

Wprowadzenie do Baz Danych

Projekt Stacji serwisowania pojazdów

Prowadzący mgr inż. Piotr Parewicz

Spis treści:

1.	Zakres i cel projektu	
2.	Definicja systemu	
1.	Perspektywy użytkowników	
3.	Model konceptualny	6
1.	Definicja	zbiorów encji określonych w projekcie
2.	Określenie	atrybutów i ich dziedzin 7
3.	Ustalenie	związków i ich typów między encjami 11
4.	Dodatkowe	reguły integralnościowe (reguły biznesowe)
5.	Klucze	kandydujące i główne 14
6.	Schemat	ER na poziomie konceptualnym 15
7.	Problem	pułapek szczelinowych i wachlarzowych 2
4.	Model logiczny	16
1.	Charakterystyka modelu relacyjnego	16
2.	Usunięcie właściwości niekompatybilnych z modelem relacyjnym	16
3.	Proces normalizacji	20
4.	Schemat	ER na poziomie modelu logicznego 20
5.	Więzy integralności	21
6.	Proces denormalizacji	21
5.	Faza fizyczna	22
1.	Projekt	transakcji
2.	Strojenie	bazy danych – dobór indeksów 2
3.	Skrypt	SQL tworzący bazę danych 22
4.	Przykłady	zapytań i poleceń SQL odnoszących się do bazy danych 22
6.	Bibliografia	23
7.	Załączniki	23
1.	Załącznik	nr. 1 24
2.	Załącznik	nr.2 25
3.	Załącznik	nr. 3 Skrypt SQL tworzący bazę danych
4.	Załącznik	nr.4 Skrypt SQL wypełniający bazę danych danymi 30

1. Zakres i cel projektu

Celem projektu jest zaprojektowanie (na poziomie koncepcyjnym oraz logicznym), a także fizyczna implementacja relacyjnej bazy danych obsługującej stację serwisowania pojazdów. Założona baza danych będzie oparta na systemie zarządzania relacyjnymi bazami danych firmy Oracle. Zastosowanym językiem zapytań będzie ANSI SQL.

Oprogramowanie użyte podczas realizacji projektu:

- Oracle Database 12c
- Toad Data Modeler
- SQL Developer

Założenia funkcjonalne:

- W bazie są przechowywane dane dotyczące pojazdów i ich właścicieli którzy są klientami warsztatu
- W bazie są przechowywane dane o pojazdach takie jak marka i model samochodu, przebieg oraz rok produkcji pojazdu
- W bazie są przechowywane dane o dostawach wraz z przypisanymi do nich dostawcami i częściami które dostarczają.
- W bazie są przechowywane dane dotyczące wszystkich części jakich pracownicy warsztatu używają do naprawy pojazdów
- W bazie są przechowywane dane dotyczące wszystkich operacji wykonanych na pojeździe.

1. Definicja systemu

W ramach systemu bazy danych zdefiniowane zostały 4 perspektywy, zdefiniowane

- 1.
2. poprzez przypisanie użytkownikom systemu odpowiednich uprawnień.

1.
 - 1.
 2. Perspektywy użytkowników
 - 3.

Wyróżnione funkcjonalności systemu:

- 1.
2. Podgląd danych personalnych pracowników.
- 3.
4. Modyfikacja/dodawanie/usuwanie danych personalnych pracowników.
- 5.
6. Podgląd danych personalnych klientów.

- 7.
8. Modyfikacja/wprowadzanie/usuwanie danych personalnych klientów
- 9.
10. Podgląd wykonywanych usług na pojazdach.
- 11.
12. Wprowadzanie/usuwanie/modyfikacja danych dotyczących wykonywania usług na pojazdach.
- 13.
14. Podgląd harmonogramu pracy każdego pracownika.
- 15.
16. Podgląd historii serwisowania każdego z pojazdów.
- 17.
18. Modyfikacja/wprowadzanie harmonogramy pracy dla warsztatu.
- 19.
20. Uprawnienia do modyfikacji struktury bazy danych.
- 21.
22. Podgląd w stan części na magazynie
- 23.
24. Wgląd do danych wszystkich dostawców.
- 25.
26. Wgląd w historię serwisowania każdego z pojazdów.
- 27.
28. Możliwość przydziału stanowiska i urządzeń pod daną operację.
- 29.
30. Możliwość otrzymania przez klienta karty stałego klienta.

Opis perspektyw występujących w systemie oraz dostępnych im funkcjonalności:

---(tu obstawiam że nie ma co się tak rozpisywać z tymi funkcjonalnościami i perspektywami tylko ogólny opis co można robić i co widzi każdy użytkownik)---

2.1.1. Administrator

Administrator ma dostęp do wszystkich funkcjonalności systemu oraz modyfikacji struktury bazy danych. Ma on uprawnienia administratora bazy danych Oracle (rola wytworzona automatycznie po stworzeniu bazy).

2.1.2. Właściciel warsztatu

Właściciel może zobaczyć wszystkie dane przechowywane w bazie danych.

Uprawnienia:

2.1.3. Pracownik warsztatu

uprawn

uprawnienia:

ienia:

2.1.4. Klient

Funkcjonalność	Administrator	Właściciel	Pracownik	Klient
1				
2				
3				
4				
5				
6				

7				
8				
9				
10				
11				
12				
13				

14				
15				

- 1.
2. Model konceptualny
- 3.
- 4.

- 1.
2. Definicja zbiorów encji określonych w projekcie

---(tu kilka przykładów różnych encji a nie wszystkie. Definicja encji. Opis jaki element świata fizycznego one reprezentują. Kryteria wyróżniania encji czy tworzyć je osobno czy jako grupę atrybutów i poprzez przykładem. Definicja i przykład encji słownikowej)---

Encja Wlasciciel_Pojazdu:

Encja Pojazd:

Encja Model:

Encja Pozycja_Planu_Serwisowego:

Encja Przegląd:

Encja Pracownik:

Encja Pozycja_Grafiku:

Encja Wykonanie_Operacji:

Encja Czesć:

Encja Stanowisko:

Encja Dostawca:

Encja Dostawa:

Encja Urządzenie:

Encja Typ_Urządzenia:

Encja Operacja:

Encja Stanowisko:

Encja Typ_Stalowiska:

Encja Pozycja_Wyposażenia:

3.2. Ustalenie związków i ich typów między encjami

---(opisać pokrótce wszystkie rodzaje związków, jeden do jednego, jeden do wielu, wiele do wielu, obowiązkowość / opcjonalność + przykłady dla każdego rodzaju)---

3.3 Określenie atrybutów i ich dziedzin

---(jakieś przykłady różnych atrybutów i ich dziedzin, varchar2, number, inne, typy wyliczeniowe, ograniczenia danych że mogą być tylko jakieś z przedziału <x,y>)---

3.4 Dodatkowe reguły integralnościowe

---(warunki określające poprawność danych i warunki łączące atrybuty różnych encji)---

3.5 Klucze kandydujące i główne

---(Definicje kluczy głównych, obcych, kandydujących, kryteria wyboru czy klucz złożony czy sztuczny i która kolumna jako PK)---

3.6 Schemat ER na poziomie konceptualnym

??? tu screen z Toada ???

3.7 Problem pułapek szczelinowych i wachlarzowych

---(Definicja pułapek, anomalii i poparcie przykładem z bazy)---

4. Model Logiczny

4.1 i 4.2 Charakterystyka modelu relacyjnego

---(Definicja modelu relacyjnego, krótki opis. Co się dzieje przekształcając ER na logiczny model, że wiele do wielu zamienia się na bridżujące i od razu opis co tam jest kluczem i jakie są atrybuty, że w relacyjnym nie ma miejsca na atrybuty wielowartościowe)---

4.3 Proces normalizacji

---(Definicja, kryteria normalizacji, że musimy wybrać kompromis pomiędzy wydajnością i pozbyciem się anomalii)---

4.4 Schemat ER na poziomie modelu logicznego

---(tu schemat z Toada)---

4.5 Więzy integralności

---(z wykładu podobno wziąć + przykłady np. CHECK, definicje kluczy)---

4.6 Proces denormalizacji

---(w sumie o tym kompromisie powiemy w 4.3 więc tu się tylko doda że aby zwiększyć wydajność robi się operację odwrotną do normalizacji - denormalizację)---

5. Faza fizyczna

5.1 i 5.4 Projekt transakcji i przykłady zapytań i poleceń SQL ---(opis scenariuszy)---

5.2 Strojenie bazy danych - dobór indeksów

---(czy indeks przyspiesza czy nie)---

5.3 Skrypt SQL zakładający bazę danych

---(plik .sql wyeksportowany z działającej pięknie bazy danych)---

1.

2. **Zakres i cel projektu**

Celem projektu jest zaprojektowanie (na poziomie koncepcyjnym oraz logicznym), a także fizyczna implementacja relacyjnej bazy danych obsługującej stację serwisowania pojazdów. Założona baza danych będzie oparta na systemie zarządzania relacyjnymi bazami danych firmy Oracle. Zastosowanym językiem zapytań będzie ANSI SQL.

Oprogramowanie użyte podczas realizacji projektu:

- Oracle Database 12c
- Toad Data Modeler
- SQL Developer

Założenia funkcjonalne:

- W bazie są przechowywane dane dotyczące pojazdów i ich właścicieli którzy są klientami warsztatu
- W bazie są przechowywane dane o pojazdach takie jak marka i model samochodu, przebieg oraz rok produkcji pojazdu
- W bazie są przechowywane dane o dostawach wraz z przypisanymi do nich dostawcami i częściami które dostarczają.
- W bazie są przechowywane dane dotyczące wszystkich części jakich pracownicy warsztatu używają do naprawy pojazdów
- W bazie są przechowywane dane dotyczące wszystkich operacji wykonanych na pojeździe.

1.

2. **Definicja systemu**

W ramach systemu bazy danych zdefiniowane zostały 4 perspektywy, zdefiniowane

1.

2. poprzez przypisanie użytkownikom systemu odpowiednich uprawnień.

1.

1.

2. **Perspektywy użytkowników**

3.

Wyróżnione funkcjonalności systemu:

- 1.
2. Podgląd danych personalnych pracowników.
- 3.
4. Modyfikacja/dodawanie/usuwanie danych personalnych pracowników.
- 5.
6. Podgląd danych personalnych klientów.
- 7.
8. Modyfikacja/wprowadzanie/usuwanie danych personalnych klientów
- 9.
10. Podgląd wykonywanych usług na pojazdach.
- 11.
12. Wprowadzanie/usuwanie/modyfikacja danych dotyczących wykonywania usług na pojazdach.
- 13.
14. Podgląd harmonogramu pracy każdego pracownika.
- 15.
16. Podgląd historii serwisowania każdego z pojazdów.
- 17.
18. Modyfikacja/wprowadzanie harmonogramy pracy dla warsztatu.
- 19.
20. Uprawnienia do modyfikacji struktury bazy danych.
- 21.
22. Podgląd w stan części na magazynie
- 23.
24. Wgląd do danych wszystkich dostawców.
- 25.
26. Wgląd w historię serwisowania każdego z pojazdów.
- 27.
28. Możliwość przydziału stanowiska i urządzeń pod daną operację.
- 29.
30. Możliwość otrzymania przez klienta karty stałego klienta.

Opis perspektyw występujących w systemie oraz dostępnych im funkcjonalności:

2.1.1. Administrator

Administrator ma dostęp do wszystkich funkcjonalności systemu oraz modyfikacji struktury bazy danych. Ma on uprawnienia administratora bazy danych Oracle (rola wytworzona automatycznie po stworzeniu bazy).

2.1.2. Właściciel warsztatu

Właściciel może zobaczyć wszystkie dane przechowywane w bazie danych. Zajmuje się opracowaniem grafiku dla swoich pracowników. Ma możliwość zatwierdzenia wszystkich dostaw a także ma uprawnienia do modyfikacji wyposażenia hali serwisowej.

2.13. Pracownik warsztatu

Pracownik ma podgląd do magazynu części, do harmonogramu pracy, wszystkie informacje potrzebne do wykonywania operacji.

2.1.4. Klient

Klient ma możliwość wprowadzenia do bazy danych swoich danych osobowych a także dane dotyczące swojego pojazdu. Klientowi zostaje udostępniona lista operacji jaką warsztat wykonuje na pojeździe i także może on wybrać jaką operację chce zrealizować w przypadku naprawy. Dostaje informacje o planowanym czasie realizacji.

1.

2. Model konceptualny

Model konceptualny jest to jeden z pierwszych etapów projektowania bazy danych. Jest to opis wymagań w postaci sformalizowanej, abstrahujący od problemów implementacyjnych. Przedstawia wszystkie wymagania funkcjonalne, definiuje encje i relacje między nimi, bez wnikania w użycie w implementacji. Jest przedstawiany za pomocą schematu ER.

1.

1.

2. Definicja zbiorów encji określonych w projekcie

3.

4.

5.

Encja – wyróżniony obiekt który powinien być odzwierciedlony w bazie danych. Własności encji zapisuje się za pomocą atrybutów.

Jest to decyzją projektową czy dany obiekt fizyczny powinien być reprezentowany jako oddzielna encja. Jeżeli ten obiekt nie będzie wykorzystany w wielu relacjach z innymi encjami to może zostać zapisany jako grupa atrybutów. W przeciwnym wypadku tworzone są dwie encje które są w relacji ze sobą.

Jako przykład możemy podać encje Model gdzie można by było umieścić informacje o modelu w encji Pojazd, jednakże tworzyło by to problem z przydzieleniem planu serwisowania dla każdego pojazdu.

Encja słownikowa – encja której celem jest ułatwić kontrolę poprawności wprowadzanych przez użytkownika wartości atrybutu. Jako przykład możemy podać encję jednostki organizacyjnej gdzie występuje atrybut *Typ*, którego wartością jest typ jednostki organizacyjnej np. "Departament", "Wydział". Zbiór takich wartości jest mało-liczny i rzadko ulega modyfikacji.

W projekcie mieliśmy zaprojektować przykład dziedziczenia podany na rysunku poniżej

Mieliśmy do wyboru 3 możliwości wyboru najlepszego rozwiązania do naszej bazy

1. Utworzenie jednej encji Usługa, gdzie atrybutami byłyby Naprawa i Przegląd. Została przez nas odrzucona ze względu na to iż chcielibyśmy aby móc wykonać naprawę bez przeglądu przez co nie było by to możliwe stosując to rozwiązanie.
2. Stworzenie jednej encji która by była podtypem jednocześnie naprawy i przeglądu i była by w z obiema encjami w relacji jeden do jednego. Tę możliwość także odrzuciliśmy ze względu na brak takich atrybutów.
3. Utworzenie dwóch encji Naprawa i Przegląd które są w relacji z encją Pojazd. Wybraliśmy tę opcję gdyż jest to rozsądne rozwiązanie dla naszej bazy, gdzie klient ma do wyboru czy chce aby zrobić przegląd czy też wykonać operacje naprawy na swoim pojeździe.

Przykłady opisu kliku encji:

Encja Wlasciciel_Pojazdu:

Encja pozwalająca na przechowywanie podstawowych informacji o właścicielach pojazdów, a także możemy się dowiedzieć czy klient jest stałym klientem czy też nie.

Encja Pojazd:

Encja zawierająca dane wszystkich pojazdów których właściciele są klientami warsztatu.

Encja Pracownik:

Encja przechowująca dane o pracownikach danego warsztatu. Możemy się dowiedzieć jaką operacje może wykonać każdy z pracowników.

1.

2.

Określenie atrybutów i ich dziedzin

3.

Wszystkie atrybuty posiadające typ VARCHAR2 mają dziedzinę zawsze określoną, jako ilość znaków (CHAR) a nie, jako ilość bitów. Wszystkie klucze główne są obowiązkowe.

Atrybuty wartości typu INTEGER są liczbami całkowitymi z przedziału -32768 do 32767.

Atrybuty wartości typu NUMBER są liczbami zmiennoprzecinkowymi z precyzją do 38 znaków.

Atrybuty wartości typu DATE są zapisywane w formacie DD-MON-YY

Typy wyliczeniowe

Enum – typ danych wyliczeniowy może przechowywać wartości wybrane z danej listy, maksymalna ilość składowych listy wynosi 65535 elementów, np. Płeć w encji Właściciel_Pojazdu (M,K)

•

- Encja Wlasciciel_Pojazdu

Nazwa atrybutu	Typ i dziedzina	Obowiązkow y	Unikatowy	Opis
Id_wlasciciel a	INTEGER	TAK	TAK	Klucz główny
Imie	VARCHAR2(30)	TAK	NIE	Imię klienta
Nazwisko	VARCHAR2(30)	TAK	NIE	Nazwisko klienta
Miasto	VARCHAR2(30)	TAK	NIE	

Ulica	VARCHAR2(30)	TAK	NIE	
Nr_budynku	INTEGER	TAK	NIE	
Nr_lokalu	INTEGER	NIE	NIE	
Nr_telefonu	VARCHAR2(30)	NIE	NIE	

Adres_e_mai l	VARCHAR2(30)	NIE	NIE	
Staly_klient	VARCHAR2(30)	TAK	NIE	Informacja czy klient jest stałym klientem

-
- Encja Pojazd

Nazwa atrybutu	Typ i dziedzina	Obowiązkow y	Unikatowy	Opis
Id_pojazdu	INTEGER	TAK	TAK	Klucz główny
		TAK		

Nr_rejestracyjny	VARCHAR2(30)		NIE	
Rok produkcji	INTEGER	TAK	NIE	
Przebieg	INTEGER	TAK	NIE	Wartość w km

-
- Encja Pracownik

Nazwa atrybutu	Typ i dziedzina	Obowiązkowy	Unikatowy	Opis
	INTEGER	TAK	TAK	

Id_pracowni ka				Klucz główny
Imie	VARCHAR2(30)			Imię pracowni ka
Nazwisko	VARCHAR2(30)			Nazwisko pracowni ka
Nr_telefonu	INTEGER			
Zawód	VARCHAR2(30)			Rodzaj wykonyw anej pracy

1.

1.

2.

encjami

3.

Ustalenie

związków i ich typów między

Związek jeden do jednego - każdemu rekordowi (krotce) z tabeli nadrzędnej odpowiada dokładnie jeden rekord (jedna krotka) w powiązanej z nią tabeli podrzędnej i odwrotnie
przykład: jednemu modelowi pojazdu odpowiada jedna pozycja planu serwisowego.

Związek jeden do wielu - każdemu rekordowi (krotce) z tabeli nadrzędnej może odpowiadać wiele rekordów (krotek) w powiązanej z nią tabeli podrzędnej, ale każdemu rekordowi (krotce) z tabeli podrzędnej musi odpowiadać dokładnie jeden rekord w powiązanej z nią tabelą nadrzędną
przykład: jeden właściciel może mieć wiele pojazdów, ale pojazd ma jednego właściciela

Związek wiele do wielu - każdemu rekordowi (krotce) z tabeli nadrzędnej może odpowiadać wiele rekordów w powiązanej z nią tabeli podrzędnej i odwrotnie
przykład: jedna część do wielu operacji, wiele części do jednej operacji

Obowiązkowość atrybutów polega na konieczności zamieszczenia informacji danych np. pojazd nie może być zarejestrowany w bazie bez podania numeru rejestracyjnego.
Opcjonalność polega na tym że wartość atrybutu nie musi wystąpić np. numer lokalu Właściciela Pojazdu.

Wszystkie klucze główne są obowiązkowe.

Związki są obowiązkowe wtedy gdy dany obiekt nie może istnieć bez drugiego np. Książka nie może istnieć bez autora, natomiast autor może istnieć bez książki.

Związki nieobowiązkowe oznaczają iż dana encja może istnieć z pustymi kluczami obcymi.

1.

1.

2.

Dodatkowe reguły integralnościowe

3.

4.

5.

Warunki łączące atrybuty różnych encji – atrybut w jednej encji wpływa na dziedzinę atrybutu w drugiej encji jeśli występuje między nimi relacja.

Przykład:

Mamy encje Zamówienie w której jest warunek mówiący że można złożyć zamówienie tylko o danej wadze mniejszej niż X. Atrybut waga jest wyliczany z atrybutów ilość (z encji Pozycja zamówienia) i masa(z encji Towar).

Dodac przykład

1.

1.

2. **Klucze kandydujące i główne**
3.

Klucz główny – jest atrybutem który jednoznacznie identyfikuje dany rekord w tabeli. Jeśli klucz główny składa się z kilku pól określa się go mianem złożonego klucza głównego. Jest najistotniejszym kluczem w bazie danych.

Klucz obcy - kombinacja jednego lub wielu atrybutów tabeli, które wyrażają się w dwóch lub większej liczbie relacji. Wykorzystuje się go do tworzenia relacji pomiędzy parą tabel, gdzie w jednej tabeli ten zbiór atrybutów jest kluczem obcym, a w drugiej kluczem głównym. Jako przykład możemy podać encję Pojazd gdzie kluczem głównym jest ID_Pojazdu, a kluczem obcym jest ID_Wlasciciela gdzie ta wartość jest kluczem głównym w Encji Wlasciciel_Pojazdu.

Kluczem kandydującym możemy nazwać atrybuty które nie są kluczami głównymi ale także mogą jednoznacznie identyfikować dany atrybut w bazie danych.

Z przyczyn bezpieczeństwa zdecydowaliśmy się na stworzenie sztucznych kluczy ID.

Wprowadzenie sztucznego klucza daje nam gwarancję jednoznaczności i spójności danych. Poniżej znajdują się atrybuty które pełnią rolę potencjalnych kluczy kandydujących poszczególnych encji znajdujących się w naszej bazie.

Nazwa Encji	Potencjalny klucz kandydujący
Czesc	Nr_seryjny
Dostawa	Numer_dostawy
Pojazd	Nr_rejestracyjny
Pracownik	Nr_telefonu
Stanowisko	Nr_stanowiska

Urządzenie	Nr_seryjny_urzadzenia
------------	-----------------------

1.

1.

2.

Schemat

ER na poziomie konceptualnym

3.

Schemat ER na poziomie konceptualnym znajduje się w załączniku

1.

1.

2.

Problem

pułapek szczelinowych i

wachlarzowych

3.

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

Pułapka taka może wystąpić gdy co najmniej dwa związki tyłu jeden do wielu wychodzą z tej samej encji

Przykład :

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

Pułapka taka może wystąpić gdy w modelu znajduje się co najmniej jeden związek o minimalnej krotności zero, który jest elementem ścieżki pomiędzy powiązanymi encjami

Przykład:

1.

2. **Model logiczny**

3.

4.

1.

- 3.
- 4.
- 5.

Model logiczny stworzono w programie Toad Data Modeler. Aby otrzymać model logiczny należało usunąć niekompatybilności z modelem relacyjnym. Usunięto związki wiele do wielu. Dodano w tym celu niezbędne tabele bridge'ujące. Atrybutami tej encji są klucze obce relacji połączonych z nią w związek. Nie dodaje się do nich oddzielnego klucza głównego. Nazwy encji zostały zamienione na liczbę mnogą w celu odróżnienia relacji od encji. Mają one także ustalone klucze główne. Nazwy atrybutów nie potrzebowały zmian, ponieważ były unikalne w ramach encji już na poziomie conceptualnym. Schemat ER na poziomie conceptualnym nie zawierał pól wyliczalnych. Wszystkie atrybuty miały określone dziedziny.

Poniżej przykład zastąpienia związku wiele do wielu z modelu konceptualnego poprzez wstawienie dodatkowej tabeli bridge'ującej:

1.		
2.	Proces	normalizacji
3.		

Cele normalizacji:

- Normalizacja ma na celu takie przekształcenie relacji, by uniknąć redundancji i anomalii
- Przekształcenie relacji do kolejnych postaci normalnych wiąże się często ze zmniejszeniem ilości pamięci potrzebnej do przechowywania informacji
- Unikanie powtórzeń pozwala na łatwiejszą i szybszą aktualizację danych
- Doprowadzenie bazy do wysokiej postaci normalnej może spowolnić odczyt w dużych bazach ze względu na skomplikowany schemat danych
- W większości przypadków po znormalizowaniu bazy danych przychodzi kolej na rozważanie możliwości wykonania odwrotnej operacji (a więc denormalizacji), polegającej na połączeniu niektórych znormalizowanych tabel, z myślą o przyspieszeniu dostępu do pewnych danych

Decyzją projektową jest wypracowanie kompromisu pomiędzy niepożądanymi anomaliami a wydajnością systemu. Wykonuje się to poprzez częściową denormalizację bazy danych.

Przykład decyzji projektowej:

W naszej bazie mamy dwie encje Właściciel_Pojazdu i Pojazd, przy denormalizacji a więc zrobieniu jednej encji gdzie będą zawarte atrybuty obu tych encji mamy sytuację gdzie przy wpisywaniu pojazdu do naprawy za każdym razem musimy wpisywać dane właściciela które są udostępnione wszystkim pracownikom warsztatu. Wprowadza to możliwą niespójność i nie zachowuje niezależności przechowywania danych. W postaci znormalizowanej (dwie encje połączone związkiem jeden do wielu) pracownicy znają dane tylko konkretnego pojazdu a nie właściciela.

Rodzaje anomalii jakie mogą wystąpić :

Anomalia usuwania – usuwanie części informacji powoduje utratę informacji których nie chcielibyśmy stracić

Przykład:

Anomalia dodawania – wprowadzenie pewnej informacji jest możliwe tylko wtedy, gdy jednocześnie wprowadzamy jakąś inną informację, która może być obecnie niedostępna

Przykład:

Anomalia aktualizacji - wartość występująca w wielu miejscach może ulec zmianie w jednym miejscu a w drugim nie co powoduje niespójność danych

Przykład:

1.

1.

2.

Schemat

ER na poziomie modelu logicznego

- 3.
- 4.
5. Schemat ER na poziomie modelu logicznego w załączniku
- 6.

1.

1.

2.

3.

Więzy integralności

Więzami integralności są klucze główne.

PRIMARY KEY (= połączenie NOT NULL oraz UNIQUE) - Ograniczenie to zapewnia unikalność wartości w kolumnie, dlatego jest nałożone na klucze główne. W danej tabeli tylko jeden z atrybutów może być PRIMARY KEY. Wszystkie klucze główne muszą być unikalne.

FOREIGN KEY - Ograniczenie zapewniające integralność danych z obydwu tabel znajdujących się w relacji.

UNIQUE - Zapewnia unikalną wartość danego atrybutu dla każdej z krotek.

NOT NULL - Atrybuty z danym ograniczeniem nie mogą mieć wartości null

DEFAULT - Wprowadzanie do kolumn domyślnych wartości przy tworzeniu nowych krotek.

CHECK – ograniczenie wartości dopuszczalnych do zapisania jako wartość atrybutu

CHECK(KOSZT>0)
przykład

1.

2. **Faza fizyczna**

1.

1.

2. Projekt transakcji oraz przykłady zapytań i poleceń SQL odnoszących się do bazy danych

3.

1.

2. **Strojenie bazy danych – dobór indeksów**

3.

1.

1.

2.

3.

4.

Skrypt

SQL tworzący bazę danych