

## Cykl życia systemu bazy danych

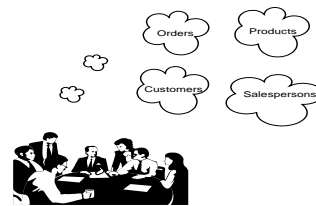
## Cykl życia systemu bazy danych

- Cyklem życia systemu bazy danych nazywamy zbiór kroków niezbędnych do zaprojektowania globalnego schematu logicznego bazy danych, alokacji danych w sieci komputerowej oraz zdefiniowania lokalnych schematów baz danych.

## Analiza wymagań miniświata (1)

- Wymagania miniświata definiujemy w oparciu o rozmowy, ankiety, zapytania skierowane do producentów i konsumentów danych. Wynikiem analizy wymagań jest formalna specyfikacja tych wymagań. Specyfikacja ta zawiera opis niezbędnych danych, naturalne związki pomiędzy danymi, oraz opis platformy programowej do implementacji bazy danych

## Analiza wymagań miniświata (1)

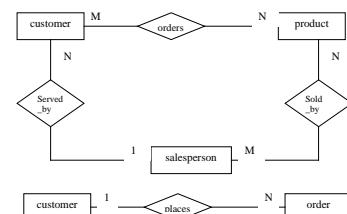


## Projekt logiczny (2)

- W tym kroku jest realizowany projekt schematu pojęciowego bazy danych opisującego wszystkie dane i ich wzajemne powiązania. Do projektowania wykorzystujemy model encji-związków (ER model). Model ten, następnie, transformujemy do modelu implementacyjnego (np. modelu relacyjnego)
  - Modelowanie diagramów ER
  - Integracja perspektyw
  - Transformacja modelu ER do modelu relacyjnego
  - Normalizacja schematów relacji

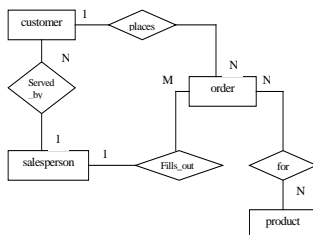
## Modelowanie diagramów ER (2.1)

- Dane przedsiębiorstwa są analizowane i modelowane za pomocą diagramów ER.



## Integracja perspektyw (2.2)

- Fragmenty schematu pojęciowego tworzone przez niezależnych projektantów są integrowane w jeden schemat globalny. Rozwiązywane są niespójności nazw encji i związków. Wykonywana jest analiza synonimów, agregacja i generalizacja encji



## Transformacja modelu ER do modelu relacyjnego (2.3)

- Diagram ER jest transformowany do modelu relacyjnego - otrzymujemy schemat logiczny relacyjnej bazy danych w postaci schematów relacji.

## Transformacja modelu ER do modelu relacyjnego (2.3)

Customer		
cust_id	cust_name	...

prod_no	prod_name	qty_in_stock
---------	-----------	--------------

order_no	sales_name	cust_no
----------	------------	---------

order_no	prod_no
----------	---------

sales_name	addr	dept	job_level	vacation_days
------------	------	------	-----------	---------------

## Normalizacja schematów relacji (2.4)

- Dysponując informacją o zależnościach funkcyjnych i wielowartościowych (FD i MVD) dokonujemy normalizacji schematów relacji (3NF, BCNF, 4NF, 5NF)

np. dana jest zależność funkcyjna:

job\_level  $\longrightarrow$  vacation\_days

## Normalizacja schematów relacji (2.4)

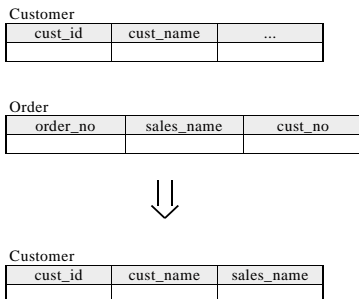
Salesperson			
sales_name	addr	dept	job_level

Sales_vacation	
job_level	vacation_days

## Denormalizacja (3)

- Normalizacja może prowadzić do pogorszenia efektywności przetwarzania zapytań. Denormalizacja polega na rozszerzeniu relacji o dodatkowe atrybuty w celu poprawy efektywności. Denormalizacja wymaga uwzględnienia częstości zapytań, ich priorytetów, wolumenów danych
- Założenie: 90% zapytań ma następujący charakter:  
Kto obsługiwał klienta o nazwisku “Koszlajda”?

## Denormalizacja (3)



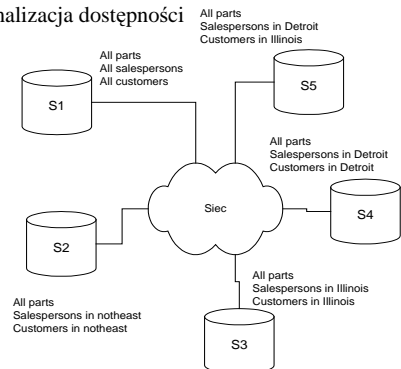
## Rozproszenie danych (4)

- Faza rozproszenia danych składa się z dwóch kroków:
  - fragmentacji danych (schemat fragmentacji)
  - alokacji danych (schemat alokacji)
- W tej fazie jest uwzględniane środowisko fizyczne (m. in. konfiguracja sieci)
- Schemat fragmentacji** opisuje odwzorowanie typu “jeden-do-wielu” fragmentacji schematu relacji na podrelacje (tzw. fragmenty). Podrelacje stanowią logiczne jednostki fizycznie rozproszone na jednym lub wielu stanowiskach

## Rozproszenie danych (4)

- Schemat alokacji** opisuje fizyczną alokację podrelacji - tj. przypisanie danej podrelacji do stanowiska
  - 1:1 - rozproszona niereplikowana baza danych
  - 1:N - rozproszona replikowana baza danych
    - rozproszona w pełni replikowana baza danych
    - rozproszona częściowo replikowana baza danych
- Cele projektu bazy danych w środowisku rozproszonym:
  - separacja fragmentacji i alokacji relacji (pierwsza jest logiczna, druga fizyczna)
  - kontrola replikacji
  - niezależność od lokalnego SZBD

Cele: minimalizacja czasu odpowiedzi  
minimalizacja kosztów komunikacji  
maksymalizacja dostępności



## Projekt schematu logicznego lokalnej bazy danych i projekt fizyczny

- Celem tej fazy jest projekt schematu logicznego lokalnej bazy danych, zdefiniowanie struktur fizycznych i określenie metod dostępu
- SQL (model relacyjny)**

## SQL (model relacyjny)

```
create table customer
( cust_no integer,
  cust_name char(15),
  cust_addr char(30),
  sales_name char(15),
  prod_no integer,
  primary key (cust_no),
  foreign key (sales_name)
    reference salesperson,
  foreign key (prod_no)
    reference product);

create [unique] index indexname on relation
(columnname [asc | desc], columnname)

create unique index nazw_ind on customer
(cust_name desc)
```

### **Implementacja bazy danych, jej strojenie, modyfikacje i monitorowanie**

- Po zaprojektowaniu schematu logicznego, można zaimplementować bazę danych korzystając z języka definicji danych (SQL) dowolnego SZBD. Od momentu utworzenia można realizować dostęp do bazy danych via SQL
- Od momentu uruchomienia SBD system ten powinien być monitorowany w celu sprawdzenia czy spełnia nałożone na niego wymagania efektywnościowe
- W przypadku, gdy wymagania nie są spełnione należy dokonać modyfikacji schematu logicznego lub fizycznego w celu poprawy efektywności działania systemu

### **Implementacja bazy danych, jej strojenie, modyfikacje i monitorowanie**

- Podobnie, należy dokonać modyfikacji schematu logicznego lub fizycznego w przypadku zmiany wymagań lub ewolucji miniświata. Może to pociągać za sobą konieczność modyfikacji aplikacji użytkowych.
- Cykl życia systemu obejmuje więc monitorowania działania systemu bazy danych, modyfikacje schematu logicznego oraz fizycznego, oraz strojenie aplikacji użytkowych