeli podczas projektowania bazy danych.

Mówiąc o bazach relacyjnych nie sposób pominąć zagadnienia normalizacji oraz optymalizacji.

Po co stosować normalizację relacyjnych baz danych?

Okazuje się, że gromadzenie nadmiarowych danych przyczynia się do marnowania przestrzeni dyskowej oraz kłopotów z wydajnością. Jeżeli zmieniamy dane w jednym miejscu, muszą one zostać zmienione w ten sam sposób w innych lokalizacjach. O wiele łatwiej zmienić konkretną informację (np. adres klienta), jeśli przechowuje się ją wyłącznie w jednej konkretnej tabeli (np. klienci).

O ile przechowywanie nazw, adresów czy kontaktów personalnych do istotnych pracowników w tabeli poświęconej naszym klientom ma sens, o tyle nie powinny się tam już znaleźć informacje zbędne z punktu widzenia zawierania transakcji czy nawiązywania kontaktów handlowych.

Gdy dane nie są spójne, dochodzić może do skomplikowanych problemów w kontekście znalezienia interesujących nas informacji. Tego typu niespójności zależności należy unikać i właśnie temu służą procesy normalizacyjne. W modelach relacyjnych chodzi przecież o łatwe łączenie i odnajdywanie danych według właściwie określonego klucza.

Przy omawianiu normalizacji baz danych nie sposób pominąć pioniera w dziedzinie relacyjnych baz danych Edgara Codd, który zaproponował kluczowe pojęcia związane z tym procesem.

Zostały one zapisane w postaci postulatów, z którymi można zapoznać się na poniższej stronie:

<https://brasil.cel.agh.edu.pl/~09seenglert/index3911.html?category=podstawy&menu=2&language=PL>

Dzięki tym zasadom udaje się uniknąć wszelkich anomalii wynikających z nadmiarowego lub błędnego zarządzania danymi.

Poniżej pokazane zostały zasady tworzenia poprawnych baz relacyjnych. Są to bardzo ważne zasady, nie należy jednak uczyć się ich na pamięć na tym etapie nauki. Nie przygotowujemy się do roli projektantów baz danych, a mamy jedynie wiedzieć czym jest relacyjna baza danych, jak jest zbudowana i na jakiej zasadzie tworzy się w niej powiązania.

Dla porządku jednak zdecydowałem się na umieszczenie poniższych informacji w naszym materiale.

Pierwsza postać normalna 1NF

Wśród najważniejszych czynników określających zasadę działania tej reguły, należy wymienić:

Usunięcie powtarzających się grup w ramach pojedynczych tabel.

Tworzenie oddzielnej tabeli dla pojedynczego zestawu danych relacyjnych.

Określenie każdego z zestawów dzięki kluczowi podstawowemu.

Reguła ta dotyczy atomowości danych, a więc sytuacji, gdy tabela (encja) przechowuje dane atomowo. Każda komórka to konkretna informacja, co zapewnia najbardziej efektywną pracę z zapytaniami. Dzięki tej podstawowej formie pojawia się też klucz główny. To on odpowiedzialny jest za identyfikację każdego wiersza. Nie występują tutaj kolekcje, a struktura bazodanowa opiera się na pojedynczych informacjach w pojedynczych wierszach.

Druga postać normalna 2NF

Wprowadzenie drugiej formy normalnej ma na celu przede wszystkim:

Budowę osobnych tabel dla zestawów wartości powiązanych z wieloma zapisami.

Ustalenie relacji pomiędzy tabelami przy użyciu klucza obcego.

Zgodnie z teorią tej reguły, poszczególne zapisy powinny być zależne wyłącznie od klucza głównego konkretnej tabeli. Przykładowo, warto pochylić się nad poruszonym wcześniej adresem klienta. Informacji tej potrzebować mogą tabele z danymi klientów, ale także dotyczące zamówień, dostaw, fakturowania czy należności. Nie musisz przetrzymywać adresu klienta w każdej tabeli z osobna, jeśli znormalizujesz dane do jednej tabeli z adresami.

Trzecia postać normalna 3NF

Głównym celem implementacji tej formy w relacyjnej bazie danych jest:

Usunięcie pól, które nie zależą od klucza.

Jeżeli wartości zapisu nie stanowią części klucza rekordu, wówczas nie powinny się tam znajdować. Ogólnie rzecz biorąc, zawsze, gdy zawartość grup lub pól dotyczy więcej niż jednego rekordu w tabeli, warto przenieść je do osobnej encji.

Przykładowo, możesz potrzebować nazwy i adresu uniwersytetu, który kończyły osoby ubiegające się o pracę i zostały wpisane do tabeli przeznaczonej dla kandydatów. Z drugiej strony, lista wszystkich uczelni wyższych może być potrzebna również w innych celach. Jeżeli informacje o uniwersytetach znajdują się w tabeli Kandydaci, wówczas nie da się korzystać z informacji dotyczących szkół, bez odwoływania się do tabeli zawierającej kandydatów. Informacje o uczelniach będą niepełne i ograniczone wyłącznie do tych, które zostały ukończone przez osoby kandydujące do pracy. W takim przypadku przydatne jest stworzenie osobnej tabeli Uniwersytety oraz połączenie jej z Kandydatami za pomocą klucza kodu uczelni.

Jak już wspomniałem, zagadnieniami normalizacji oraz optymalizacji nie będziemy się zajmowali, więc materiał dotyczący poszczególnych reguł należy potraktować jako ciekawostkę 😊.

Administrator bazy danych – co robi, zakres obowiązków, zarobki administratora DB.

Każda baza danych ma swojego opiekuna. Osobę (zespół osób), które dbają o spójność bazy, o jej bezpieczeństwo oraz zajmują się

tworzeniem mechanizmów analitycznych. Są to administratorzy baz danych.

Administrator baz danych przede wszystkim odpowiada za funkcjonowanie systemu bazodanowego.

Poza tym realizuje zadania związane z tworzeniem kwerend pod potrzeby zadań stawianych przez użytkowników baz. Scala dane, przygotowuje raporty oraz zestawienia.

Praca na stanowisku DBA. Jakie są zarobki administratorów baz danych?

Wynagrodzenie admina Oracle czy PostgreSQL w dużej mierze zależy od doświadczenia. Warunki wynagrodzenia za pracę w branży administracji danymi wahać się mogą od kilku do kilkunastu tysięcy miesięcznie przy zatrudnieniu na umowę o pracę. Według najnowszych danych, DBA mogą liczyć na zarobki w wysokości:

Młodszy specjalista – do 5670 PLN

Specjalista – do 9420 PLN

Starszy specjalista – do 14 530 PLN

Expert DBA – od 20.000 PLN

Gdyby ktoś byłby zainteresowany taką pracą, to zachęcam do zgłębiania tematów związanych z tym zagadnieniem.

Materiały dodatkowe:

Hurtownia danych - co to jest i jak działa? Porównanie z relacyjną bazą danych.

FILM: <https://www.youtube.com/watch?v=K5kUfaeFMLk>

Kluczowe modele bazy danych

FILM: <https://www.youtube.com/watch?v=J4l0gyLQFzI>

Jak bazy danych działają od środka?

FILM: <https://www.youtube.com/watch?v=h9FjxdfoYIU>